



**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E  
INDUSTRIAL**

**CARRERA DE TELECOMUNICACIONES**

**Tema:**

---

**DISPENSADOR INTELIGENTE DE FRUTOS SECOS BASADO EN  
ARQUITECTURA IoT PARA PRODUCCIÓN PERSONALIZADA  
ORIENTADO A LA INDUSTRIA 4.0 PARA LA EMPRESA LULA ORGANIC**

---

Trabajo de Integración Curricular Modalidad: Proyecto de Investigación, presentado  
previo a la obtención del título de Ingeniero en Telecomunicaciones

**ÁREA:** Electrónica

**LÍNEA DE INVESTIGACIÓN:** Tecnologías de la Información y Sistemas de Control

**AUTOR:** Bryan Patricio Silva Naranjo

**TUTOR:** Ing. Ana Pamela Castro Martin, Mg.

**Ambato – Ecuador**

**septiembre – 2022**

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del Trabajo de Integración Curricular con el tema: **DISPENSADOR INTELIGENTE DE FRUTOS SECOS BASADO EN ARQUITECTURA IoT PARA PRODUCCIÓN PERSONALIZADA ORIENTADO A LA INDUSTRIA 4.0 PARA LA EMPRESA LULA ORGANIC**, desarrollado bajo la modalidad Proyecto de Investigación por el señor **BRYAN PATRICIO SILVA NARANJO**, estudiante de la Carrera de Telecomunicaciones , de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la ejecución de la Unidad de Integración Curricular y la obtención del título de tercer nivel, de grado en la Universidad Técnica de Ambato y sus reformas y el numeral 7.4 del respectivo instructivo.

Ambato, septiembre 2022.

-----  
Ing. Ana Pamela Castro Martin, Mg.

TUTOR

## AUTORÍA

El presente trabajo de Integración Curricular titulado: **DISPENSADOR INTELIGENTE DE FRUTOS SECOS BASADO EN ARQUITECTURA IoT PARA PRODUCCIÓN PERSONALIZADA ORIENTADO A LA INDUSTRIA 4.0 PARA LA EMPRESA LULA ORGANIC** es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, septiembre 2022.



---

Bryan Patricio Silva Naranjo

C.C. 1803621968

AUTOR

## APROBACIÓN TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Integración Curricular presentado por el señor **BRYAN PATRICIO SILVA NARANJO**, estudiante de la Carrera Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad proyecto de investigación, titulado **DISPENSADOR INTELIGENTE DE FRUTOS SECOS BASADO EN ARQUITECTURA IoT PARA PRODUCCIÓN PERSONALIZADA ORIENTADO A LA INDUSTRIA 4.0 PARA LA EMPRESA LULA ORGANIC**, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la ejecución de la Unidad de Integración Curricular y la obtención del título de tercer nivel, de grado en la Universidad Técnica de Ambato y sus reformas y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora presidente del Tribunal.

Ambato, septiembre 2022.

-----

Ing. Pilar Urrutia, Mg.  
PRESIDENTE DEL TRIBUNAL

-----

Ing. Andrea Sánchez Zumba, Mg.  
PROFESOR CALIFICADOR

-----

Ing. Clara Sánchez Benitez, Mg.  
PROFESOR CALIFICADOR



## DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Integración Curricular como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Integración Curricular en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, septiembre 2022.



---

Bryan Patricio Silva Naranjo

C.C. 1803621968

AUTOR

## **Dedicatoria**

*El presente proyecto de grado se lo dedico a mis padres, Soraya y Cristian, por ser mi principal pilar durante toda mi vida y haberme enseñado valores y principios éticos que me han permitido ser un buen ser humano al igual que ellos. Papá, mamá, ¡Esto es por ustedes!*

***Pato***

## **Agradecimiento**

*A mi familia, por su incondicional apoyo y por siempre creer en mí; a mis amigos, por los incontables momentos de alegría, sacrificio y melancolía a mi lado; a aquellos docentes dignos de admirar, que forjaron mis conocimientos durante toda la carrera, especialmente a mi tutora Ing. Pamela Castro, quién desde los primeros semestres despertó un gran interés en mi hacia la carrera de Telecomunicaciones.*

***Pato***

## RESUMEN EJECUTIVO

El presente proyecto de investigación se enfocó en la implementación de un Dispensador inteligente de cuatro variedades de frutos secos categorizadas en tres tamaños distintos, basado en una arquitectura IoT propuesta por el investigador, la misma que sirve de modelo referencial para más proyectos IoT de interés, debido a que, en el mundo de IoT, no existe una arquitectura estandarizada que sirva de base para proyectos enfocados a la Industria 4.0.

La arquitectura consta de tres capas. La capa de dispositivos, la cual cuenta con sensores y actuadores que permiten la adquisición y visualización de datos junto con el accionar de mecanismos diseñados e impresos en 3D que dispensan los frutos secos de manera adecuada hacia un solo recipiente en común. La capa de procesamiento de datos, la cual es de Fog computing, ya que el microcontrolador Arduino Mega en conjunto con la NodeMCU son los encargados de la toma de decisiones y envío de datos válidos hacia el servidor, el mismo que se encuentra dentro de la tercera y última capa, la capa de servicios en la Nube, la cual, consta de múltiples archivos de programación web elaborados con conocimientos en PHP, HTML, CSS, SQL y JavaScript que en conjunto forman Interfaces Web para que tanto el propietario de la empresa, como los clientes, tengan acceso a un entorno de visualización IoT, el mismo que, interactúa con una base de datos, registrando múltiples variables como datos de usuarios, pedidos y los valores adquiridos por los sensores del dispensador inteligente.

Se espera que el diseño de este prototipo sea de gran ayuda para la empresa Lula Organic en cuanto se refiere a mejorar sus servicios hacia los clientes, de tal forma que, se satisfaga su demanda de participación y personalización de productos por medio de una experiencia agradable e innovadora que está dispuesta a ser recompensada con un cargo adicional a cambio de disfrutarla, hechos que son sustentados por las características de la Industria 4.0.

**Palabras clave:** arquitectura IoT, industria 4.0, sensores, actuadores, servidor, base de datos, interfaz web, Fron-End, Back-End, dispensador.

## ABSTRACT

This research project focused on the implementation of an intelligent dispenser of four varieties of nuts categorized in three different sizes, based on an IoT architecture proposed by the researcher, which serves as a reference model for more IoT projects of interest, because, in the world of IoT, there is no standardized architecture that serves as the basis for projects focused on Industry 4.0.

The architecture consists of three layers. The devices layer, which has sensors and actuators that allow the acquisition and visualization of data together with the actuation of mechanisms designed and printed in 3D that dispense the nuts in an appropriate way into a single common container. The data processing layer, which is from Fog computing, since the Arduino Mega microcontroller together with the NodeMCU are in charge of making decisions and sending valid data to the server, which is inside the third and last layer, the layer of services in the Cloud, which consists of multiple web programming files elaborated with knowledge in PHP, HTML, CSS, SQL and JavaScript that together form Web Interfaces so that both the owner of the company , like customers, have access to an IoT visualization environment, the same one that interacts with a database, registering multiple variables such as user data, orders and the values acquired by the smart dispenser sensors.

It is expected that the design of this prototype will be helpful to Lula Organic company in terms of improving its services to customers, in such a way that their demand for participation and product customization is satisfied through a pleasant and innovative experience and that is willing to be rewarded with an additional charge in exchange for enjoying it, facts that are supported by the characteristics of Industry 4.0.

**Keywords:** IoT architecture, industry 4.0, sensors, actuators, server, database, web interface, Front-End, Back-End, dispenser.

## ÍNDICE GENERAL

|  |    |
|--|----|
| CAPÍTULO I.....                                    | 1  |
| MARCO TEÓRICO.....                                 | 1  |
| 1.1    Antecedentes Investigativos.....            | 1  |
| 1.2    Contextualización del problema.....         | 2  |
| 1.3    Fundamentación Teórica.....                 | 4  |
| 1.3.1    Sistemas Electrónicos.....                | 4  |
| 1.3.1.1    Sensores .....                          | 4  |
| Sensores Mecánicos .....                           | 5  |
| Sensores Ultrasónicos .....                        | 5  |
| Sensores Fotoeléctricos .....                      | 5  |
| Sensores Magnéticos .....                          | 5  |
| 1.3.1.2    Actuadores .....                        | 6  |
| Actuadores Neumáticos .....                        | 6  |
| Actuadores Mecánicos .....                         | 6  |
| Actuadores Hidráulicos.....                        | 6  |
| Actuadores Eléctricos.....                         | 6  |
| Actuadores Térmicos .....                          | 6  |
| 1.3.1.3    Microcontrolador .....                  | 7  |
| 1.3.2    Internet de las cosas (IoT).....          | 7  |
| Conectividad .....                                 | 7  |
| 1.3.3    Las Cuatro Revoluciones Industriales..... | 7  |
| 1.3.3.1    Industria 1.0 .....                     | 8  |
| 1.3.3.2    Industria 2.0 .....                     | 9  |
| 1.3.3.3    Industria 3.0 .....                     | 9  |
| 1.3.3.4    Industria 4.0 .....                     | 9  |
| Ventajas y desventajas .....                       | 10 |

|          |  |    |
|----------|--|----|
| 1.3.4    | Producción Personalizada en la industria 4.0.....      | 11 |
| 1.3.5    | IoT en la Industria 4.0 .....                          | 12 |
| 1.3.6    | Arquitectura IoT.....                                  | 12 |
| 1.3.7    | Ventajas y desventajas de la tecnología IoT .....      | 13 |
| 1.3.8    | Protocolos para IOT .....                              | 14 |
| 1.3.8.1  | Protocolos inalámbricos para IoT (Tecnologías).....    | 14 |
| 1.3.8.2  | Protocolos alámbricos para IoT .....                   | 15 |
| 1.3.8.3  | Protocolos de mensajería IoT cliente - servidor .....  | 16 |
|          | HTTP.....  | 16 |
|          | XMPP.....  | 16 |
|          | CoAP.....  | 16 |
|          | MQTT .....   | 16 |
|          | AMQP .....   | 16 |
| 1.3.9    | Procesamiento de Datos .....                           | 17 |
| 1.3.9.1  | Edge Computing .....                                   | 18 |
| 1.3.9.2  | Fog Computing .....                                    | 18 |
| 1.3.9.3  | Cloud Computing.....                                   | 18 |
| 1.3.10   | Análisis de Arquitecturas IoT de interés .....         | 19 |
| 1.3.10.1 | Arquitectura con siete capas – Modelo IoTWF.....       | 19 |
| 1.3.10.2 | Arquitectura con seis capas – Modelo Intel IoT.....    | 20 |
| 1.3.10.3 | Arquitectura con cinco capas.....                      | 21 |
| 1.3.10.4 | Arquitectura con cuatro capas.....                     | 22 |
| 1.3.10.5 | Arquitectura con tres capas.....                       | 23 |
| 1.3.11   | Sistemas mediante codificación .....                   | 24 |
| 1.3.12   | Dispensador Automático de productos .....              | 24 |
| 1.3.13   | Frutos secos, sus beneficios y como conseguirlos ..... | 25 |
| 1.3.14   | Servidor.....  | 25 |

|                                |  |    |
|--------------------------------|--|----|
| 1.3.14.1                       | Servidor Web .....                                   | 26 |
| 1.3.14.2                       | Servidor FTP.....                                    | 26 |
| 1.3.15                         | Sistema Gestor de Base de Datos.....                 | 26 |
| 1.3.15.1                       | Bases de Datos Relacionales (SQL) .....              | 26 |
| 1.3.15.2                       | Bases de Datos Relacionales (NoSQL) .....            | 27 |
| 1.3.16                         | Conceptos de Desarrollo Web.....                     | 27 |
| 1.3.16.1                       | Front-End .....                                      | 27 |
| 1.3.16.2                       | Back-End .....                                       | 28 |
| 1.4                            | Objetivos .....                                      | 29 |
| Objetivo General .....         | 29   |    |
| Objetivos Específicos.....     | 29   |    |
| CAPÍTULO II .....              | 30   |    |
| METODOLOGÍA .....              | 30   |    |
| 2.1                            | Materiales .....                                     | 30 |
| 2.2                            | Métodos.....   | 31 |
| 2.2.1                          | Modalidad de Investigación .....                     | 31 |
| 2.2.2                          | Recolección de Información .....                     | 31 |
| 2.2.3                          | Procesamiento y Análisis de Datos .....              | 32 |
| 2.2.4                          | Desarrollo del Proyecto.....                         | 32 |
| CAPÍTULO III.....              | 35   |    |
| RESULTADOS Y DISCUSIONES ..... | 35   |    |
| 3.1                            | Análisis y Discusión de los resultados .....         | 35 |
| 3.2                            | Desarrollo de la propuesta.....                      | 35 |
| 3.2.1                          | Análisis de Factibilidad.....                        | 37 |
| 3.2.2                          | Arquitectura IoT propuesta para el Dispensador ..... | 38 |
| 3.2.3                          | Consideraciones Técnicas .....                       | 39 |
| Selección de Sensores .....    | 39   |    |



|   |    |
|---|----|
| Selección de Actuadores .....   | 41 |
| Selección del controlador .....   | 43 |
| Fuentes de alimentación .....   | 44 |
| Selección de la Base de Datos.....  | 45 |
| 3.2.4 Mecanismos de Dispensación .....  | 45 |
| 3.2.4.1 Mecanismo de Dispensación para el Contenedor #1 .....                           | 46 |
| 3.2.4.2 Mecanismo de Dispensación para el Contenedor #2 .....                           | 51 |
| 3.2.4.3 Mecanismo de Dispensación para los Contenedores #3 y #4.....                    | 56 |
| 3.2.5 Bosquejo completo de la arquitectura del sistema .....                            | 61 |
| 3.2.5.1 Capa de Dispositivos .....  | 63 |
| 3.2.5.2 Capa de Procesamiento de Datos (FOG COMPUTING) .....                            | 65 |
| 3.2.5.3 Capa de Servicios en la Nube .....  | 67 |
| Selección del Protocolo de mensajería IoT .....   | 68 |
| Selección modelo de servicio en la Nube .....   | 68 |
| 3.2.6 Diseño de la placa de circuito impreso (PCB) .....                                | 69 |
| 3.2.7 Direccionamiento I2C .....  | 72 |
| 3.2.8 Mecanismo guía de la caída de los frutos secos.....                               | 74 |
| 3.2.9 Esquema del Dispensador .....   | 77 |
| 3.2.10 Diagrama de flujo para el Algoritmo de programación del Arduino<br>Mega2560..... | 79 |
| a. Diagrama de flujo para la Subrutina ultrasónico1().....                              | 81 |
| b. Diagrama de flujo para la Subrutina ultrasónico2().....                              | 82 |
| c. Diagrama de flujo para la Subrutina ultrasónico3().....                              | 83 |
| d. Diagrama de flujo para la Subrutina ultrasónico4().....                              | 84 |
| e. Diagrama de flujo para la Subrutina sensor_hall_1() .....                            | 85 |
| f. Diagrama de flujo para la Subrutina sensor_hall_2() .....                            | 86 |
| g. Diagrama de flujo para la Subrutina sensor_hall_3() .....                            | 87 |

|        |   |     |
|--------|---|-----|
| h.     | Diagrama de flujo para la Subrutina sensor_hall_4() .....               | 88  |
| i.     | Diagrama de flujo para la Subrutina Lectura_Datos().....                | 89  |
| j.     | Diagrama de flujo para la Subrutina Envío_Datos() .....                 | 92  |
| k.     | Diagrama de flujo para la Subrutina pesar_grano1() .....                | 93  |
| l.     | Diagrama de flujo para la Subrutina pesar_grano2().....                 | 93  |
| m.     | Diagrama de flujo para la Subrutina pesar_grano3().....                 | 94  |
| n.     | Diagrama de flujo para la Subrutina pesar_grano4().....                 | 94  |
| o.     | Diagrama de flujo para la Subrutina Dispensar_grano1() .....            | 95  |
| p.     | Diagrama de flujo para la Subrutina Dispensar_grano2() .....            | 95  |
| q.     | Diagrama de flujo para la Subrutina Dispensar_grano3() .....            | 96  |
| r.     | Diagrama de flujo para la Subrutina Dispensar_grano4() .....            | 97  |
| 3.2.11 | Diagrama de flujo para el Algoritmo de programación de la NodeMCU<br>99 |     |
| a.     | Diagrama de flujo para la Subrutina Consultas_Pedidos() .....           | 100 |
| b.     | Diagrama de flujo para la Subrutina Lectura_Datos_NODEMCU().            | 101 |
| 3.2.12 | Desarrollo de la Interfaz Web de Inicio de Sesión.....                  | 101 |
| 3.2.13 | Desarrollo de la Interfaz Web de Registro de Usuarios.....              | 104 |
| 3.2.14 | Registro de un Administrador .....                                      | 105 |
| 3.2.15 | Desarrollo de la Interfaz Web de Administrador .....                    | 106 |
|        | Presentación de Páginas Web para el Usuario Administrador.....          | 109 |
| 3.2.16 | Desarrollo de la Interfaz Web de Cliente.....                           | 114 |
|        | Presentación de Páginas Web para el Usuario Cliente.....                | 117 |
| 3.2.17 | Instalación de herramientas y librerías necesarias del prototipo.....   | 120 |
|        | Instalación de la Base de Datos.....                                    | 120 |
|        | Creación de las Tablas de Datos .....                                   | 122 |
|        | Instalación de las librerías en Arduino IDE .....                       | 126 |
|        | Conexión de la Base de Datos con el Servidor .....                      | 128 |

|  |     |
|--|-----|
| Conexión del microcontrolador con el servidor.....                           | 129 |
| 3.2.18 Seguridad y Encriptación de Datos .....                               | 129 |
| 3.3 Pruebas de Funcionamiento y Resultados.....                              | 131 |
| 3.4 Presupuesto .....  | 135 |
| CAPÍTULO IV.....   | 139 |
| CONCLUSIONES Y RECOMENDACIONES.....  | 139 |
| 4.1 Conclusiones .....   | 139 |
| 4.2 Recomendaciones.....   | 140 |
| BIBLIOGRAFÍA .....   | 142 |
| ANEXOS .....   | 147 |
| Anexo 1: Circuito esquemático del proyecto .....                             | 148 |
| Anexo 2: Programación del Microcontrolador Arduino MEGA .....                | 149 |
| Anexo 3: Programación del Microcontrolador NodeMCU.....                      | 171 |
| Anexo 4: Programación Web del Sitio de Log-In.....                           | 178 |
| Anexo 5: Programación Web del Sitio “Registrar usuarios” .....               | 182 |
| Anexo 6: Programación Web de script “Log Out” .....                          | 186 |
| Anexo 7: Programación Web de las Páginas de la Interfaz de Administrador ... | 187 |
| Anexo 8: Especificaciones del sensor HC-SR04 .....                           | 188 |
| Anexo 9: Especificaciones del sensor KY-024.....                             | 191 |
| Anexo 10: Especificaciones del servo MG-995.....                             | 192 |
| Anexo 11: Especificaciones de la celda de Carga de 5 KG .....                | 194 |
| Anexo 12: Especificaciones del controlador HX711 .....                       | 195 |
| Anexo 13: Especificaciones de la placa NodeMcu esp8266.....                  | 201 |
| Anexo 14: Especificaciones de la placa Arduino Mega 2560.....                | 204 |
| Anexo 15: Especificaciones del LCD 16x2 I2C .....                            | 212 |

## ÍNDICE DE FIGURAS

|   |    |
|---|----|
| Figura 1. Estructura del Sistema Electrónico .....              | 4  |
| Figura 2. Evolución de la Industria .....                       | 8  |
| Figura 3. Arquitectura General IoT .....                        | 13 |
| Figura 4. Protocolos de comunicación inalámbrica para IoT ..... | 14 |
| Figura 5. Protocolos de comunicación IoT .....                  | 17 |
| Figura 6. Tipos de Procesamiento de Datos .....                 | 17 |
| Figura 7. Modelos de servicios en la Nube .....                 | 19 |
| Figura 8. Arquitectura IoTWF (siete capas) .....                | 20 |
| Figura 9. Arquitectura IoT Intel (seis capas) .....             | 21 |
| Figura 10. Arquitectura IoT de 5 capas .....                    | 22 |
| Figura 11. Arquitectura IoT de 4 Capas .....                    | 23 |
| Figura 12. Arquitectura IoT de 3 capas por IBM .....            | 24 |
| Figura 13. Arquitectura IoT Propuesta .....                     | 38 |
| Figura 14. Celda de Carga de 5 Kg .....                         | 41 |
| Figura 15. Fuentes DC empleadas .....                           | 44 |
| Figura 16. Semillas de Sésamo (Ajonjolí) .....                  | 45 |
| Figura 17. Maní [49] .....                                      | 46 |
| Figura 18. Pistacho y Nuez .....                                | 46 |
| Figura 19. Colector primario.....                               | 47 |
| Figura 20. Pieza Giratoria A .....                              | 47 |
| Figura 21. Colector Final .....                                 | 48 |
| Figura 22. Pieza Giratoria B – Vista 1 .....                    | 48 |
| Figura 23. Pieza Giratoria B – Vista 2 .....                    | 49 |
| Figura 24. Pieza Giratoria B – Vista 3 .....                    | 49 |
| Figura 25. Tapa de mecanismos giratorios – Vista 1 .....        | 50 |
| Figura 26. Tapa de mecanismos giratorios – Vista 2.....         | 50 |
| Figura 27. Mecanismo del contenedor # 1 ensamblado.....         | 51 |
| Figura 28. Soporte Frontal para el Servo 1 .....                | 51 |
| Figura 29. Soporte Posterior para el Servo 1 .....              | 52 |
| Figura 30. Pieza giratoria A .....                              | 52 |
| Figura 31. Colector Primario.....                               | 53 |
| Figura 32. Colector Final .....                                 | 53 |

|  |    |
|--|----|
| Figura 33. Pieza Giratoria B – Vista 1 .....                               | 54 |
| Figura 34. Pieza Giratoria B – Vista 2 .....                               | 54 |
| Figura 35. Tapa de mecanismos giratorios – Vista 1 .....                   | 55 |
| Figura 36. Tapa de mecanismos giratorios – Vista 2 .....                   | 55 |
| Figura 37. Mecanismo del contenedor # 2 ensamblado – Parte 1 .....         | 56 |
| Figura 38. Mecanismo del contenedor # 2 ensamblado – Parte 2.....          | 56 |
| Figura 39. Soporte Frontal para el Servo 1 .....                           | 57 |
| Figura 40. Soporte Posterior para el Servo 1 .....                         | 57 |
| Figura 41. Pieza giratoria A .....   | 58 |
| Figura 42. Colector Único.....   | 58 |
| Figura 43. Pieza Giratoria B – Vista 1 .....                               | 59 |
| Figura 44. Pieza Giratoria B – Vista 2 .....                               | 59 |
| Figura 45. Tapa de mecanismos giratorios – Vista 1 .....                   | 60 |
| Figura 46. Tapa de mecanismos giratorios – Vista 2.....                    | 60 |
| Figura 47. Mecanismo de los contenedores #3 y #4 ensamblado – Parte 1..... | 61 |
| Figura 48. Mecanismo de los contenedores #3 y #4 ensamblado – Parte 2..... | 61 |
| Figura 49. Bosquejo de la arquitectura Interna del Sistema.....            | 62 |
| Figura 50. Esquema de conexión de un contenedor individual .....           | 64 |
| Figura 51. Esquema de conexión de la celda de carga.....                   | 65 |
| Figura 52. Esquema de conexión de un contenedor individual .....           | 65 |
| Figura 53. Esquema de conexión entre Arduino Mega y NODEMCU.....           | 66 |
| Figura 54. Diseño Circuito Impreso.....                                    | 70 |
| Figura 55. Diseño Circuito Impreso 3D – Capa Superior .....                | 70 |
| Figura 56. Diseño Circuito Impreso 3D – Capa Inferior .....                | 71 |
| Figura 57. Circuito Impreso Implementado .....                             | 71 |
| Figura 58. Conexiones del Sistema.....                                     | 72 |
| Figura 59. Pines del Direcccionamiento I2C .....                           | 72 |
| Figura 60. Direcccionamiento I2C Establecido en Arduino.....               | 73 |
| Figura 61. Diseño de la Tolva Invertida – Vista 1 .....                    | 74 |
| Figura 62. Diseño de la Tolva Invertida – Vista 2 .....                    | 74 |
| Figura 63. Tolva Invertida Implementada – Vista 1 .....                    | 75 |
| Figura 64. Tolva Invertida Implementada – Vista 2 .....                    | 75 |
| Figura 65. Boca final de la Tolva – Vista 1 .....                          | 76 |

|  |     |
|--|-----|
| Figura 66. Boca final de la Tolva – Vista 2 .....                              | 76  |
| Figura 67. Diseño 3D del dispensador completo – Vista 1 .....                  | 77  |
| Figura 68. Diseño 3D del dispensador completo – Vista 2.....                   | 78  |
| Figura 69. Dispensador de frutos Secos Implementado – Vista 1 .....            | 78  |
| Figura 70. Dispensador de frutos Secos Implementado – Vista 2 .....            | 79  |
| Figura 71. Diagrama de Flujo General del Arduino Mega .....                    | 80  |
| Figura 72. Diagrama de Flujo de la Subrutina ultrasónico1().....               | 81  |
| Figura 73. Diagrama de Flujo de la Subrutina ultrasónico2().....               | 82  |
| Figura 74. Diagrama de Flujo de la Subrutina ultrasónico3().....               | 83  |
| Figura 75. Diagrama de Flujo de la Subrutina ultrasónico4().....               | 84  |
| Figura 76. Diagrama de Flujo de la Subrutina Sensor_hall_1() .....             | 85  |
| Figura 77. Diagrama de Flujo de la Subrutina Sensor_hall_2() .....             | 86  |
| Figura 78. Diagrama de Flujo de la Subrutina Sensor_hall_3() .....             | 87  |
| Figura 79. Diagrama de Flujo de la Subrutina Sensor_hall_4() .....             | 88  |
| Figura 80. Diagrama de Flujo de la Subrutina Lectura_datos() – Parte 1/3 ..... | 89  |
| Figura 81. Diagrama de Flujo de la Subrutina Lectura_datos() – Parte 2/3 ..... | 90  |
| Figura 82. Diagrama de Flujo de la Subrutina Lectura_datos() – Parte 3/3 ..... | 91  |
| Figura 83. Diagrama de Flujo de la Subrutina Envío_datos() .....               | 92  |
| Figura 84. Diagrama de Flujo de la Subrutina Pesar_Grano1().....               | 93  |
| Figura 85. Diagrama de Flujo de la Subrutina Pesar_Grano2().....               | 93  |
| Figura 86. Diagrama de Flujo de la Subrutina Pesar_Grano3().....               | 94  |
| Figura 87. Diagrama de Flujo de la Subrutina Pesar_Grano4().....               | 94  |
| Figura 88. Diagrama de Flujo de la Subrutina Dispensar_Grano1() .....          | 95  |
| Figura 89. Diagrama de Flujo de la Subrutina Dispensar_Grano2() .....          | 96  |
| Figura 90. Diagrama de Flujo de la Subrutina Dispensar_Grano3() .....          | 97  |
| Figura 91. Diagrama de Flujo de la Subrutina Dispensar_Grano4() .....          | 98  |
| Figura 92. Diagrama de Flujo General de la NodeMCU .....                       | 99  |
| Figura 93. Diagrama de Flujo de la Subrutina Consultas_Pedidos().....          | 100 |
| Figura 94. Diagrama de Flujo de la Subrutina Lectura_Datos_NODEMCU().....      | 101 |
| Figura 95. Diagrama de Secuencia “Log In” .....                                | 102 |
| Figura 96. Interfaz Web “Log In” .....   | 103 |
| Figura 97. Interfaz Web “Log In” .....   | 103 |
| Figura 98. Diagrama de Secuencia “Registrar Usuario” .....                     | 104 |

|  |     |
|--|-----|
| Figura 99. Interfaz Web “Registrar Usuario” .....                                | 105 |
| Figura 100. Registro de un Administrador.....                                    | 106 |
| Figura 101. Registro de un Administrador.....                                    | 106 |
| Figura 102. Diagrama de Secuencia de las Páginas Dinámicas de Administrador..    | 107 |
| Figura 103. Diagrama de Secuencia de las Páginas Estáticas de Administrador..... | 108 |
| Figura 104. Diagrama de Secuencia de las Páginas Mixtas de Administrador.....    | 109 |
| Figura 105. Dashboard General de Administrador .....                             | 110 |
| Figura 106. Segundo Dashboard de Administrador.....                              | 110 |
| Figura 107. Tercer Dashboard de Administrador .....                              | 111 |
| Figura 108. Dashboard “Pedidos y Ventas” de Administrador.....                   | 111 |
| Figura 109. Dashboard “Datos Históricos” de Administrador.....                   | 112 |
| Figura 110. Dashboard “Clientes Registrados” de Administrador .....              | 112 |
| Figura 111. Dashboard “Administradores Registrados” de Administrador.....        | 112 |
| Figura 112. Tabla de Datos Ascendentes por ID .....                              | 113 |
| Figura 113. Tabla de Datos Descendentes por Nivel de Grano 3 .....               | 113 |
| Figura 114. Tabla de Datos De estados de los contenedores.....                   | 114 |
| Figura 115. Diagrama de Secuencia de las Páginas Dinámicas de Clientes .....     | 115 |
| Figura 116. Diagrama de Secuencia de las Páginas Estáticas de Administrador..... | 116 |
| Figura 117. Diagrama de Secuencia “Registrar Pedido” Cliente.....                | 117 |
| Figura 118. Dashboard Inicial de Cliente .....                                   | 118 |
| Figura 119. Página Web “Realizar Pedido” – Vista 1 .....                         | 118 |
| Figura 120. Página Web “Realizar Pedido” – Vista 2 .....                         | 119 |
| Figura 121. Página Web “Realizar Pedido” – Vista 3 .....                         | 119 |
| Figura 122. Página Web “Historial de Pedidos” .....                              | 120 |
| Figura 123. Página Web “Credenciales de Cliente” .....                           | 120 |
| Figura 124. Panel de Control de XAMP activado.....                               | 121 |
| Figura 125. Ingreso a PhpMyAdmin.....  | 122 |
| Figura 126. Base de Datos creada .....   | 122 |
| Figura 127. Tablas de Datos .....  | 123 |
| Figura 128. Estructura de la Tabla “contenedores_control” .....                  | 124 |
| Figura 129. Datos de la Tabla “contenedores_control” .....                       | 124 |
| Figura 130. Estructura de la Tabla “pedidos” .....                               | 125 |
| Figura 131. Datos de la Tabla “pedidos” .....                                    | 125 |

|  |     |
|--|-----|
| Figura 132. Estructura de la Tabla “usuarios_clientes” ..... | 126 |
| Figura 133. Datos de la Tabla “usuarios_clientes” .....      | 126 |
| Figura 134. Gestor de Librerías .....                        | 127 |
| Figura 135. Instalación de Librería en Arduino .....         | 128 |
| Figura 136. Carpeta Includes .....                           | 128 |
| Figura 137. Conexión NodeMCU con Servidor .....              | 129 |
| Figura 138. Hashes Almacenados .....                         | 130 |
| Figura 139. Atributo Required .....                          | 131 |
| Figura 140. Caso de Campos vacíos .....                      | 131 |
| Figura 141. Caso de Campos con Datos erróneos.....           | 131 |
| Figura 142. Pruebas con Clientes de la Empresa .....         | 132 |
| Figura 143. Pruebas con Clientes de la Empresa .....         | 132 |
| Figura 144. Pruebas con Clientes de la Empresa .....         | 133 |
| Figura 145. Pruebas con Clientes de la Empresa .....         | 133 |



## ÍNDICE DE TABLAS

|   |     |
|---|-----|
| Tabla 1: Actividades para el Desarrollo del Proyecto (Parte 1/3)..... | 32  |
| Tabla 2. Actividades para el Desarrollo del Proyecto (Parte 2/3)..... | 33  |
| Tabla 3. Actividades para el Desarrollo del Proyecto (Parte 3/3)..... | 34  |
| Tabla 4. Parámetros de Sensores Ultrasónicos .....                    | 39  |
| Tabla 5. Parámetros de Sensores Magnéticos .....                      | 40  |
| Tabla 6. Parámetros de Dispositivos Giratorios.....                   | 42  |
| Tabla 7. Parámetros del controlador .....                             | 43  |
| Tabla 8. Guía para el Direccionamiento I2C .....                      | 73  |
| Tabla 9. Direccionamiento I2C Seleccionado.....                       | 73  |
| Tabla 10. Datos de los clientes de prueba .....                       | 134 |
| Tabla 11. Error Porcentual .....                                      | 135 |
| Tabla 12. Presupuesto de Diseño .....                                 | 136 |
| Tabla 13. Presupuesto de Construcción 1 .....                         | 137 |
| Tabla 14. Presupuesto de Construcción 2 .....                         | 138 |
| Tabla 15. Presupuesto total de implementación.....                    | 138 |

# CAPÍTULO I

## MARCO TEÓRICO

### 1.1 Antecedentes Investigativos

Al finalizar la investigación bibliográfica en los principales repositorios de universidades, así como en diversas bases de datos de artículos científicos, se pueden detallar los siguientes antecedentes:

Daniela Guevara y Jorge Luis Guaca (2017) de la Fundación Universitaria Los Libertadores, en Bogotá, desarrollaron un Dispensador para mascotas controlado remotamente, su funcionamiento fue construido en base a lenguajes de programación web HTML5, CSS, Python y Java Script. Por medio de la interfaz web, el usuario puede controlar el dispensador ensamblado en una Raspberry Pi, de esta manera, se facilita el cuidado de mascotas cuando sus dueños no están presentes en sus hogares [1].

En la Universidad San Ignacio de Loyola de Lima, Carlos Chang, José Chávez, Wendy Chinga, Jhon Fernández y Moisés Miranda (2018) realizaron un proyecto con el tema “Investigación de Pre-Factibilidad para la Fabricación de un Dispensador de Comida para Mascotas Automatizado”, el cual, puede ser controlado por medio de una aplicación en un dispositivo móvil, que acciona los mecanismos el dispensador, el mismo que registra sus datos por medio de códigos QR. Obtuvo como resultado de un estudio estadístico del proyecto, la generación de ganancias usando un WACC de 32.18% [2].

José Cabañeros (2019) de la Universidad de Oviedo, en España, realizó un “Dispensador inteligente de medicamentos”, que personaliza la dispensación de pastillas con horarios establecidos por el médico. Consta de tres sistemas: el sistema de dispensar las pastillas, el cual, contiene un microcontrolador ESP8266 para el control y manipulación del sistema; el sistema de sensado para la detección de objetos; y el sistema de registro en la base de datos. La interfaz web la realizó en Mockup para la monitorización, gestión y control del dispensador. De tal manera que el dispensador de pastillas gestiona correctamente los medicamentos y notifica al paciente mediante la activación de una alarma [3].

Juan Escobar (2019) de la Universidad Técnica de Ambato, desarrolló un “Diseño de sistemas de control industrial de robots basado en industria 4.0”, el sistema de control de robots está basado en la tarjeta Raspberry Pi para la manipulación del robot Kuka Youbot mediante Open Source y una plataforma de entorno de desarrollo ROS para la adquisición de datos de los sensores. Obtuvo como resultado el correcto posicionamiento del brazo robótico Kuka Yubot [4].

Lenin Dávila (2020), de la Universidad Técnica de Ambato propuso un “Sistema Electrónico Automático de Alimentación para mascotas en el hogar”, en el cual, implementó un dispensador automático de comida de mascotas adaptándose a las necesidades de la mascota en cuestión, tiene en cuenta el peso y el tamaño de ésta para una adecuada alimentación. Realizó el proceso en dos etapas: la etapa de control, que consta de un microcontrolador NODEMCU para el accionamiento de los actuadores mediante una interfaz web o aplicación móvil, y la etapa de monitoreo para la vigilancia de la mascota durante su alimentación mediante una cámara IP. Además, cuenta con una base de datos en la nube para el accionamiento remoto en la plataforma firebase. Finalmente, obtuvo como resultado un sistema automático que proporciona las tres comidas diarias a la mascota con opción a establecer las horas específicas para la dosificación de comida de la mascota [5].

## **1.2 Contextualización del problema**

El aumento de la competitividad empresarial, la demanda, la exigencia del consumidor final y los elevados costos de producción, son factores que todo microempresario intenta combatir al momento de emprender. Países como Corea del Sur, Dinamarca, Estados Unidos y Reino Unido disponen de estrategias basadas en la fabricación y personalización inteligente que sitúan a sus industrias en el mercado competitivo global. Los nuevos mercados, se basan en la personalización y la creación de nuevos productos y servicios innovadores en los que el cliente, por supuesto, exige calidad a sus productos, pero está dispuesto a pagar por la experiencia o el servicio más que por el producto en sí. Razones, por las cuales, surge la necesidad de añadir servicios nuevos, creativos y personalizados en las empresas. [6].

El 36% de los consumidores afirman que están interesados en productos o servicios personalizados, no obstante, la oferta de estos por parte de las empresas es muy escasa. Sectores comerciales como turismo, textiles y muebles son aquellos en que, sus compradores, exigen mayor demanda de personalización. Las empresas pueden aprovechar esta demanda y beneficiarse por medio de un sobrecoste que, el 71% de los consumidores que expresan interés en productos personalizados, están dispuestos a pagar a cambio de satisfacerla. Sin embargo, según el estudio, “Readiness for the Future of Production“, realizado en 2018, Ecuador se encuentra en el puesto 89/100 en la evaluación de la preparación para futuro de la producción, esto se debe a la baja puntuación en cuanto a Innovación y Tecnología, aun así, el gobierno impulsa el desarrollo y crecimiento del sector productivo en temas de tecnología y comunicación para lograr procesos más eficientes y servicios con valor agregado, pero muy pocas empresas en el país dan el gran paso a la innovación tecnológica en sus procesos. [7], [8], [9] y [10]

Desde el 2020, autoridades locales y nacionales en el Ecuador tienen como objetivo la transición de las ciudades tradicionales a ciudades inteligentes y sostenibles con el uso de tecnologías para que las personas e industrias puedan mejorar su competitividad. Sin embargo, el mundo de IoT es tan amplio, que ofrece una gran variedad de comunicaciones, redes, sensores, módulos y computación, lo cual, define su principal problema: la falta de una arquitectura o modelo de referencia que sea estandarizado para que los empresarios puedan adoptarla al momento de renovar sus servicios [11], [12].

En Ambato, existe una alta competencia de empresarios o microempresarios, cuyo rubro de negocio es la venta de alimentos a granel. Este sector comercial tiene grandes aspiraciones de crecimiento ante un mercado cada vez más exigente. Ante este desafío, la empresa Lula Organic, promueve el consumo responsable mediante la venta de productos a granel, como semillas, frutos secos, frutas deshidratadas, y dulces varios. Sin embargo, la empresa, se encuentra día a día con la necesidad de satisfacer las crecientes necesidades del cliente en cuanto a una mejor experiencia en sus servicios, factor que pocas o ninguna otra empresa del mismo sector comercial ha logrado identificar. La empresa actualmente no dispone de una tecnología basada en los conceptos de IoT e Industria 4.0 [13].

Este proyecto, hace visible la viabilidad de la implementación de una arquitectura IoT para producción personalizada aplicado a un dispensador automático de frutos secos a granel, beneficia directamente a la empresa Lula Organic y, de esta manera, brinda una mejora en la experiencia de sus servicios para personalizar su pedido en cuanto a variedad y cantidad mediante una interfaz interactiva que se la puede usar en cualquier momento accediendo a esta desde una red WiFi, además, la opción de arquitectura propuesta, puede ser aplicado en varias empresas más, por consiguiente, optimiza recursos, reduce tiempo de verificación de inventarios, obtiene el producto acorde a la necesidad y brinda ese valor agregado que el cliente busca constantemente.

### 1.3 Fundamentación Teórica

#### 1.3.1 Sistemas Electrónicos

Los sistemas electrónicos son un conjunto de circuitos que operan con señales eléctricas con el fin de cumplir funciones específicas. Está formado por tres etapas que se muestran en la figura 1: La etapa de entrada que recoge variables físicas del exterior; la etapa de proceso o control donde se interpretan, procesan, transforman y tratan los datos recogidos para poder tomar decisiones y activar o desactivar dispositivos o actuadores de la etapa de Salida, siendo ésta la última etapa. [14].



**Figura 1.** Estructura del Sistema Electrónico [14]

##### 1.3.1.1 Sensores

Los sensores son dispositivos capaces de detectar magnitudes físicas o químicas de cualquier entorno y transformarlas a magnitudes eléctricas, que pueden ser tratadas en los circuitos electrónicos. Las magnitudes físicas o químicas también son denominadas variables de instrumentación. Las variables eléctricas son las que establecen el tipo de

sensor, que pueden ser Digitales, Analógicos y de Comunicación por Bus. Entre los principales tipos de sensores que se ocupan en el mundo de la automatización se encuentran: [15].

### **Sensores Mecánicos**

Los sensores mecánicos son utilizados principalmente para medir la fuerza o deformación de algún objeto referencial, mediante esta deformación es posible obtener otro tipo de magnitudes físicas de forma indirecta, como peso, presión, temperatura. Generalmente este tipo de sensores posee su propio controlador que se encarga de realizar el acondicionamiento de la señal. [16]

### **Sensores Ultrasónicos**

Son dispositivos que se encargan de medir la distancia que existe con respecto al mismo, siempre y cuando el objeto a medir se encuentre dentro del rango de visión del sensor. Su funcionamiento se basa en el envío de ondas sonoras a una alta frecuencia que se reflejan en los objetos que obstaculizan el paso de las mismas dentro del rango de visión del sensor, estas ondas son rebotadas en la superficie del objeto y retornan al sensor provocando, mediante cálculos básicos, la correcta medición de la distancia entre el objeto y el sensor en base al tiempo generado entre la onda emitida y la recibida. [17]

### **Sensores Fotoeléctricos**

Un sensor fotoeléctrico detecta objetos presentes en su rango de alcance mediante las variaciones de las propiedades ópticas en las superficies del objeto. Constan principalmente de un emisor de luz infrarroja y un receptor para recibir la misma. Cuando la luz emitida es interrumpida o reflejada por el objeto de detección, cambia la cantidad de luz que llega al receptor. El receptor detecta este cambio y lo convierte en una salida eléctrica. [18]

### **Sensores Magnéticos**

Son sensores que efectúan una conmutación electrónica mediante la presencia de un campo magnético externo, próximo y dentro del área sensible. Estos sensores pueden ser sensibles a los polos del imán, o solamente a un polo.

Son sensores que emiten una señal electrónica a su salida mediante la presencia de campos magnéticos, generalmente, trabajan de la mano de imanes debido a que estos producen campos magnéticos que pueden ser detectados por el sensor, la salida emitida puede ser digital o analógica, y dependiendo del fabricante, pueden tener una alta sensibilidad a ambos polos del imán o simplemente a uno. [19]

### **1.3.1.2 Actuadores**

Los actuadores son dispositivos que reciben la orden de un controlador y son capaces de transformar energía hidráulica, eléctrica o neumática en la activación de un proceso con la finalidad de interactuar elementos externos de una forma automática [15].

#### **Actuadores Neumáticos**

Los actuadores neumáticos utilizan como fuente energía al aire comprimido para producir un movimiento deseado, por ejemplo, un pistón de aire [15].

#### **Actuadores Mecánicos**

Los actuadores mecánicos operan mediante la conversión de un movimiento que es producido por la combinación de engranes, poleas, cadenas, piezas guías, etc [15].

#### **Actuadores Hidráulicos**

Los actuadores hidráulicos utilizan como fuente energía a líquidos para producir diferentes tipos de movimientos, a diferencia de los actuadores neumáticos, los actuadores hidráulicos son empleados en movimiento de cargas donde se necesita una fuerza inmensa, por lo tanto, son utilizados en maquinaria pesada [15].

#### **Actuadores Eléctricos**

Son los actuadores más comunes en el mercado, necesitan de una fuente externa de voltaje o amperaje y convierten dicha energía eléctrica en energía mecánica, luminosa, sonora, etc. Son los más utilizados en automatización electrónica [15].

#### **Actuadores Térmicos**

Los actuadores térmicos brindan una alta potencia y su fuente de energía es el calor, emplean la energía térmica para producir algún movimiento deseado [15].

### **1.3.1.3 Microcontrolador**

Los microcontroladores son sistemas embebidos inteligentes y programables en cuya entrada detecta las variables eléctricas enviadas por un sensor, éste se encarga de acondicionarlas y procesarlas para enviar a su salida una orden que es recibida por un actuador. Los microcontroladores cuentan con sus propios recursos, como son, procesador, memorias flash y RAM, y por último, periféricos de entrada y/o salida. [20]

Son dispositivos que cumplen una tarea en específico y son escalables en cuanto a su funcionalidad con módulos externos que facilitan la automatización de proyectos electrónicos. [21]

### **1.3.2 Internet de las cosas (IoT)**

El internet de las cosas es una tecnología basada en la interconexión de dispositivos y sensores como: tablets, cerraduras, computadoras, celulares, aire acondicionado, focos, cámaras entre otros, mediante el internet a través de redes inalámbricas y alámbricas. Su ventaja es que permite conectar dispositivos inteligentes desde cualquier ubicación e interactuar entre sí, en el entorno que les rodea para la toma de decisiones de forma autónoma [22].

#### **Conectividad**

Los dispositivos IoT pueden estar conectados con una amplia variedad de tecnologías, que pueden ir desde tecnologías de Redes de Área Personal hasta redes WWAN como LTE, etc [23].

### **1.3.3 Las Cuatro Revoluciones Industriales**

La cuarta revolución industrial (origen del término Industria 4.0) hace referencia a las cuatro fases de la revolución industrial [24]:

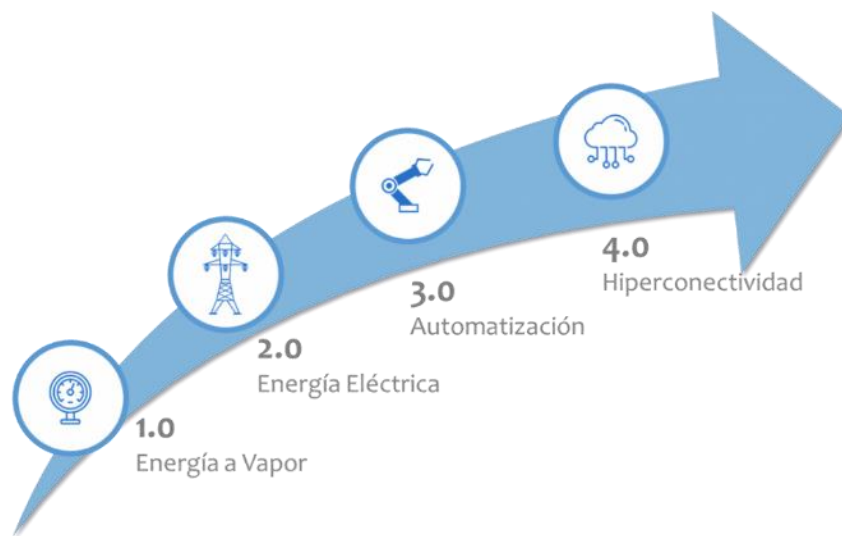
- **Primera revolución industrial.** Máquinas de vapor y ferrocarril en el siglo XIX.
- **Segunda revolución industrial.** Motores eléctricos y producción en masa a principios del siglo XX. Aparece el motor de combustión, se desarrolla el



aeroplano y el automóvil, y como grandes inventos aparece el teléfono y la radio.

- **Tercera revolución industrial.** Automatización y la informática en los años setenta del siglo XX.
- **Cuarta revolución industrial.** Los actuales sistemas ciberfísicos que recopilan y procesan información, toman decisiones inteligentes y ejecutan tareas en entornos cambiantes.

En la figura 2, se muestra la evolución desde la industria 1.0 a la industria 4.0.



**Figura 2.** Evolución de la Industria [25]

### 1.3.3.1 Industria 1.0

La primera revolución industrial supuso posiblemente el mayor cambio socioeconómico, tecnológico y cultural en la historia de la humanidad desde el neolítico. La economía basada en el trabajo manual fue reemplazada por otra dominada por la industria y la manufacturera. Se caracteriza porque la tecnología celebra: 1) el nacimiento del ferrocarril, que revoluciona y comienza a permitir el transporte de personas y mercancías en grandes dimensiones; 2) la máquina de vapor, que incrementa notablemente la capacidad de producción con el carbón como materia prima, ya que era el combustible de la máquina de vapor, descubierta por James Watt (1785); 3) la principal fuente de energía sería el carbón, aunque va naciendo también el petróleo como materia prima. Un factor de impacto es el éxodo de personas del campo a la ciudad y el surgimiento del sector obrero [24].

### **1.3.3.2 Industria 2.0**

La segunda revolución industrial se convierte en un proceso de innovaciones tecnológicas, científicas, sociales y económicas producidas en paralelo con la consolidación del capitalismo como sistema económico. Aparece el motor de combustión, se desarrolla el aeroplano y el automóvil, y como grandes inventos aparece el teléfono y la radio. Las principales fuentes de energía son el gas y el petróleo. Aparecen materias primas derivadas del petróleo y otras que no proviene de la naturaleza tales como el plástico y otros tipos de tejidos que se van a usar en la industria textil; la madera deja de usarse en profusión y la utilización de minerales aumenta [24].

### **1.3.3.3 Industria 3.0**

La tercera revolución industrial se caracteriza en cuanto a tecnología por: microelectrónica como base, computador como la máquina más destacada, Internet el gran dinamizador del cambio y el uso de la energía atómica y energías renovables. En cuanto a sociedad y economía, se caracteriza por: comportamientos globales de los medios de comunicación, un incremento demográfico, un incremento del consumo y la mercadotecnia. En esencia, han evolucionado las tecnologías, pero el concepto no ha variado: “se elimina el material que tanto le ha costado a la naturaleza unir para obtener mediante métodos substractivos el producto final deseado”. Por primera vez en la historia existe la posibilidad de fabricar de un modo diferente, sin embargo, los diseñadores y profesionales se ven limitados debido a las convencionales formas de producción [24].

### **1.3.3.4 Industria 4.0**

Industria 4.0 viene asociado con el nacimiento de la Cuarta Revolución Industrial, corresponde a una nueva manera de organizar los medios de producción, utilizando las tecnologías digitales y la información inteligente de datos a partir del Big Data, es decir, las enormes cantidades de datos que se podrán transmitir entre objetos

inteligentes a través del Internet de las Cosas. El Internet pasa de ser un método de comunicación entre personas a ser un método de comunicación entre objetos y personas [24].

La cuarta revolución industrial trae consigo una tendencia a la automatización total de la manufactura (fabricación). La automatización se basa en los sistemas ciberfísicos facilitados por la Nube (cloud computing) y el Internet de las Cosas, con la ayuda indispensable de la fabricación aditiva mediante las impresoras 3D y, además, el soporte indispensable de la inteligencia artificial y de big data, como tecnologías clave para la conversión de los grandes volúmenes de datos y su uso eficiente en la toma de decisiones [24].

La Industria 4.0 es el producto más tangible de la cuarta revolución industrial y está favoreciendo la fabricación inteligente en un marco revolucionario para diseñar, implantar y gestionar ecosistemas complejos que proporcionan información en tiempo real y posibilitan las interacciones autónomas entre máquinas, sistemas, objetos y cosas. Este modelo permite sacar el máximo partido y rendimiento del Internet de las cosas (IoT), la nube, los big data y la analítica de datos, la inteligencia artificial, las aplicaciones de última generación y la ciberseguridad [24].

Existen cuatro principios de diseño que le atribuye a la industria 4.0 las cuales son [26]:

- Transparencia de la información
- Interoperabilidad a través del internet de las cosas
- Decisiones descentralizadas
- Asistencia técnica a través de la cibernética

### **Ventajas y desventajas**

Las ventajas de las industrias 4.0 son [26]:

- Optimización de los niveles de calidad como precisión en envases y contenido
- Reducción de tiempo y costos de producción
- Seguridad ya que las personas no se exponen a procesos peligrosos
- Requieren de menos personal de operación

Las desventajas de las industrias 4.0 son las siguientes [26]:

- Requiere de personal especializado para controlar la automatización de los procesos
- La dependencia de la tecnología
- Costo de inversión inicial elevado

#### **1.3.4 Producción Personalizada en la industria 4.0**

La exigencia de participación de los clientes en los procesos de diseño y producción de las empresas con el fin de obtener un producto personalizado obliga a las compañías a adaptarse a nuevos modelos de negocio que cumplan con estas expectativas. Esto da lugar a nuevas tendencias de cambio responsables de la variación generada al pasar del antiguo paradigma empresarial (industria 3.0), enfocado en la producción masiva automatizada, a un nuevo sistema de producción que poco a poco se está inculcando en la mente de las compañías: **la personalización masiva de productos**, sistema que es posible gracias la industria 4.0 [27].

Esta nueva metodología de producción consiste en personalizar los productos a gran escala con un costo relativamente bajo, lo que permite desarrollar ventajas competitivas dentro del mercado. En la actualidad algunas compañías ya se han adelantado a este cambio en los procesos de producción, por lo que, los empresarios deben entender cuáles son los factores a tener en cuenta para comenzar una estrategia de este tipo, con aspectos de interés, dado que el nuevo paradigma representa una tendencia importante en el ámbito mundial, por lo que, se piensa irá ganando terreno en un futuro próximo. Éste nuevo paradigma brinda al cliente la oportunidad escoger la etiqueta, cantidad, tipo de sellado, volumen, entre otros factores que no todos los usuarios concuerdan al momento de comprar un mismo producto, y que solo la industria 4.0 ofrece a esta escala [27].

La personalización en la industria 4.0 significa situar al cliente en el centro de todos los procesos, es decir, brindarle aquello que realmente quiere en el tiempo, lugar y formato que desea a un precio igual o similar de un artículo producido en masa. En el mundo existen muchas empresas que han innovado sus procesos hacia la personalización masiva, y los resultados de sus ventas hablan por sí solas; un ejemplo

es la conocida marca alemana automotriz BMW, se estima en promedio que, de todos sus autos vendidos por año, solo dos autos son idénticos en cada variante de sus modelos. La empresa alemana Mymuesli vende mezclas de cereales de alta gama y al mismo tiempo ofrece a sus clientes el servicio de personalizar su propia mezcla de cereales a un precio apenas más alto que los cereales premezclados de forma masiva. Otro ejemplo es la empresa sueca Absolute Vodka, la cual, lanzó al mercado su “Absolute Unique”, producto que fue producido en cuatro millones de unidades con diseños en sus envases totalmente distintos el uno al otro, de esta forma cada una de ellas era única e irrepetible, sus ventas fueron un éxito. La personalización en masa, la interconectividad, la integración de las personas como personal colaborativo y la implantación de nuevas tecnologías son aspectos esenciales en la industria 4.0 [28] y [7].

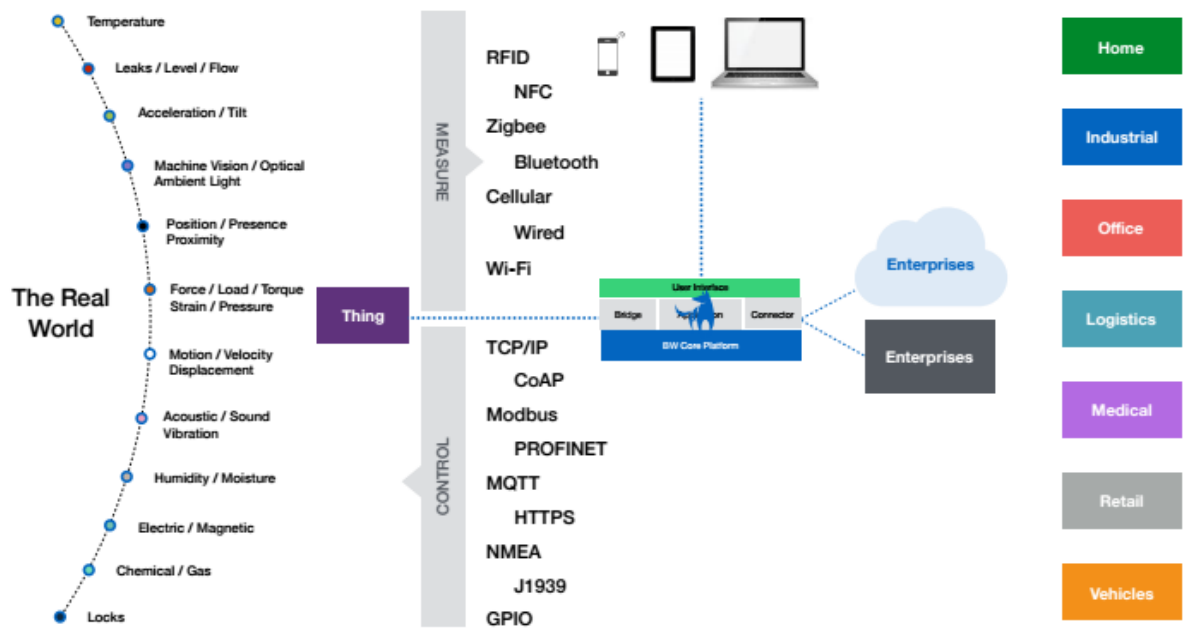
### **1.3.5 IoT en la Industria 4.0**

El IoT está impulsando a la creación de productos cada vez más inteligentes que abren nuevas oportunidades no solo dentro de la producción industrial, sino en cualquier situación y para cualquier tipo de campo, que es una de las tendencias más importantes del siglo XXI, y está destinada a cambiar el enfoque de negocios y mercados con la implementación de avances tecnológicos relacionados con la automatización y la robótica. Esto es gracias a la innovación que están experimentando los sistemas de producción con la adopción de la Industria 4.0 [29].

Con esta cuarta revolución industrial, se ha dejado de separar empresas en aquellas que solo hacen servicios y otras que solo se ocupan de productos, actualmente existe una unión de lo físico con lo digital (llamado "Phygital"), que permite a todos aquellos que tienden a revolucionar su forma de industrializar, la audacia y las habilidades para comenzar a usar la tecnología de IoT e imaginar un mundo diferente, utilizando dispositivos y medios inteligentes autoadaptables para hacer marketing ,y así, llegar a más clientes [29].

### **1.3.6 Arquitectura IoT**

La arquitectura general de la tecnología IoT se detalla en la **Figura 3**.



**Figura 3.** Arquitectura General IoT [22]

La capa de percepción es la capa física, que incluye sensores, actuadores, RFID, dispositivos móviles, entre otros, éstos permiten la comunicación con la puerta de enlace IoT. Detectan y recopilan información sobre el entorno, e identifican diferentes tipos de dispositivos de manera inteligente [22].

La capa de red recopila los datos enviados desde la capa inferior y los enruta a través de la puerta de enlace a internet, proporcionando conectividad, procesamiento de información, conversión de protocolos y manejo de fallas [22].

La capa de aplicación define varias aplicaciones en las que se puede implementar el Internet de las cosas, por ejemplo, hogares inteligentes, ciudades y salud inteligentes. También incluye infraestructuras de servidor y nube, que comparten contenido y proporcionan servicios en tiempo real [22].

### 1.3.7 Ventajas y desventajas de la tecnología IoT

Las ventajas de la tecnología IoT se detallan a continuación [22]:

- La tecnología IoT facilita la vida diaria de las personas y organizaciones públicas y privadas, además, permite monitorear las actividades de las personas y ayuda en la planificación y mejoramiento de cualquier proceso.

- Permite la interacción de objetos inteligentes en cualquier momento, lugar y medio.
- Permite que las actividades cotidianas sean más fáciles para los usuarios dando paso a nuevas posibilidades de vida más confortable mediante la interacción de la persona, el mundo real y las máquinas.
- Ahorro de tiempo y dinero en tareas de gestión de monitorización de presupuestos y suministros.

Las desventajas de IoT se describen en lo siguiente [22]:

- La seguridad y privacidad depende de las aplicaciones de los artefactos de IoT
- La red de comunicación en caso de fallas interrumpirá el funcionamiento de las aplicaciones y servicios de IoT.
- No existe una arquitectura específica estándar definida, las arquitecturas varían según la aplicación de los dispositivos.

### 1.3.8 Protocolos para IOT

#### 1.3.8.1 Protocolos inalámbricos para IoT (Tecnologías)

Existen muchos protocolos de comunicación inalámbrica en el mundo de IoT, cada uno se basa en tecnología y alcance distinto, entre los más destacados son [30] y [31]:

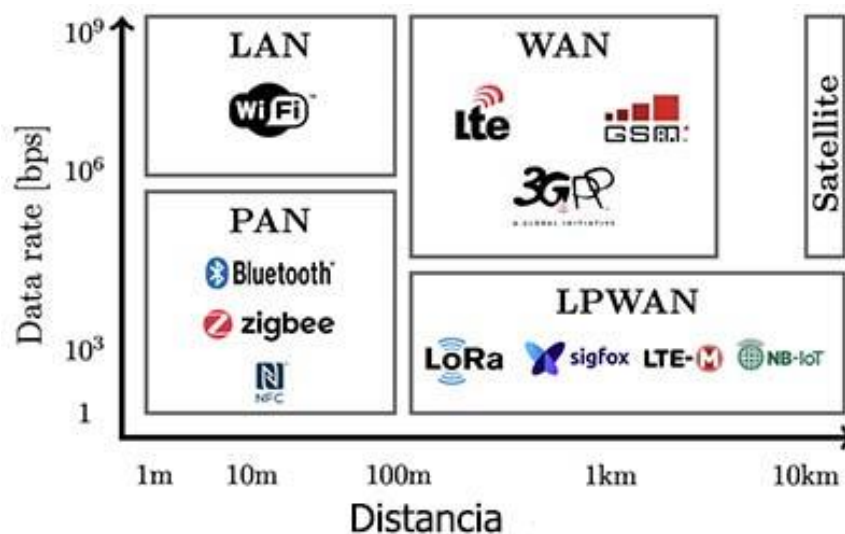


Figura 4. Protocolos de comunicación inalámbrica para IoT [30]

- WirelessHART - IEC 62591, IEEE 802.15.4
- Bluetooth/LE
- 802.15.4 - low-rate wireless personal area network (LR-WPAN).
- 802.11 WiFi
- Zigbee – Basado en IEEE 802.15.4.
- 6LoWPAN - IPv6 over Low power Wireless Personal Area Networks
- IEEE 1609 WAVE
- 802.15.6 WBAN - Wireless Body Area Network.
- Z-Wave – Comunicación inalámbrica, 800 -900 MHz.
- Thread – Es una red tipo mesh basada en IPv6. De bajo poder. Usando 6LoWPAN.
- LTE - Long Term Evolution
- GPRS - General Packet Radio Service
- CDMA - Code Division Multiple Access
- LoRaWAN - Low Power Wide Area Network, 915MHz América, 868 MHz Europa, 433MHz Asia.
- Sigfox - Industrial, Scientific and Medical ISM radio band 868MHz in Europe and 902MHz in the US.
- NFC - Near Field Communication, ISO 14443 a 13.56 MHz.
- DASH7 – ISO/IEC 18000-7 protocolo a 433 MHz.

### **1.3.8.2 Protocolos alámbricos para IoT**

Los protocolos alámbricos son usados para establecer una comunicación entre los sensores y el microcontrolador, los más destacados son [31]:

- RS-232
- RS-422
- UART - Transmisor-Receptor Asíncrono Universal.
- SPI - Serial Peripheral Interface
- RS-485
- I2C - Inter-Integrated Circuit



### **1.3.8.3 Protocolos de mensajería IoT cliente - servidor**

Los protocolos más comunes para comunicar dispositivos IOT con el servidor son los siguientes: [32]

#### **HTTP**

Hypert-Text Transport Protocol. Este protocolo transfiere datos usando los servicios RESTful que incluye los métodos POST, GET, PUT, PATCH y DELETE [31].

#### **XMPP**

Extensible Messaging and Presence Protocol. Es un protocolo abierto desarrollado para el intercambio de información de los dispositivos IoT en tiempo real [31].

#### **CoAP**

Constrained Application Protocol. Es un protocolo alternativo a HTTP pensado para usarse en sistemas de poco ancho de banda y recursos de energía limitados, puede usar los mismos métodos de transferencia de datos que usa HTTP (POST, GET, PUT, PATCH y DELETE), sin embargo, es más complejo con respecto a HTTP, pocas librerías existentes [33].

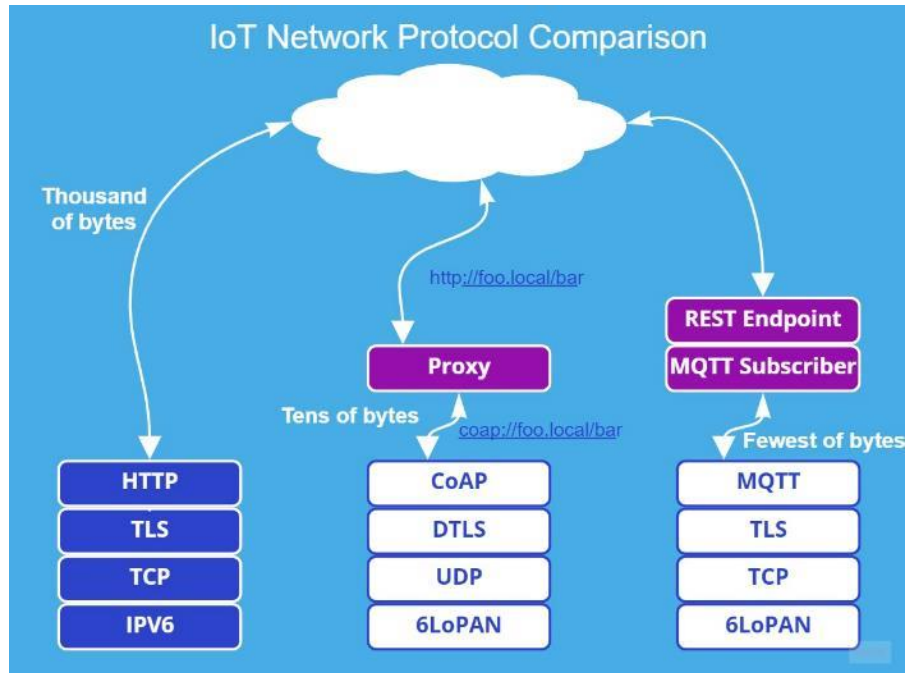
#### **MQTT**

Message Queue Telemetry Transport. Es un protocolo muy usado por la mayoría de los proveedores, funciona en base a su modelo publicador/suscriptor, los dispositivos IoT publican el mensaje mientras que el cliente que consume el mensaje es el suscriptor. Es difícil crear una red MQTT escalable [33].

#### **AMQP**

Advanced Message Queuing Protocol. Es una especificación abierta para envíos de mensajes asíncronos y está definido como estándar internacional ISO/IEC 19464. Utiliza el modelo publicador/suscriptor en colas [33].

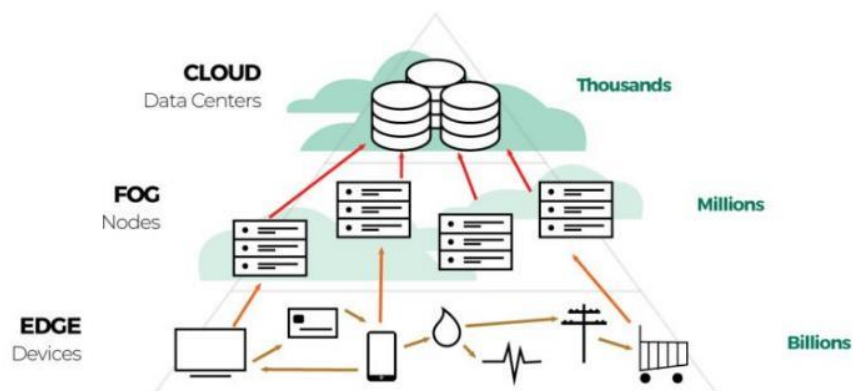
En la **Figura 5** se muestra una comparación general de los protocolos de mensajería IoT más comunes según la cantidad de bytes que pueden soportar.



**Figura 5.** Protocolos de comunicación IoT [32]

### 1.3.9 Procesamiento de Datos

Cuando los dispositivos IoT están conectados a la nube para administrar sus datos de manera centralizada, algunos desafíos se vuelven críticos. No siempre tiene sentido mover todos los datos a la nube para recién procesarlos y tomar decisiones, existen algunos casos en los que el tiempo de respuesta es crítico. En esos casos, la distribución de la capacidad computacional es la solución, existen tres formas principales de hacerlo y están representadas en la **Figura 6** [34].



**Figura 6.** Tipos de Procesamiento de Datos [34]

### **1.3.9.1 Edge Computing**

Edge Computing hace referencia a la computación cercana o de borde. Consiste en desarrollar sistemas de computación que puedan dar una respuesta al instante y en tiempo real, es decir, los datos se procesan en el mismo sensor o en el dispositivo en sí, sin ser transferidos a ninguna parte. Es muy útil cuando los datos deben alterar comportamientos relacionados con el funcionamiento en tiempo real de las máquinas, por ejemplo, detener la fabricación de forma inmediata si se detecta alguna anomalía en algún sensor [35].

### **1.3.9.2 Fog Computing**

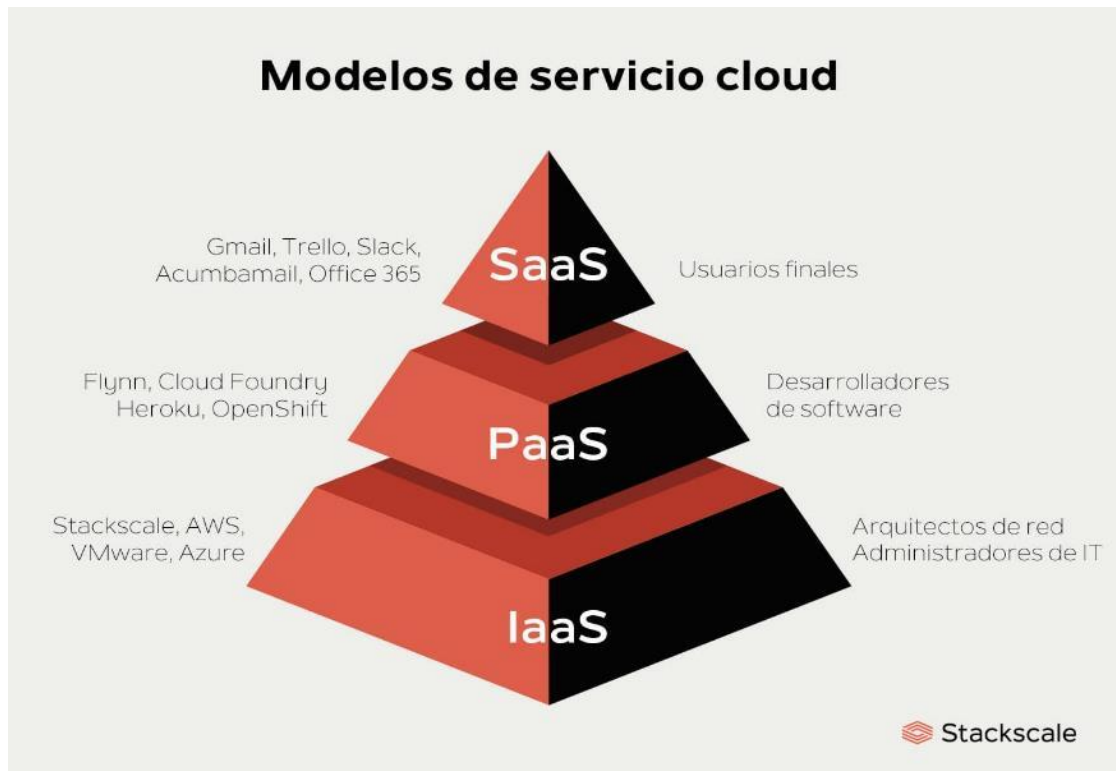
También denominada computación en la niebla, es la capa consiguiente a Edge computing. Aquí los datos son tratados en centros de procesamiento previos a la nube, que actúan de instancia mediadora entre los distintos dispositivos IoT (sensores) y la nube, los datos son tratados y enviados a la nube solo si es necesario, de esta manera, se reduce el tráfico de datos, el consumo de energía y evita el uso de protocolos intermedios. Además, aumenta la seguridad en los datos, ya que, los actuadores se activan con mediciones reales directamente de los sensores sin haber corrido el riesgo de ser vulnerados al enviarlos a la nube para su análisis. En conclusión, el Fog Computing es procesado en los microcontroladores o mini ordenadores conectados directamente a los sensores o dispositivos recolectores de información [35].

### **1.3.9.3 Cloud Computing**

Los datos son tratados y procesados en un servidor remoto al cuál se accede por medio de internet, dichos datos pueden ser tratados por algún software o algoritmo de inteligencia artificial propios del servidor. Es el tipo de procesamiento de datos que produce más latencia [35].

Si los datos ya son procesados con Edge y/o Fog computing, surge la pregunta, ¿para qué se necesita la Nube? Los datos ya procesados generalmente se envían a lagos de datos centralizados en la nube para ser tratados como datos fríos, es decir, que ya no están sujetos a actualizaciones. Pero también se utilizan para analizarlos a otro nivel,

normalmente con fines más estratégicos y con visión de negocio. Cloud computing ofrece diferentes tipos de soluciones según la aplicación del cliente, SaaS, PaaS e IaaS; estos tres tipos de soluciones son representados en la **Figura 7** según el tipo de aplicación y/o usuario [34].



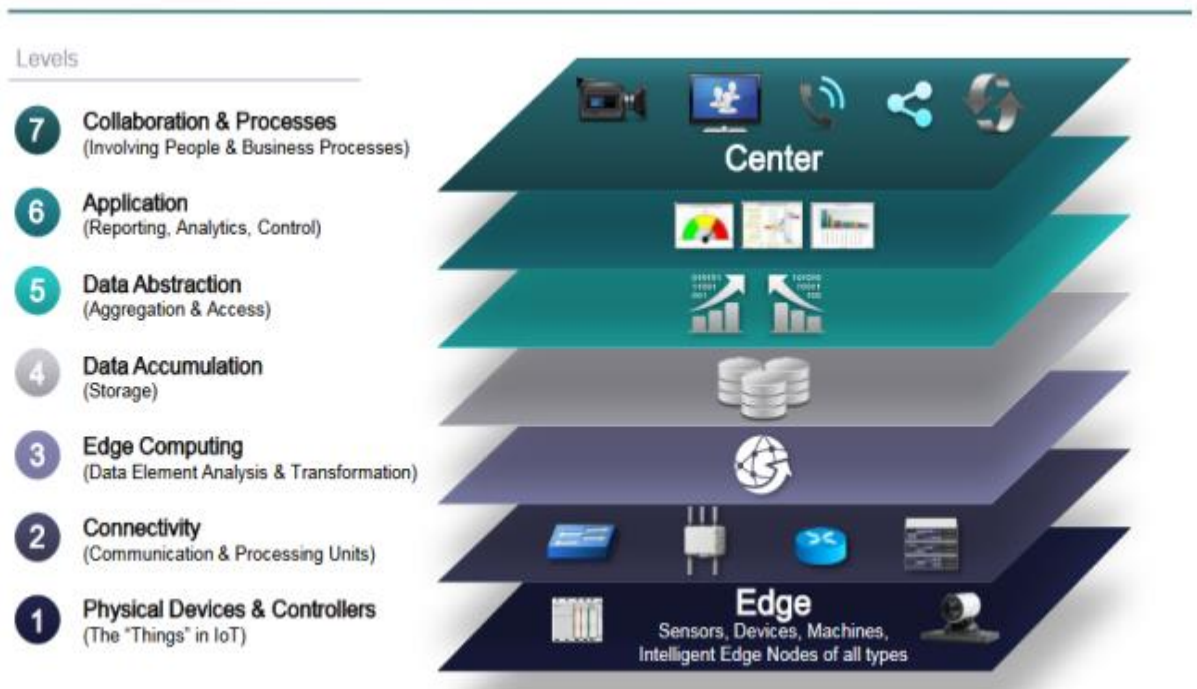
**Figura 7.** Modelos de servicios en la Nube [36]

### 1.3.10 Análisis de Arquitecturas IoT de interés

Una arquitectura IoT debe cumplir ciertos requerimientos para que esta sea viable, los mismos que son conectividad y comunicación, gestión de dispositivos, recolección y procesamiento de datos, escalabilidad y seguridad. A continuación, se citan algunos ejemplos de arquitecturas propuestos por distintas empresas que, a pesar de ser diferentes una con otra, cumplen con todos los requerimientos mencionados [37].

#### 1.3.10.1 Arquitectura con siete capas – Modelo IoTWF

El modelo IoTWF se lo puede visualizar en la **Figura 8**.

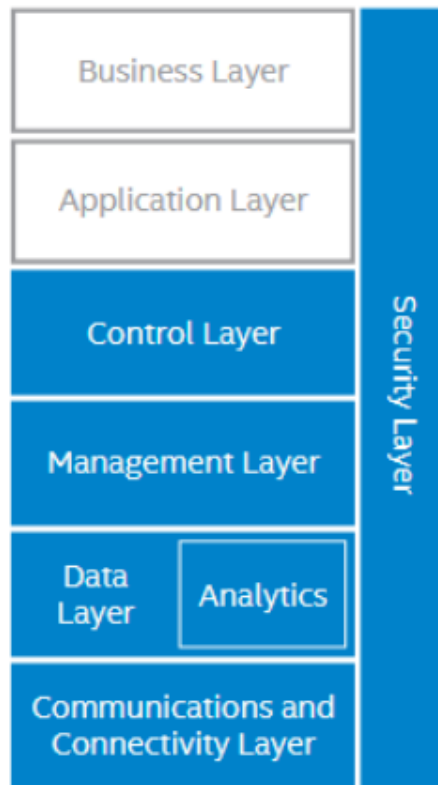


**Figura 8.** Arquitectura IoTWF (siete capas) [38]

Los dispositivos transmiten información ya sea de forma inalámbrica (Gateway IoT) o alámbrica, los datos son tratados y procesados por Edge Computing antes de ser almacenados en la Base de Datos para después ser visualizados por aplicaciones con interfaces gráficas. De esta manera, los datos son expuestos a usuarios específicos para que puedan continuar con procesos posteriores [31].

### 1.3.10.2 Arquitectura con seis capas – Modelo Intel IoT

Este modelo presenta una capa transversal de seguridad ligada a todas las demás capas de la arquitectura. La arquitectura de 6 capas se la representa en la **Figura 9** [31].

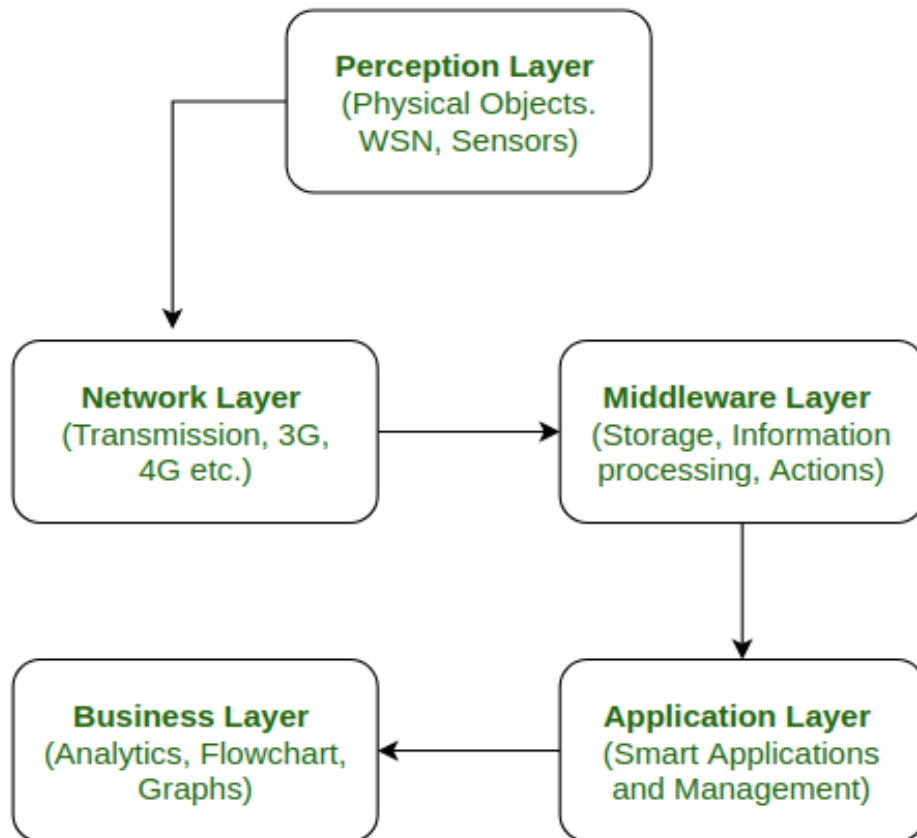


**Figura 9.** Arquitectura IoT Intel (seis capas) [31]

La capa de comunicaciones y conectividad se basa en una conexión mediante una red PAN/LAN o WAN. La capa de análisis de datos es procesada mediante Edge computing. La capa de Gestión supervisa las operaciones de los dispositivos. La capa de control se encarga del gestionar el acceso a los dispositivos previo a la capa de aplicación al usuario [31].

### 1.3.10.3 Arquitectura con cinco capas

Cuando se realiza un proyecto o sistema empleando varias tecnologías innovadoras en un área relativamente amplia, la arquitectura IoT de cinco capas, misma que se muestra en la **Figura 10**, es considerada la mejor opción [39].

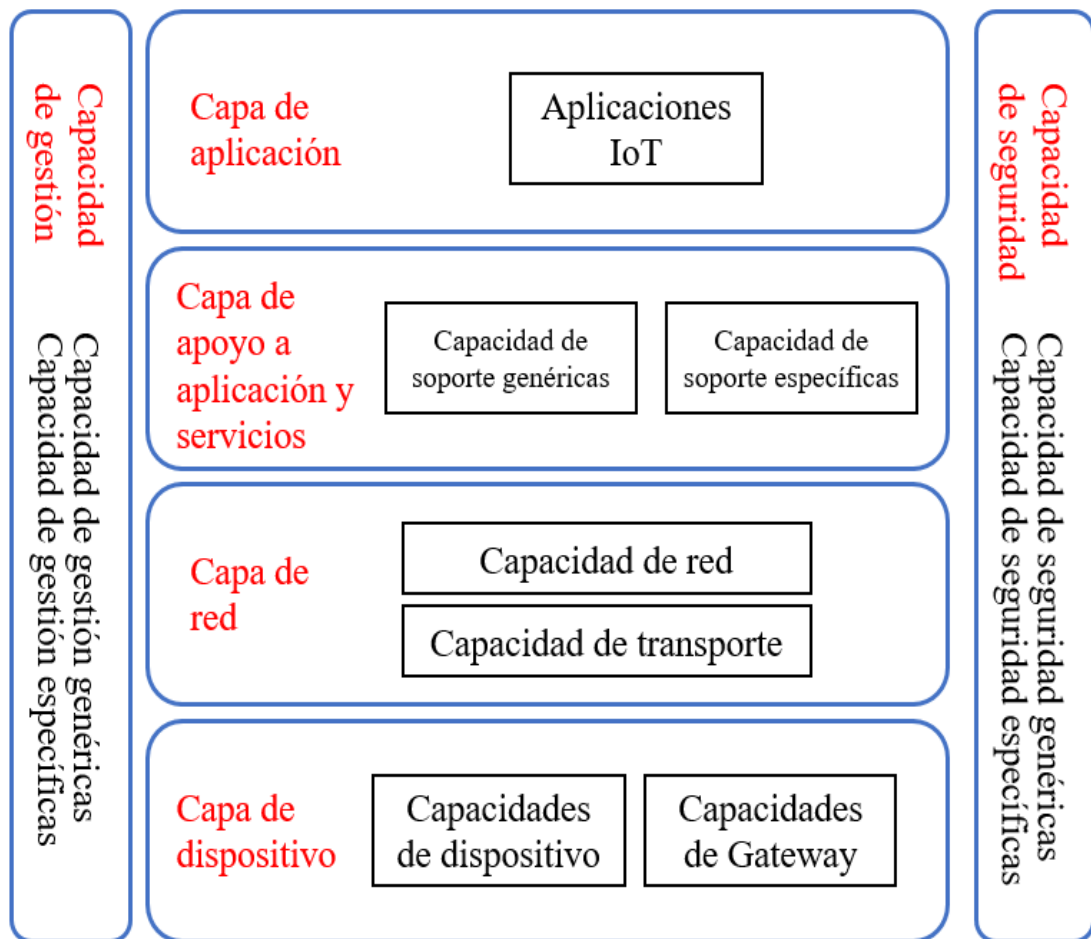


**Figura 10.** Arquitectura IoT de 5 capas [39]

En esta arquitectura, el procesamiento de datos se lo realiza en la capa de middleware, es decir, los datos son tratados por Fog computing [39].

#### 1.3.10.4 Arquitectura con cuatro capas

El modelo de arquitectura propuesto por la ITU, mostrada en la **Figura 11**, consta de cuatro capas, la misma que está formada por dos capas transversales en cuanto a seguridad (privacidad, control, protección y verificación de los datos en todas las capas) y gestión (dispositivos, diagnóstico, actualización de software, topología y tráfico) [31].

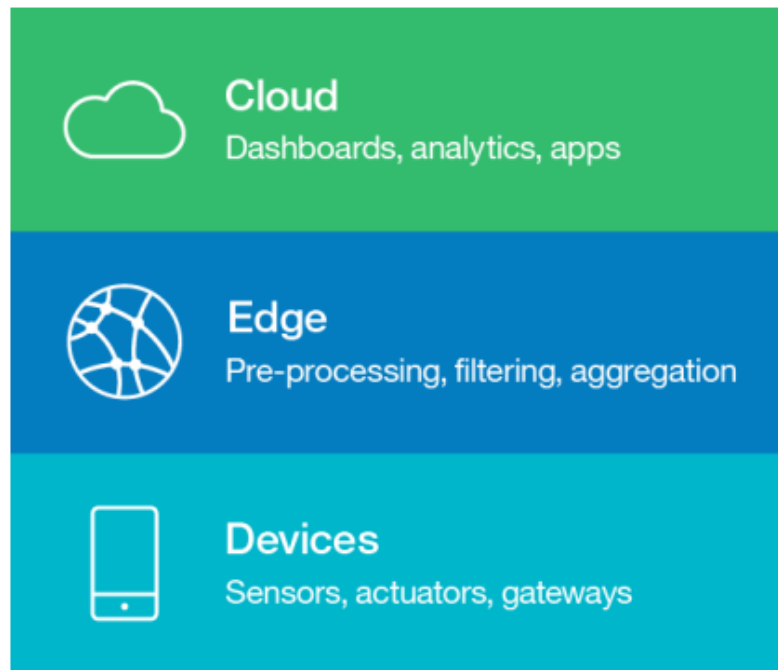


**Figura 11.** Arquitectura IoT de 4 Capas [31]

### 1.3.10.5 Arquitectura con tres capas

Existe un modelo de tres capas propuesto por IBM, representado en la **Figura 12**, el cual está conformado por: la capa de dispositivos, la capa de procesamiento Edge y la capa de nube. Esta arquitectura trata los datos por medio de Edge computing para reducir la latencia al nivel más bajo posible y filtra los datos previo al envío a la nube. Es el modelo de arquitectura más simple de IoT [31].





**Figura 12.** Arquitectura IoT de 3 capas por IBM [40]

### 1.3.11 Sistemas mediante codificación

La codificación es un proceso mediante, el cual, se asignan claves numéricas a las respuestas de otro determinado proceso que se encuentra integrado a una etapa de tratamiento integral de la información, que incluye la captura, validación y explotación de ésta. Su propósito es asignar a cada descripción un código único. En base a este principio, la codificación por producción personalizada consiste en el almacenamiento de un código único generado a un corto tiempo que se descartará al momento de caducidad o ejecución del código, puede ser codificada mediante codificación numérica o alfanumérica según sea la necesidad y la cantidad de datos para manipularse [41].

### 1.3.12 Dispensador Automático de productos

Un dispensador es un contenedor que tiene como propósito traspasar volúmenes de algún producto de un recipiente a otro. Los dispensadores automáticos, son máquinas para la venta de productos de consumo bajo condiciones de higiene y calidad disponibles las 24 horas del día, situados en lugares donde exista mayor circulación de

personas como: hospitales, bares, universidades, entre otros. Estas máquinas se han convertido en un mercado potencial, de desarrollo y de crecimiento, que cada vez revoluciona gracias a la industria 4.0, dando origen a máquinas dispensadoras inteligentes basadas en la tecnología del internet de las cosas y la inteligencia artificial [42].

### **1.3.13 Frutos secos, sus beneficios y como conseguirlos**

Comer frutos secos como parte de una dieta saludable resulta beneficioso para el corazón. Los frutos secos contienen ácidos grasos insaturados y otros nutrientes. Además, son un excelente tentempié, económicos, fáciles de guardar y de empaquetar para cuando las personas se tienen que movilizar de un lugar a otro. Pese a esto, durante un tiempo los frutos secos han tenido mala fama, ya que al ser altamente calóricos muchas personas optaban por eliminarlos por completo de sus dietas, lo cierto es que su consumo es más que recomendable, no solo no hay que desterrarlos por completo de la alimentación, sino que deben ser consumidos a diario por tener grandes beneficios como [43]:

- Cuidan la salud del corazón
- Protegen los huesos y músculos
- Aportan nutrientes para la piel y el cabello
- Son poderosos antioxidantes para nuestras células
- Fuente de energía y grasas saludables
- Favorecen el sistema nervioso y la memoria
- Ayudan a mantener un peso saludable
- Son una potencial fuente de proteína

### **1.3.14 Servidor**

Es un sistema informático que se encarga de almacenar, distribuir y suministrar información. El funcionamiento de los servidores se basa en el modelo cliente-servidor. El cliente puede ser un computador, máquina informática o aplicación que realiza una petición de información al servidor. Por lo tanto, el servidor ofrece la información demandada por el cliente autorizado. Los servidores generalmente nunca se apagan, es decir, se mantienen constantemente brindando sus servicios a los

requerimientos continuos del cliente. Los servidores pueden ser locales o alojado en la nube [44].

#### **1.3.14.1 Servidor Web**

Este tipo de servidor almacena y gestiona el contenido de las páginas web, como archivos php, JavaScript, HTML, CSS, etc. Funciona de la mano con sistemas de gestión de bases de datos, los cuales, en conjunto, proporcionan la información requerida al usuario por medio de un navegador Web. Los servidores web más conocidos son: Apache web server, Nginx, y Microsoft. Los servidores Web pueden ser ofertados en Internet a manera de PaaS [44].

#### **1.3.14.2 Servidor FTP**

Su nombre proviene de Protocolo de Transferencia de Datos (File Transfer Protocol). Este servicio trata de la transferencia de archivos entre un cliente y un servidor, de esta forma, el servidor puede recibir todos los archivos necesarios para su funcionamiento que son enviados por el cliente. Existen un servicio derivado del FTP que es el SFTP, este último mejora la velocidad y seguridad en la transferencia de datos [44].

#### **1.3.15 Sistema Gestor de Base de Datos**

Los sistemas gestores de Bases de Datos permiten la manipulación, creación, gestión y control de los datos organizados en un fichero. Existen dos modelos principales de Bases de Datos [45].

##### **1.3.15.1 Bases de Datos Relacionales (SQL)**

Las bases de datos relacionales o SQL, son aquellas que basan su organización en un modelo relacional, es decir, establecen vínculos entre los datos almacenados. Los datos son presentados al usuario en forma de tablas con sus propios registros y atributos. Este tipo de bases de datos llevan mucho tiempo en el mercado, por lo tanto, disponen

de una gran cantidad de extensiones y soporte. Uno de los sistemas de gestión de bases de datos relacionales más utilizadas es MySQL, desarrollada bajo la licencia de Oracle Corporation, este gestor es muy común en proyectos de desarrollo web. Sin embargo, también existen otros sistemas gestores de bases de datos relaciones como: PostgreSQL, MariaDB, y SQLite [3].

### **1.3.15.2 Bases de Datos Relacionales (NoSQL)**

Las bases de datos no relacionales o NoSQL son aquellas que no basan su organización de datos en un modelo relacional, este tipo de bases presentan una gran facilidad de escalabilidad y flexibilidad debido que sus datos no están centralizados. Debido a que utilizan entornos que se encuentran siempre disponibles y operando, es posible realizar cambios en sus esquemas sin tener que parar las bases. Los sistemas de gestión de bases de datos no relacionales más utilizadas son MongoDB, Cassandra y Redis [3].

### **1.3.16 Conceptos de Desarrollo Web**

Al momento de crear un sitio web, además de contar con un dominio, etiquetas principales e hipervínculos correctos, el desarrollador se encuentra con dos grandes conceptos, el Front End y el Back End, cada una con sus características únicas y distintivas, sin embargo, ambas interactúan entre sí para asegurar el correcto funcionamiento de un sitio web [46].

#### **1.3.16.1 Front-End**

El Front-End se relaciona con toda la capa visible del sitio web, es decir, toda la experiencia visual que brinda el sitio web mediante interfaces gráficas, menús desplegables, íconos adecuados, combinación de colores, animaciones, etc. Que sirven para interactuar con el usuario. El desarrollo Front-End se basa en tres lenguajes fundamentales [46]:

- **HTML**, encargado de formar las estructuras base del sitio web, es decir, el esqueleto.

- **CSS**, encargado de los estilos correspondientes a cada elemento creado en el esqueleto HTML, como son, animaciones, colores, responsividad, posicionamiento, tamaños, etc.
- **JavaScript**, encargado de la interactividad con la información ante un determinado evento, por ejemplo, el despliegue de una tabla de datos cada cierto tiempo.

Dentro del mundo del Front-End, también existen los Frameworks, que no son más que estructuras o plantillas HTML, CSS y JavaScript previas ya armadas por terceros que pueden servir de base para la realización de un proyecto [47].

Los frameworks más comunes dentro del desarrollo Front-End son [46]:

- Bootstrap
- Foundation
- Angular JS
- Ember JS
- Backbone
- React JS

### **1.3.16.2 Back-End**

El Back-End es la capa de sitio web al cual los usuarios no tienen acceso, por ejemplo, archivos de conexión con bases de datos, rutas de almacenamiento del sitio, consultas de datos, codificación y decodificación de información, ciberseguridad, etc. Los lenguajes más comunes dentro del desarrollo Back-End son [46]:

- PHP
- Python
- Java
- .Net
- Ruby
- Node

## **1.4 Objetivos**

### **Objetivo General**

Desarrollar un dispensador inteligente de frutos secos basado en arquitectura IoT para producción personalizada orientado a la Industria 4.0 para la empresa Lula Organic.

### **Objetivos Específicos**

- Analizar la producción personalizada de alimentos a granel y su arquitectura tecnológica en aplicaciones de Industria 4.0.
- Determinar las características de hardware y software para el dispensador de frutos secos aplicado a la producción personalizada.
- Seleccionar los sistemas de comunicación adecuados para la interacción entre los elementos de la arquitectura IoT y el dispensador.
- Diseñar una interfaz gráfica para que el usuario realice el pedido personalizado del producto requerido.

## CAPÍTULO II

### METODOLOGÍA

#### 2.1 Materiales

En la implementación del presente proyecto, se emplearon materiales como: Cuatro dispensadores de plástico, cada uno corresponde a un tamaño distinto de frutos secos; leds de alta luminosidad de 5 mm que sirven para identificar los distintos estados de cada dispensador; sensores magnéticos de efecto Hall para la detectar si un dispensador está abierto o cerrado; sensores ultrasónicos HC-SR4 que sirven para medir la distancia entre la cubierta de cada dispensador y el producto interno almacenado en cada uno, de esta manera, se determina cuanto producto está presente en cada contenedor, tanto en porcentaje, como en gramos mediante conversiones adecuadas por programación; pantallas LCD 16x2 con Bus I2C para la visualización de datos en tiempo real; una celda de carga de 5 Kg, con su respectivo controlador HX711 que sirve para el pesaje de los frutos secos. En cuanto a los mecanismos de dispensación, se utilizaron piezas con medidas exactas que fueron producidas por impresiones en 3D, cada contenedor tiene un mecanismo distinto que asegura su correcta dispensación acorde al tamaño de grano almacenado; también se utiliza servomotores MG995 de giro continuo de 360°, estos tienen un alto torque que oscila de 8.5 Kg/Cm a 11 Kg/Cm dependiendo de su voltaje de alimentación, los servos de alto torque aseguran el giro de los mecanismos de dispensación en condiciones normales y extremas. Acorde al ensamblaje general, se utilizaron materiales como estructuras de madera, láminas de lata para resbalar el producto dispensado hacia un vaso biodegradable y piezas impresas en 3D para adaptar los requerimientos del prototipo a piezas con medidas exactas; posterior a esto, se utiliza dos tarjetas de desarrollo en la etapa de control: el Arduino MEGA 2560 que posibilita la comunicación con todos los componentes electrónicos gracias a su gran cantidad de pines, y la NODEMCU ESP8266 que proporciona conexión a un router WiFi para el envío de datos un servidor mediante el protocolo HTTP, el mismo que tiene instalado un sistema gestor de bases de datos MySQL, donde además, se guardarán los registros de usuarios, pedidos y ventas. A continuación, se utiliza una plantilla desarrollada en el Framework Bootstrap, la misma que fue adaptada a las necesidades Front-End del proyecto; dentro de la plantilla se incluyeron gráficos dinámicos que brinda la librería Highcharts; en cuanto al Back-End, se utilizó PHP

para realizar todas las gestiones con el sistema de Base de datos MySQL en conjunto con JavaScript para poder graficar los datos en tiempo real sin necesidad de recargar la página.

## **2.2 Métodos**

### **2.2.1 Modalidad de Investigación**

#### **Investigación Aplicada**

Para el desarrollo del presente proyecto se trabajó bajo una investigación aplicada, porque se emplearon los conocimientos adquiridos en la formación académica para solucionar los problemas de exigencia del consumidor final y los precios de producción mediante la dosificación de frutos secos a una escala personalizada que satisface las necesidades latentes de la demanda actual.

#### **Investigación Bibliográfica**

Se realizó una investigación bibliográfica para sustentar el proyecto mediante la recopilación de información de revistas técnicas, libros, artículos científicos, publicaciones en internet y tesis relacionados a los sistemas de dosificación, industrias 4.0, desarrollo web, plataformas IoT, personalización en la industria 4.0, oferta y demanda de acuerdo con la exigencia del consumidor final.

#### **Investigación de Campo**

Se realizó un proceso de estudio de campo en empresas que se dedican a la venta y distribución de frutos secos y productos a granel en general, principalmente en la empresa Lula Organic, que permitió la recopilación de información clave para solventar las exigencias del consumidor final en cuanto a personalización y brindar un valor agregado en sus servicios haciendo uso de la tecnología disponible en el mercado.

### **2.2.2 Recolección de Información**

Para la recolección de información, se realizó un profundo análisis y procesamiento de fuentes bibliográficas como libros, revistas, repositorios digitales universitarios, publicaciones en Internet y proyectos desarrollados, así como guías prácticas y



manuales de construcción, por lo que, se tomó en cuenta bases de datos confiables que permitieron el desarrollo del proyecto.

### 2.2.3 Procesamiento y Análisis de Datos

Para el procesamiento y análisis de datos se realizaron los siguientes pasos:

- Revisión y filtración de la información recopilada.
- Análisis de las características que oferta la Industria 4.0
- Estudio de las propuestas de solución planteadas para aumentar la productividad de acuerdo con las necesidades del emprendedor o microempresario, aumentando productividad y reduciendo precios.
- Planteamiento de la propuesta de solución

### 2.2.4 Desarrollo del Proyecto

Para el desarrollo del dispensador inteligente de frutos secos, se plantearon actividades mostradas en las **Tablas 1, 2 y 3**, basadas en los objetivos específicos del proyecto.

**Tabla 1:** Actividades para el Desarrollo del Proyecto (Parte 1/3)

|  |  |
|--|--|
| <b>Tema:</b> Dispensador inteligente de frutos secos basado en arquitectura IoT para producción personalizada orientado a la Industria 4.0 para la empresa Lula Organic.                           |  |
| <b>Objetivo General:</b> Desarrollar un dispensador inteligente de frutos secos basado en arquitectura IoT para producción personalizada orientado a la Industria 4.0 para la empresa Lula Organic |  |
| <b>Objetivo Específico</b>   | <b>Actividades</b>   |
| Analizar la producción personalizada de alimentos a granel y su arquitectura tecnológica en aplicaciones de Industria 4.0  | <ul style="list-style-type: none"> <li>✓ Determinación de las exigencias del mercado en cuanto a precio vs calidad.</li> <li>✓ Análisis de tipos de personalización en un determinado producto de la empresa.</li> <li>✓ Determinación de los sistemas adaptables a la industria 4.0.</li> </ul> |

**Elaborado por:** El Investigador

**Tabla 2.** Actividades para el Desarrollo del Proyecto (Parte 2/3)

| Objetivo Específico  | Actividades   |
|--|---|
| <p>Determinar las características de hardware y software para el dispensador de frutos secos aplicado a la producción personalizada.</p>   | <ul style="list-style-type: none"> <li>✓ Determinación de dispositivos de interacción hardware y software y la industria 4.0.</li> <li>✓ Análisis de los requerimientos que necesita el sistema de dispensador.</li> <li>✓ Determinación de los materiales necesarios y adecuados para el desarrollo del sistema.</li> </ul>  |
| <p>Seleccionar los sistemas de comunicación adecuados para la interacción entre los elementos de la arquitectura IoT y el dispensador.</p> | <ul style="list-style-type: none"> <li>✓ Bosquejo de medios de comunicación inalámbrica IoT adaptables al sistema.</li> <li>✓ Diseño de un dispensador de frutos secos con parámetros adecuados que garanticen su producción.</li> <li>✓ Simulación de un Sistema de dispensador con el uso de tecnología inalámbrica.</li> <li>✓ Diseño del medio de comunicación entre usuario y el sistema de dispensador.</li> <li>✓ Desarrollo del medio de conexión IoT entre usuario y dispensador.</li> <li>✓ Integración de los elementos de la Arquitectura IoT en el dispensador automático</li> </ul> |

**Elaborado por:** El Investigador

**Tabla 3.** Actividades para el Desarrollo del Proyecto (Parte 3/3)

| <b>Objetivo Específico</b>   | <b>Actividades</b>  |
|--|---|
| Diseñar una interfaz gráfica para que el usuario realice el pedido personalizado del producto requerido. | <ul style="list-style-type: none"><li>✓ Programación de la interfaz gráfica usando herramientas de desarrollo web.</li><li>✓ Ejecución de pruebas, detección y corrección de errores del sistema.</li><li>✓ Análisis de resultados y la obtención de conclusiones del sistema en funcionamiento.</li><li>✓ Elaboración del informe final de la propuesta.</li></ul> |

**Elaborado por:** El Investigador

## **CAPÍTULO III**

### **RESULTADOS Y DISCUSIONES**

#### **3.1 Análisis y Discusión de los resultados**

En base al análisis realizado referente a los antecedentes investigativos de diferentes autores, se obtiene que se han desarrollado dispensadores automatizados principalmente de comida para mascotas, y dispensadores de pastillas o medicamentos, sin embargo, los proyectos enfocados a industria 4.0 no incluyen dispensadores automatizados, más bien, incluyen únicamente sistemas de control industrial enfocados a robots o maquinaria pesada.

En el mercado actual, se tiene únicamente dispensadores mecánicos de frutos secos en oferta, los únicos dispensadores micro-controlados que se puede encontrar en el mercado son de comida para mascotas, más no uno para frutos secos y mucho menos controlado o gestionado desde una interfaz Web con enfoque de industria 4.0. Además, actualmente, varios de los negocios dedicados a la venta de alimentos a granel en la ciudad de Ambato, no ofrecen una venta personalizada de sus productos, ya que, venden únicamente empaques estandarizados teniendo como ½ libra el producto con menor cantidad que pueden vender, es decir, el cliente no puede consumir pequeñas cantidades en gramos de distintos frutos secos, a menos que compre empaques completos de cada producto que desea consumir, lo que llevaría a pagar un precio elevado o la pérdida de la venta que solicita el cliente. Tomando en cuenta todos los datos recolectados, se desarrolla un dispensador que satisface las necesidades del cliente previamente mencionadas, ocupando únicamente materiales de fácil asequibilidad en el mercado actual.

#### **3.2 Desarrollo de la propuesta**

El presente proyecto de investigación se basa en el desarrollo de un dispensador automático de frutos secos que cumpla y satisfaga las diferentes demandas de personalización del cliente, y al mismo tiempo, beneficie a la empresa Lula Organic.

Para que el dispensador pueda adaptarse a las diferentes situaciones que pueden presentarse al momento de dispensar distintos tipos de frutos secos, se adaptan 4 dispensadores mecánicos, disponibles en el mercado, con piezas diseñadas e impresas en 3D y ancladas a dispositivos giratorios de alto torque, las cuales, mediante previo proceso de múltiples pruebas y errores, cumplen con la correcta dispensación de 3 categorías de frutos secos divididas por su tamaño, las cuales son: grano de tamaño fino, grano de tamaño medio y grano grueso. Cada contenedor está ensamblado con un mecanismo distinto que corresponde a cada categoría de granos respectivamente, siendo el cuarto contenedor una réplica del tercer contenedor, es decir, los dos últimos contenedores poseen el mismo mecanismo de dispensación, correspondiendo este al grano grueso, esto a manera de ofertar cuatro tipos de frutos secos distintos en el prototipo.

El prototipo completo es ensamblado en una caja de MDF resistente, que es diseñada y armada con medidas exactas acorde a los elementos empleados para el proyecto, internamente posee una tolva desarrollada en acero inoxidable que se encarga de guiar todos los frutos secos dispensados a un único envase que esta encima de una celda de carga.

El dispensador completo interactúa con una base de datos alojada en un servidor. El sistema se diseña para que pueda ser manejado por dos tipos de usuarios: administradores y clientes, mismos que pueden acceder a las interfaces web respectivas desde una misma página de Inicio de Sesión, éste dependiendo de las credenciales ingresadas, direccionará al usuario a la interfaz web que corresponda. La interfaz web de usuarios administradores ofrece un estricto control y monitoreo de todas las variables registradas del sistema, por ejemplo: cantidad de pedidos, clientes, administradores, venta total, estado en tiempo real de los contenedores, ordenamiento de datos en múltiples formas, etc. Datos que sirven para la toma de decisiones en beneficio de la empresa. Por su parte, la interfaz web de usuarios clientes ofrece una pequeña visualización del estado actual de los contenedores en conjunto con una presentación comercial y agradable de los cuatro tipos de frutos secos, finalmente el cliente puede escoger de una forma amigable, la cantidad en gramos de los frutos secos que desea comprar junto con la información del precio total de su pedido; el cliente

también puede visualizar datos de interés en su perfil, como su historial de pedidos y sus credenciales.

La funcionalidad del sistema es monitorear los contenedores, recibir y dispensar los pedidos registrados por los clientes en la base de datos. De esta manera, los clientes no tienen que pagar precios elevados por consumir distintos tipos de frutos secos en una misma compra, ya que, consumen y pagan solo por la cantidad exacta deseada. Esto evita el paradigma de comprar empaques de frutos secos con pesos mínimos estandarizados. En caso de tener múltiples pedidos, el sistema se diseñó para dispensarlos de forma ordenada, además, una vez recibido el pedido al prototipo, éste genera una alerta por medio de pantallas y leds al propietario de la empresa, indicándole que existe un pedido nuevo en espera, pero este no será dispensado hasta que el propietario presione el botón de confirmación, de esta manera, se tiene un mayor control en caso de la cancelación de pedidos. Debido a que el prototipo estará ubicado en locales comerciales, se contará con el respectivo abastecimiento de energía requerida por el sistema, que serán dos fuentes DC, una para los microcontroladores y la segunda para los dispositivos giratorios de alto torque.

### **3.2.1 Análisis de Factibilidad**

A continuación, se detallan tres estudios de factibilidad que permiten obtener información relevante para el desarrollo del presente proyecto.

#### **✓ Factibilidad Técnica**

El proyecto presenta una factibilidad técnica, debido a que se cuenta con los conocimientos necesarios para seleccionar la tecnología requerida para el mismo, como sensores, actuadores y manejo adecuado de diferentes softwares. Además, se dispone de facilidad para obtener todas las herramientas de software y hardware que requiere el proyecto, ya que, todas ellas se ofertan en el mercado nacional e internacional.

#### **✓ Factibilidad Económica**

El proyecto presenta una factibilidad económica, debido a que no se necesita pagar ninguna licencia por el uso de ningún componente de hardware o software utilizado, adicionalmente se desarrolla el proyecto en un servidor local, el cual no requiere de ningún pago. Finalmente, los materiales empleados

para la construcción del proyecto fueron financiados en su totalidad por el investigador.

✓ **Factibilidad Bibliográfica**

El presente proyecto cuenta con una factibilidad Bibliográfica, debido al fácil acceso que se dispone a información referenciada en libros, tesis, revistas y páginas web sobre la tecnología utilizada, además en Internet están alojados múltiples sitios con ejemplos prácticos y foros relacionados a la programación web necesaria para el desarrollo del proyecto.

### 3.2.2 Arquitectura IoT propuesta para el Dispensador

La arquitectura IoT propuesta para el presente proyecto es de tres capas, basado en el modelo de IBM, con la diferencia de que se propone un procesamiento de datos distinto al referencial. La arquitectura está conformada por: la capa de dispositivos, la capa de procesamiento Fog y la capa Nube. Esta arquitectura trata los datos por medio del cómputo de niebla o Fog computing, es decir, los datos son procesados en un punto intermedio entre la sensorización y la nube, este punto es el microcontrolador, de esta manera se evita la sobrecarga de datos innecesarios en el servidor y se reduce la latencia del sistema. En la **Figura 13** se representa la arquitectura propuesta para el dispensador.



**Figura 13.** Arquitectura IoT Propuesta

**Elaborado por:** El Investigador

### 3.2.3 Consideraciones Técnicas

#### Selección de Sensores

Se realiza a continuación, una comparación técnica entre los sensores ultrasónicos más comunes en el mercado, ya que se requiere de un sensor de distancia para poder medir constantemente la cantidad de producto que está presente en cada contenedor de frutos secos, los parámetros técnicos de los sensores ultrasónicos se detallan en la **Tabla 4**.

**Tabla 4.** Parámetros de Sensores Ultrasónicos

|  | <b>JSN-SR04T</b>  | <b>HC-SR04</b>   | <b>A02YYUW</b>  |
|--|---|--|---|
|  |  |  |  |
| <b>Rango de Distancia</b>                    | 20 cm – 600 cm  | 2 cm – 400 cm  | 3 cm – 450 cm   |
| <b>Precisión</b>                             | +/- 1 cm  | +/- 0.3 cm   | +/- 0.2 cm  |
| <b>Datos Seriales</b>                        | SI  | SI   | NO  |
| <b>Disponibilidad en el mercado Nacional</b> | Alta  | Alta   | Muy escasa  |
| <b>Precio</b>                                | \$17.00   | \$2.00   | \$50.00   |




**Elaborado por:** El Investigador

Se selecciona el sensor ultrasónico **HC-SR04**, debido a que su rango de medición está entre 2 y 400 cm, siendo 2 cm el caso en el que el contenedor este lleno al 100%, distancia adecuada para abarcar con mayor producto posible a cada contenedor en comparación con los otros sensores ultrasónicos, además de su bajo precio y alta disponibilidad en el mercado nacional.



En cuanto al requerimiento de detectar si un contenedor está o no abierto, se pretende usar sensores magnéticos o Infrarrojos, siempre y cuando estos presenten una salida digital que asegure la presencia o ausencia del objeto a detectar. Sin embargo, los sensores infrarrojos ofertados en el mercado nacional de bajo costo interactúan con su salida digital detectando únicamente superficies blancas o negras, lo que podría causar confusión en el sensor debido a las variantes condiciones de luz que se pueden presentar, esto puede arrojar una falsa detección de un objeto. Por lo tanto, se utilizará sensores magnéticos o también llamados de efecto Hall, éstos se activarán únicamente si detectan la presencia de un imán, con eso se evita falsas detecciones en comparación con los sensores infrarrojos. En la **Tabla 5** se detalla una comparación técnica entre los distintos sensores de efecto hall del mercado.

**Tabla 5.** Parámetros de Sensores Magnéticos

|  | <b>KY-035</b>   | <b>KY-024</b>   | <b>M44</b>  |
|--|---|---|---|
|  |  |  |  |
| <b>Voltaje de Alimentación</b>               | 3.3V a 5 V DC   | 5V DC   | 5V DC   |
| <b>Salida Analógica</b>                      | SI  | SI  | SI  |
| <b>Salida Digital</b>                        | NO  | SI  | SI  |
| <b>Dimensiones</b>                           | 18 x 15 mm  | 4,0x3,2 cm  | 36 x 16 mm  |
| <b>Disponibilidad en el mercado Nacional</b> | Baja  | Alta  | Alta  |
| <b>Precio</b>                                | \$1.00  | \$2.50  | \$2.50  |

**Elaborado por:** El Investigador

En vista que, el sensor KY-035 no dispone de una salida digital se descarta, quedando los sensores **KY-024** y **M44** como los más adecuados para usar en el proyecto, se

seleccionan ambos sensores y se podrá utilizar cualquiera de los dos según la disponibilidad en el mercado, ya que ambos cumplen con los mismos requisitos, y además, tienen exactamente el mismo precio.

En cuanto al pesaje de los frutos secos dispensados, se pretende usar una celda de carga ensamblada de fábrica con su propio controlador, con el fin de que los datos sean acondicionados de forma lineal y estables para ser detectados por una entrada analógica del microcontrolador principal del sistema. En el mercado existen celdas de carga con capacidades máximas desde 1 kg hasta modelos industriales de 500 kg. Sin embargo, los precios de los modelos no industriales son muy similares, y varían entre los \$13.00 hasta los \$19.00 incluyendo su propio controlador hx711 que puede estar o no armado. Por su parte, la única opción que oferta un dispositivo ensamblado de fábrica con el controlador hx711 junto con una doble base acrílica circular perfectamente adaptada, es la celda de carga de 5 kg mostrada en la **Figura 14**, siendo la opción más adecuada e ideal para el presente proyecto con un precio de \$18.00.








**Figura 14.** Celda de Carga de 5 Kg

**Elaborado por:** El Investigador

### **Selección de Actuadores**

Los actuadores junto con los mecanismos impresos en 3D se encargan del paso de los frutos secos hacia la celda de carga, para lo cual, en la **Tabla 6**, se realiza una comparación técnica entre dispositivos de giro continuo de 360 grados disponibles en el mercado para poder seleccionar el más adecuado.

**Tabla 6.** Parámetros de Dispositivos Giratorios

| Modelo                | Gráfico   | Torque          | Alimentación | Conexión a controlador requerida   | Costo   |
|-----------------------|---|-----------------|--------------|--|---------|
| Motor a pasos Nema 17 |    | 5 Kg/cm         | 12V / 1.5A   | SI<br>A4988<br> | \$18    |
| Servo SG – 90 (360°)  |   | 1 - 1.5 Kg/cm   | 3 – 7.2 V    | NO   | \$3.50  |
| Servo MG – 90S (360°) |  | 1.8 - 2.2 Kg/cm | 4.8 V a 6 V  | NO   | \$8.00  |
| Servo MG – 995 (360°) |  | 8.5 – 11 Kg/cm  | 4.8 – 7.2V   | NO   | \$10.00 |




**Elaborado por:** El Investigador

Se selecciona el servomotor **MG-995** de giro continuo de 360 grados, debido a que cuenta con un alto torque a un nivel de alimentación relativamente bajo, además no necesita un controlador externo para poder operar. El torque de este servomotor asegura una correcta dispensación de frutos secos en condiciones normales y extremas. Cabe recalcar que la corriente que necesita el servomotor mencionado se encuentra en el rango de **500 a 900 mA**.

## Selección del controlador

Para la selección del microcontrolador, se debe tomar en consideración algunos factores importantes, como la accesibilidad que se tiene en el mercado, la disponibilidad de pines analógicos y digitales suficientes para conectar todos los elementos, la simplicidad y compatibilidad de programación con los sensores y actuadores, y finalmente, la disponibilidad de conexión WiFi. En la **Tabla 7**, se detalla los parámetros más importantes de las tarjetas más comunes y asequibles del mercado.

**Tabla 7.** Parámetros del controlador

|                                | <b>Arduino MEGA</b>   | <b>NODEMCU v2</b>  | <b>Raspberry Pi 3 B</b>  |
|--------------------------------|---|--|--|
|                                |  |  |  |
| <b>Alimentación</b>            | 5 – 12 V  | 5 – 10 V   | 5V   |
| <b>Pines Digitales Usables</b> | 54 pines  | 11 pines   | 25 GPIO  |
| <b>Pines Analógicos</b>        | 16 entradas   | 1 entrada  | NO DISPONE   |
| <b>Conexión Inalámbrica</b>    | NO  | SI   | Si   |
| <b>Puerto PWM</b>              | 15 pines de los 54 proveen PWM  | 5 pines  | 2 Pines  |
| <b>Corriente en cada Pin</b>   | 40 mA   | 12 mA  | 50mA   |
| <b>Puertos UART</b>            | 3 Puertos   | 1 Puerto   | 1 Puerto   |
| <b>Precio</b>                  | \$18.00   | \$8.00   | \$80.00  |

**Elaborado por:** El Investigador

Después de haber comparado los parámetros entre los 3 controladores más asequibles en el mercado, se destaca que, la Raspberry Pi3 B es la primera en ser descartada, a pesar de que dispone conexión inalámbrica, no posee entradas Analógicas, las cuales son muy necesarias para poder conectar la celda de carga, además su costo es excesivamente elevado en comparación con los otros dos controladores. Por otra parte, el Arduino Mega cumple con la cantidad de entradas y salidas analógicas suficientes para conectar todos los elementos del proyecto, y aunque no dispone de conexión inalámbrica, este puede comunicarse con la NODEMCU por medio de uno de sus puertos de comunicación UART, de esta manera, se satisface el requerimiento del proyecto de disponer de una conexión inalámbrica, cualidad que la NODEMCU si la posee. Por lo tanto, los controladores seleccionados son el **Arduino Mega** junto con la **NODEMCU v2**.

### **Fuentes de alimentación**

Para evitar fallos por falta de corriente en el presente proyecto, se usan dos fuentes DC independientes. Una fuente de 12 voltios DC que alimenta a los microcontroladores Arduino Mega y NodeMCU, mientras que, una segunda fuente de 6 voltios DC de 1 Amperio que alimenta a los servomotores. Debido a que, los servomotores no se activarán todos simultáneamente, con una fuente general para todos los servos será suficiente para abastecer con la corriente requerida por estos. Las fuentes de 12 y 6 voltios se encuentran con facilidad en el mercado, y están representados en la **Figura 15**.



**Figura 15.** Fuentes DC empleadas

**Elaborado por:** El Investigador

## Selección de la Base de Datos

La selección de la base de datos se fundamenta principalmente en la experiencia del investigador usando MySQL por medio del gestor PhpMyAdmin a lo largo de la carrera universitaria. PhpMyAdmin es una aplicación Web basada en PHP que presenta una interfaz amigable al usuario y permite administrar y gestionar las bases de datos. Por medio del mencionado gestor, se pueden realizar todo tipo de operaciones como crear, borrar y actualizar datos o tablas, además, se puede crear usuarios con privilegios asignados y ejecutar sentencias SQL, todo desde una misma interfaz. Varios hostings gratuitos y de pago en el mercado poseen PhpMyAdmin.

### 3.2.4 Mecanismos de Dispensación

Previo al diseño y construcción de las piezas impresas en 3D, se tomó en consideración tres tamaños distintos de frutos secos, los cuales fueron categorizados de la siguiente manera:

- ✓ **Granos de tamaño fino:** Azúcar y harinas derivadas de frutos secos, semillas pequeñas, etc. El producto seleccionado de esta categoría para el contenedor # 1 fue semillas de sésamo (ajonjolí), y se lo representa en la **Figura 16**.



**Figura 16.** Semillas de Sésamo (Ajonjolí) [48]

- ✓ **Granos de tamaño medio:** Semillas relativamente más grandes, maní desgranado, piñones naturales, etc. El producto seleccionado de esta categoría para el contenedor # 2 fue maní desgranado, y se lo representa en la **Figura 17**.



**Figura 17.** Maní [49]

- ✓ **Granos de tamaño grueso:** En esta categoría se incluye los frutos secos con mayor tamaño con respecto a las dos categorías anteriores, por ejemplo: cacahuates, nuez, anacardos, almendras, pistachos, nuez pecana, avellanas, etc. Los productos seleccionados de esta categoría para los contenedores #3 y #4 fueron pistachos y nueces respectivamente, y se muestran en la **Figura 18**.

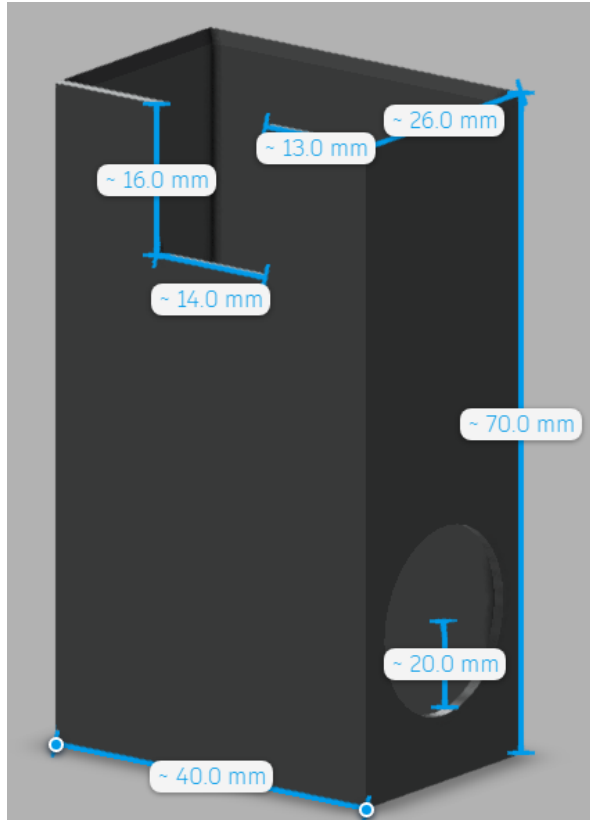


**Figura 18.** Pistacho y Nuez [50] y [51]

La construcción de los mecanismos de dispensación se los realizó mediante un previo diseño 3D en el software Fusion 360 de Autodesk con licencia gratuita para estudiantes. Cada mecanismo tiene medidas distintas acorde a los contenedores base que fueron adquiridos en el mercado, además, cada mecanismo está anclado a dos servomotores Mg-995 que sirven para controlar la caída de cada tipo de grano de una manera adecuada, misma que se la consiguió a través de múltiples pruebas y errores hasta llegar al modelado 3D que cumpla todos los requisitos mecánicos.

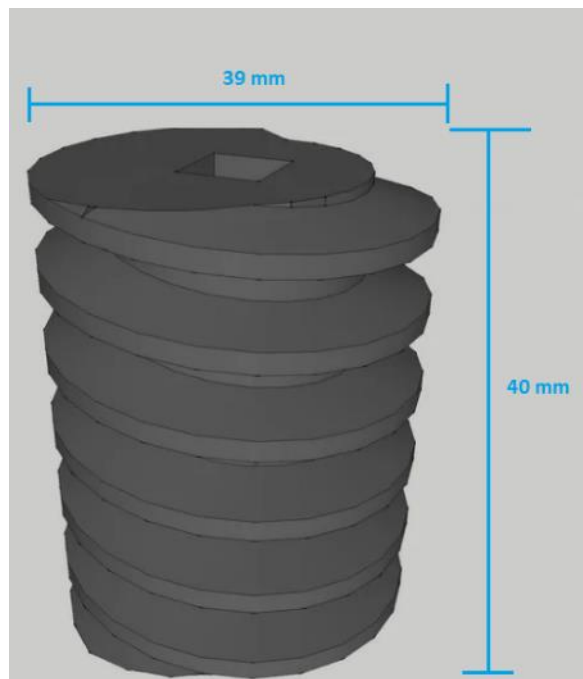
#### **3.2.4.1 Mecanismo de Dispensación para el Contenedor #1**

Las **Figuras 19** a la **26** representan el modelado 3D de las piezas necesarias para el mecanismo de dispensación del contenedor # 1, categorizado como grano fino.



**Figura 19.** Colector primario

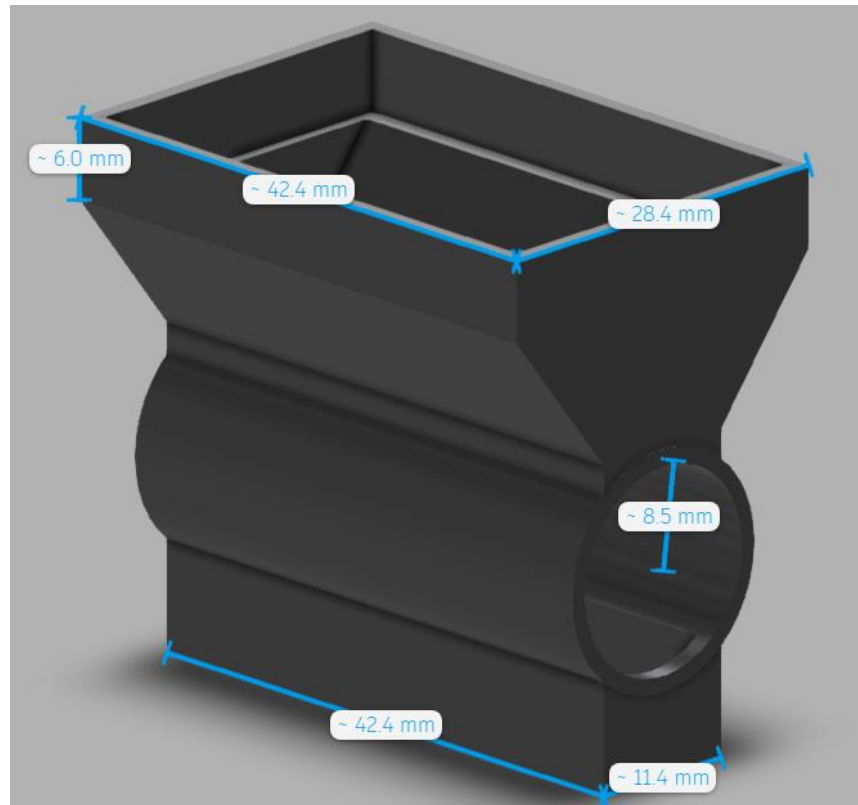
**Elaborado por:** El Investigador



**Figura 20.** Pieza Giratoria A

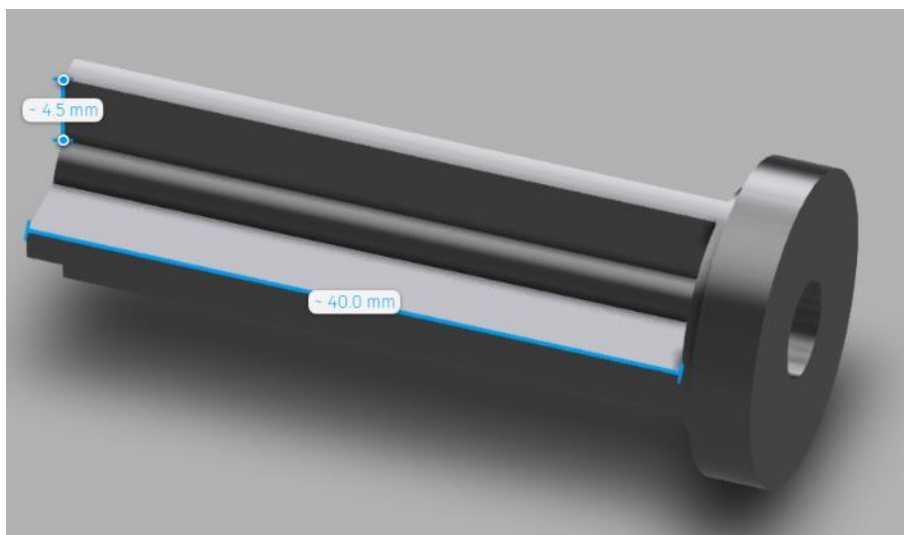
**Elaborado por:** El Investigador





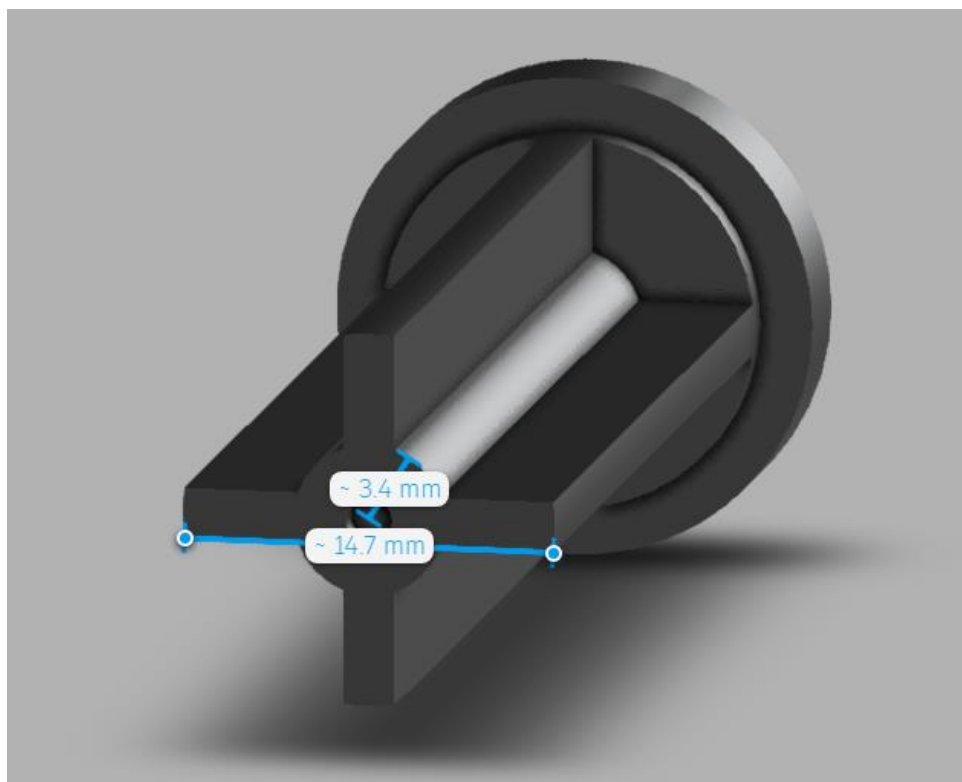
**Figura 21.** Colector Final

**Elaborado por:** El Investigador



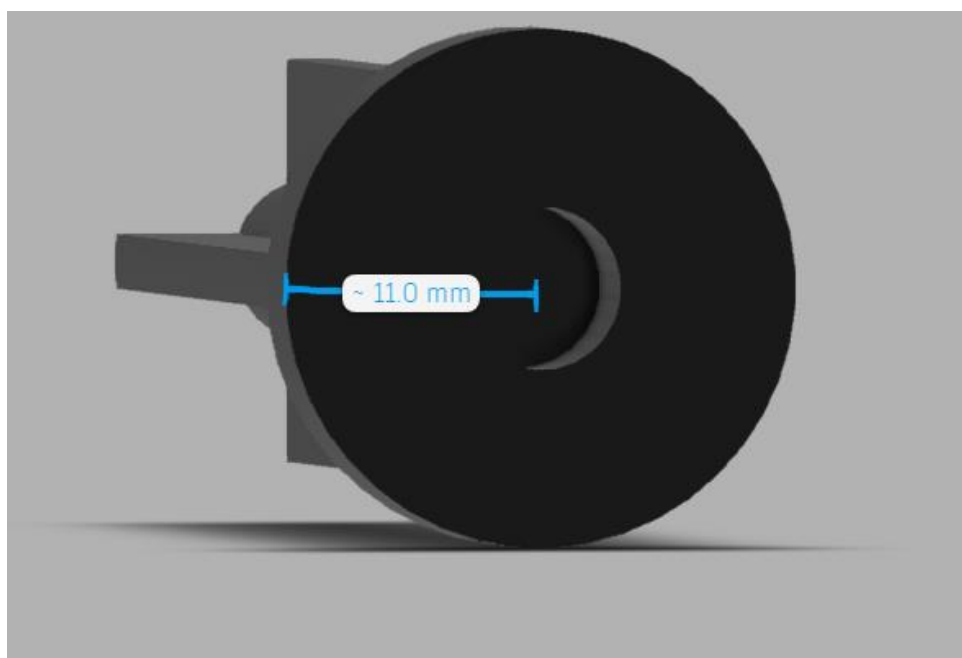
**Figura 22.** Pieza Giratoria B – Vista 1

**Elaborado por:** El Investigador



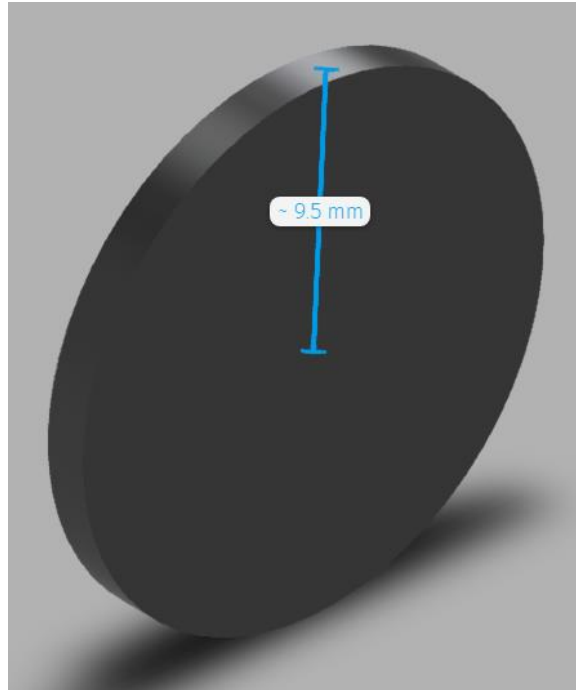
**Figura 23.** Pieza Giratoria B – Vista 2

**Elaborado por:** El Investigador



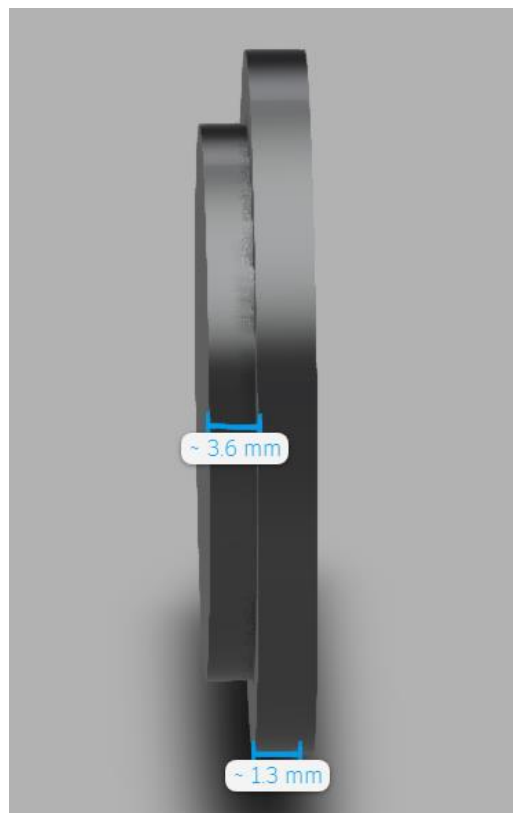
**Figura 24.** Pieza Giratoria B – Vista 3

**Elaborado por:** El Investigador



**Figura 25.** Tapa de mecanismos giratorios – Vista 1

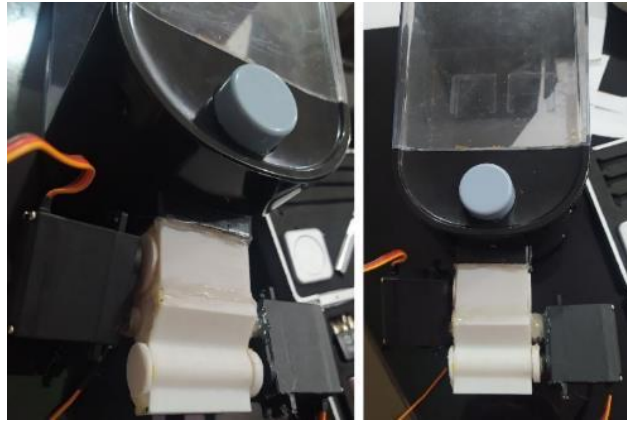
**Elaborado por:** El Investigador



**Figura 26.** Tapa de mecanismos giratorios – Vista 2

**Elaborado por:** El Investigador

En la **Figura 27**, se presenta el mecanismo de dispensación para el contenedor #1 ensamblado por completo.

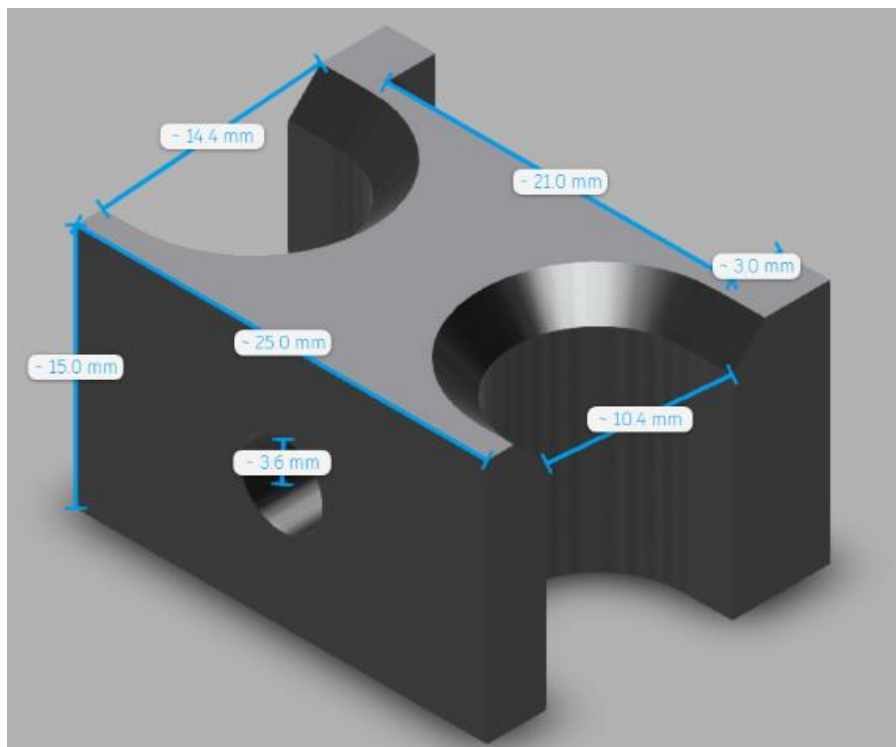


**Figura 27.** Mecanismo del contenedor # 1 ensamblado

**Elaborado por:** El Investigador

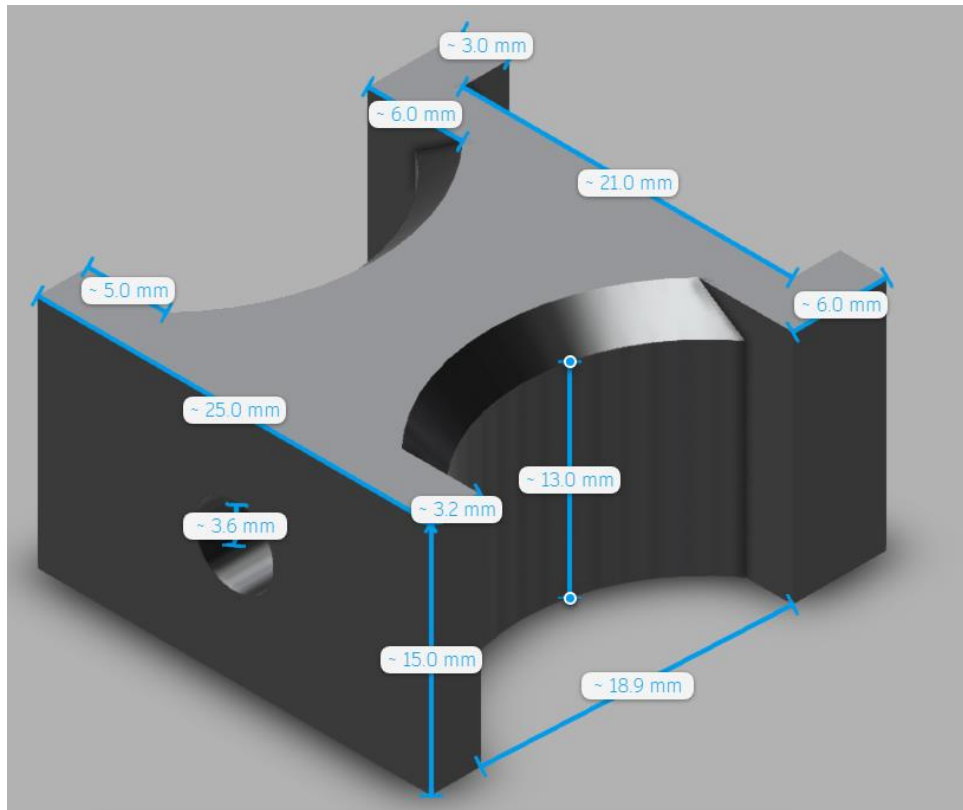
### 3.2.4.2 Mecanismo de Dispensación para el Contenedor #2

Las **Figuras 28** a la **36** representan el modelado 3D de las piezas necesarias para el mecanismo de dispensación del contenedor # 2, categorizado como grano medio.



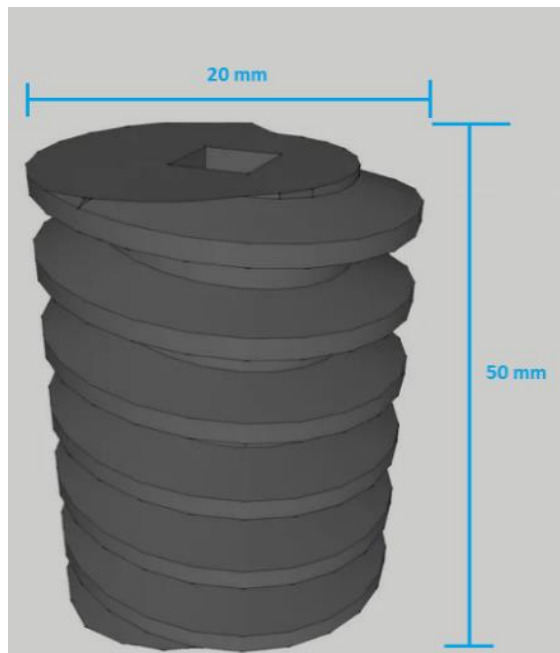
**Figura 28.** Soporte Frontal para el Servo 1

**Elaborado por:** El Investigador



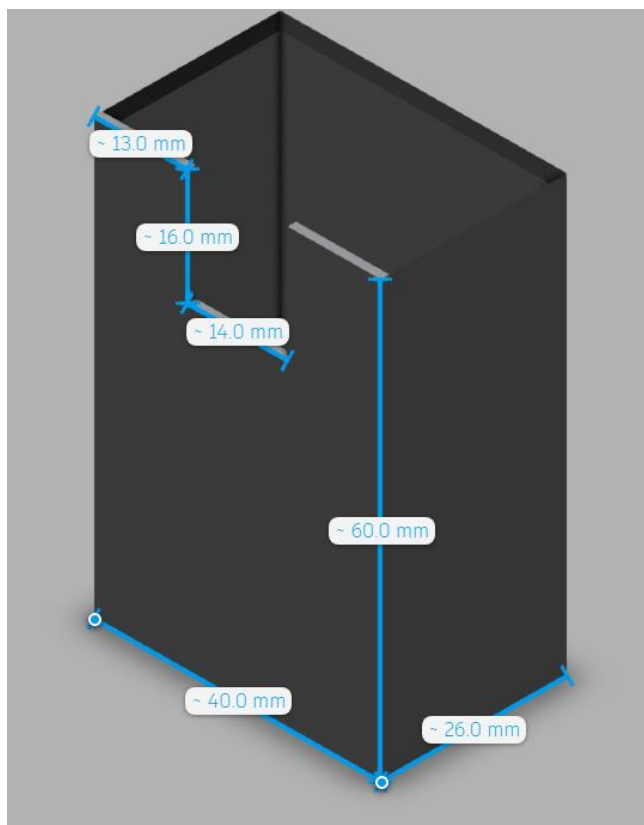
**Figura 29.** Soporte Posterior para el Servo 1

**Elaborado por:** El Investigador

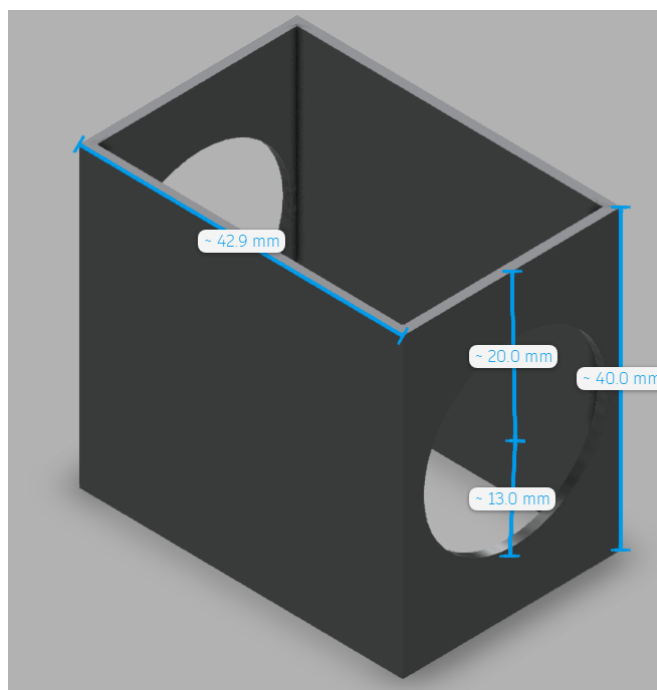


**Figura 30.** Pieza giratoria A

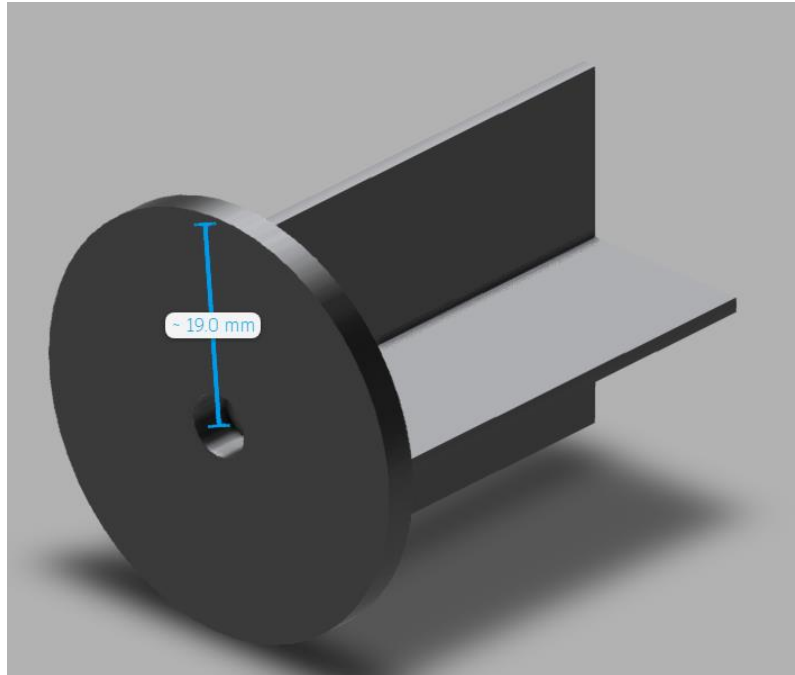
**Elaborado por:** El Investigador



**Figura 31.** Colector Primario  
**Elaborado por:** El Investigador

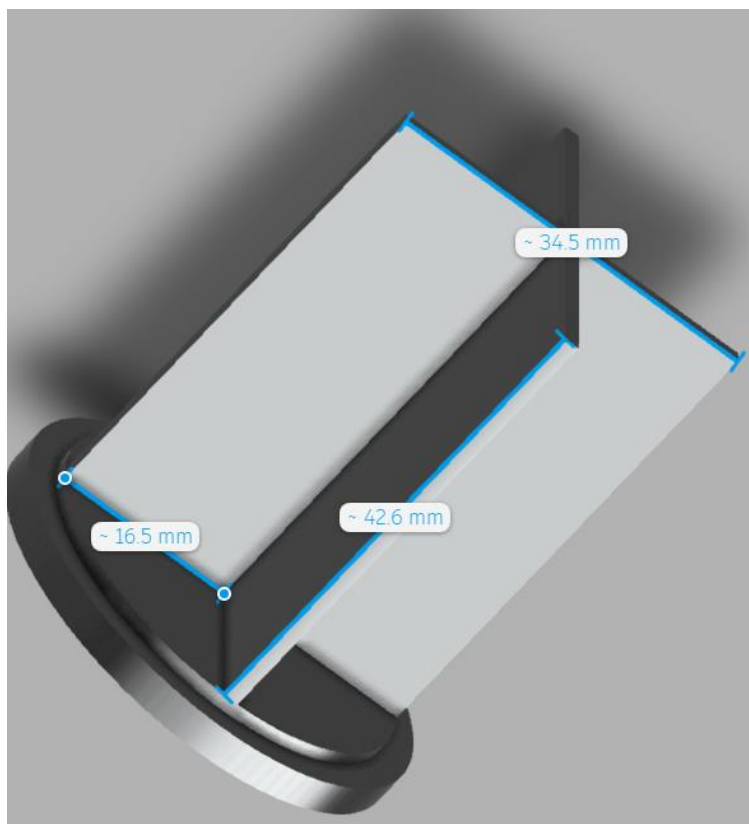


**Figura 32.** Colector Final  
**Elaborado por:** El Investigador



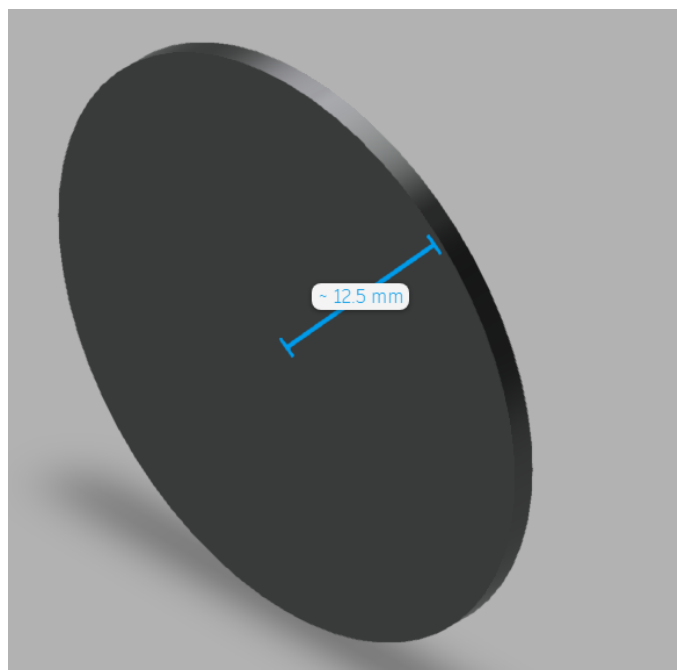
**Figura 33.** Pieza Giratoria B – Vista 1

**Elaborado por:** El Investigador



**Figura 34.** Pieza Giratoria B – Vista 2

**Elaborado por:** El Investigador



**Figura 35.** Tapa de mecanismos giratorios – Vista 1

**Elaborado por:** El Investigador

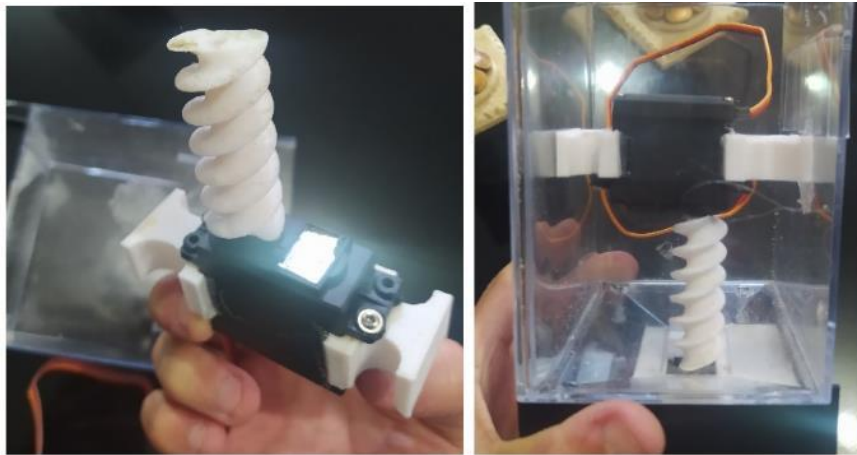


**Figura 36.** Tapa de mecanismos giratorios – Vista 2

**Elaborado por:** El Investigador

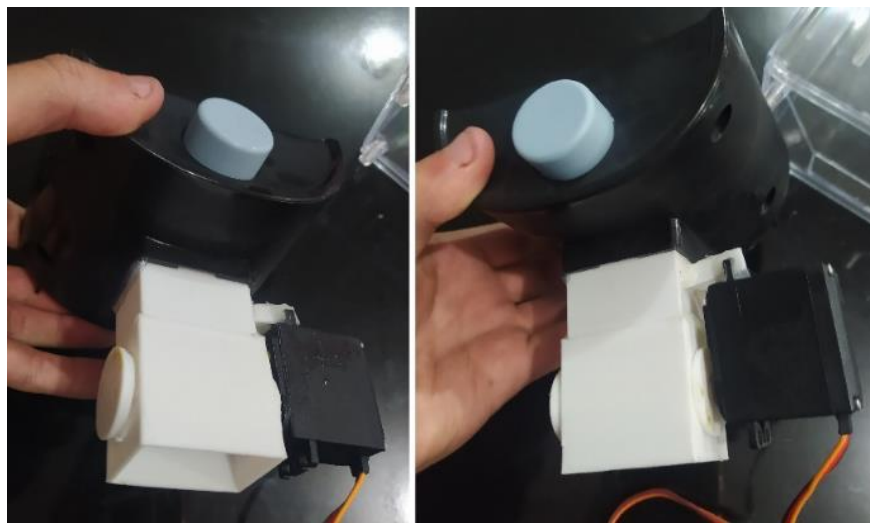


En las **Figuras 37 y 38**, se presenta el mecanismo de dispensación para el contenedor #2 ensamblado por completo.



**Figura 37.** Mecanismo del contenedor # 2 ensamblado – Parte 1

**Elaborado por:** El Investigador

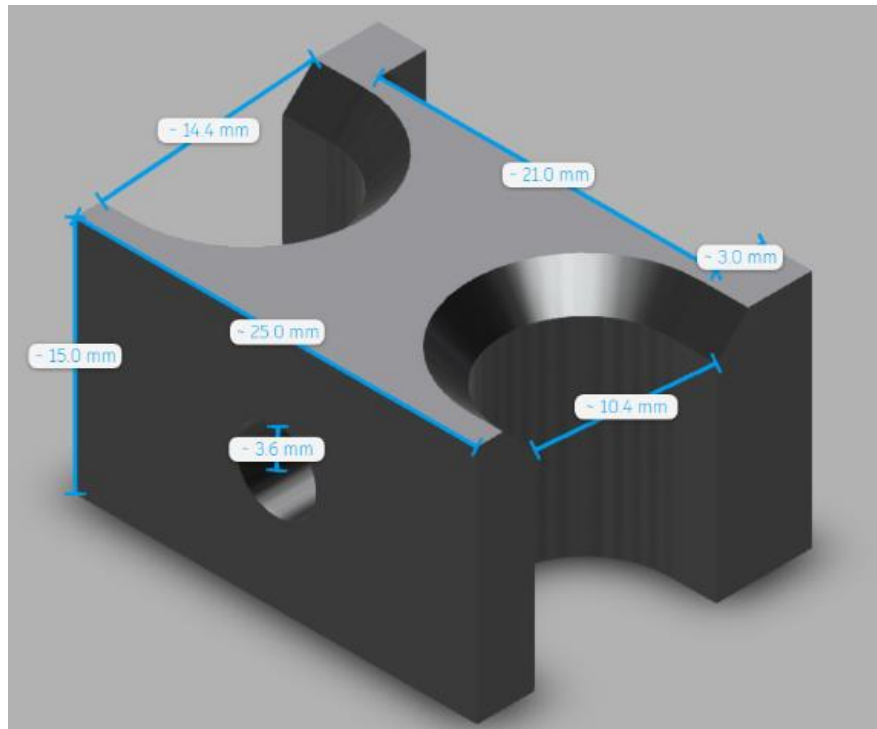


**Figura 38.** Mecanismo del contenedor # 2 ensamblado – Parte 2

**Elaborado por:** El Investigador

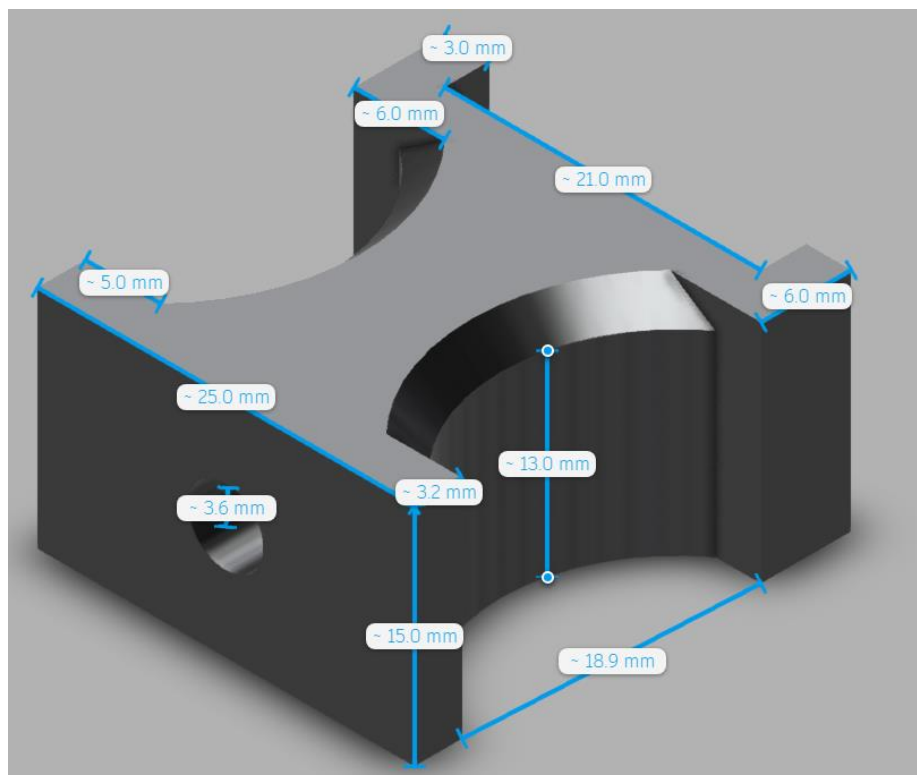
### **3.2.4.3 Mecanismo de Dispensación para los Contenedores #3 y #4**

Las **Figuras 39 a la 46** representan el modelado 3D de las piezas necesarias para el mecanismo de dispensación de los contenedores #3 y #4, categorizados como grano grueso.



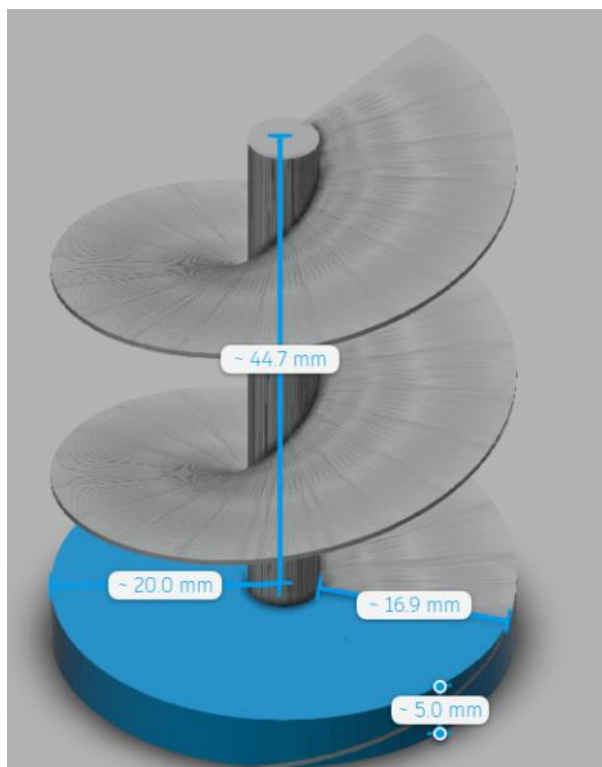
**Figura 39.** Soporte Frontal para el Servo 1

**Elaborado por:** El Investigador



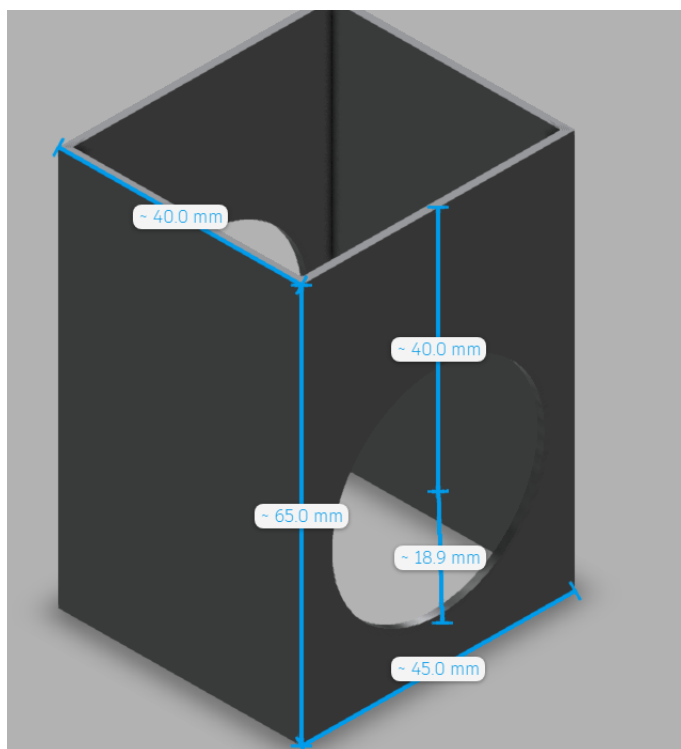
**Figura 40.** Soporte Posterior para el Servo 1

**Elaborado por:** El Investigador



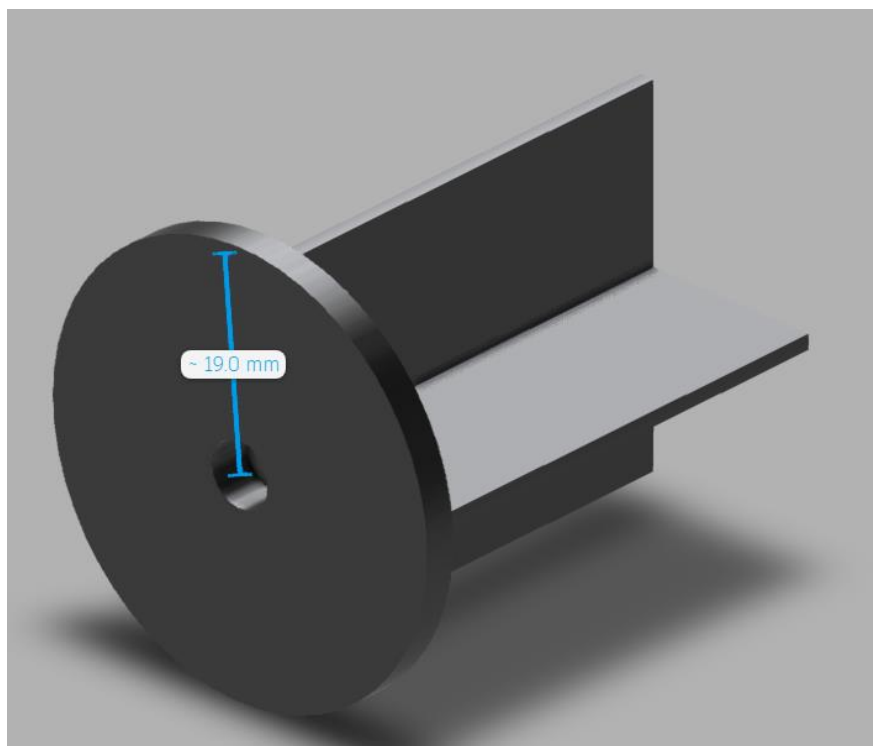
**Figura 41.** Pieza giratoria A

**Elaborado por:** El Investigador



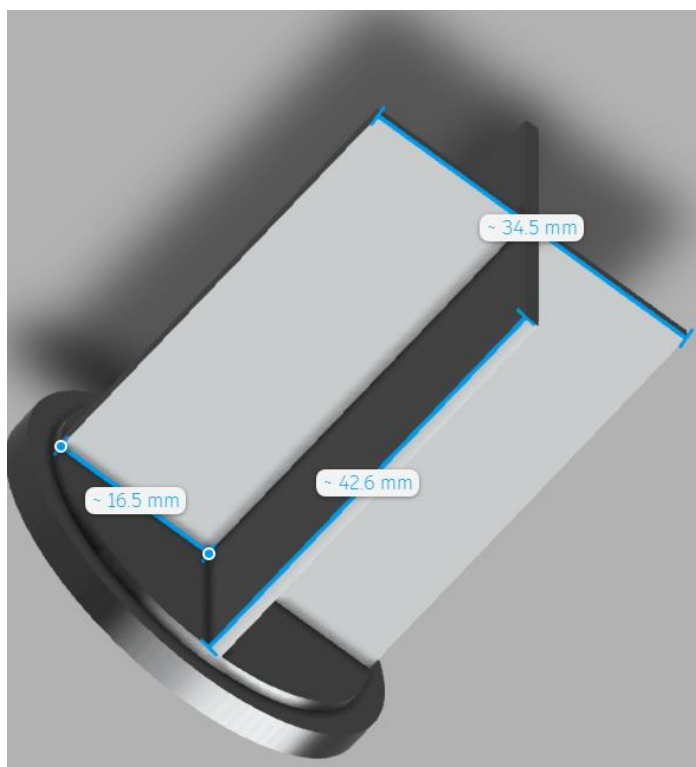
**Figura 42.** Colector Único

**Elaborado por:** El Investigador



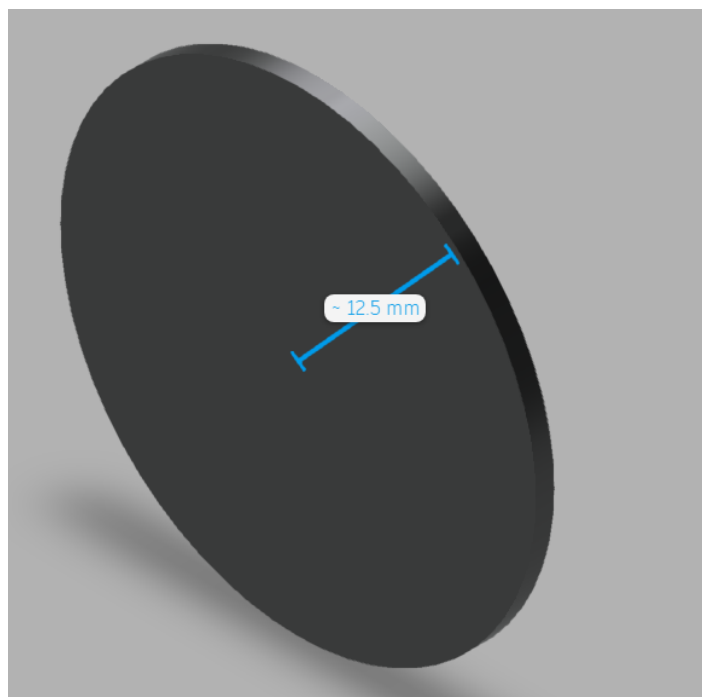
**Figura 43.** Pieza Giratoria B – Vista 1

**Elaborado por:** El Investigador



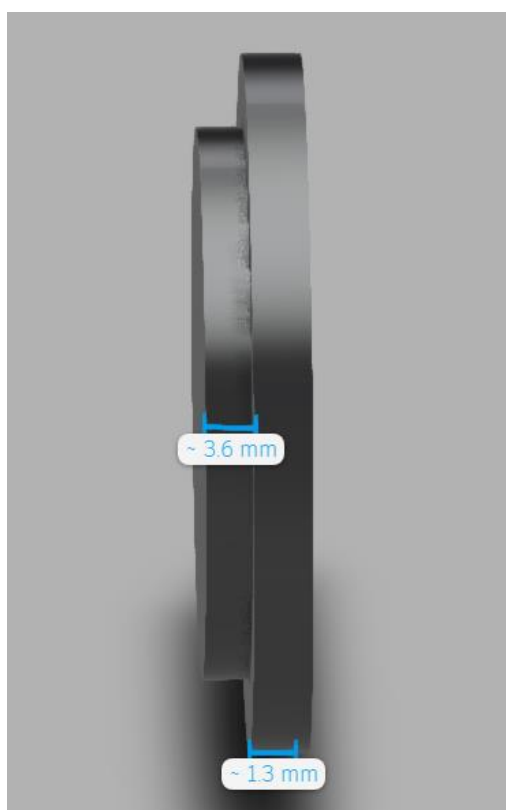
**Figura 44.** Pieza Giratoria B – Vista 2

**Elaborado por:** El Investigador



**Figura 45.** Tapa de mecanismos giratorios – Vista 1

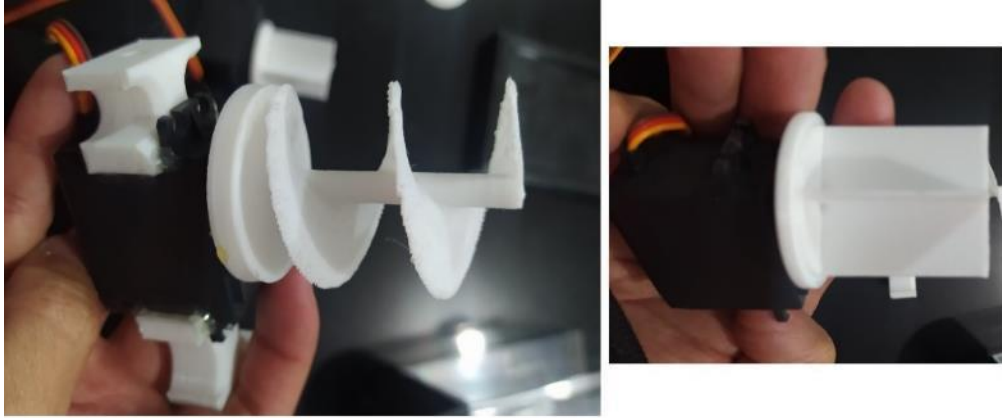
**Elaborado por:** El Investigador



**Figura 46.** Tapa de mecanismos giratorios – Vista 2

**Elaborado por:** El Investigador

En las **Figuras 47** y **48**, se presenta el mecanismo de dispensación para los contenedores #3 y #4 ensamblados por completo. El mismo mecanismo se usa tanto para dispensar pistachos como nueces, corroborando de esta manera su correcto funcionamiento para dispensar dos tipos de granos gruesos.



**Figura 47.** Mecanismo de los contenedores #3 y #4 ensamblado – Parte 1

**Elaborado por:** El Investigador



**Figura 48.** Mecanismo de los contenedores #3 y #4 ensamblado – Parte 2

**Elaborado por:** El Investigador

### **3.2.5 Bosquejo completo de la arquitectura del sistema**

La arquitectura del proyecto se muestra a detalle en la **Figura 49**, en función de la arquitectura IoT por capas propuesta y analizada en la sección 3.2.2

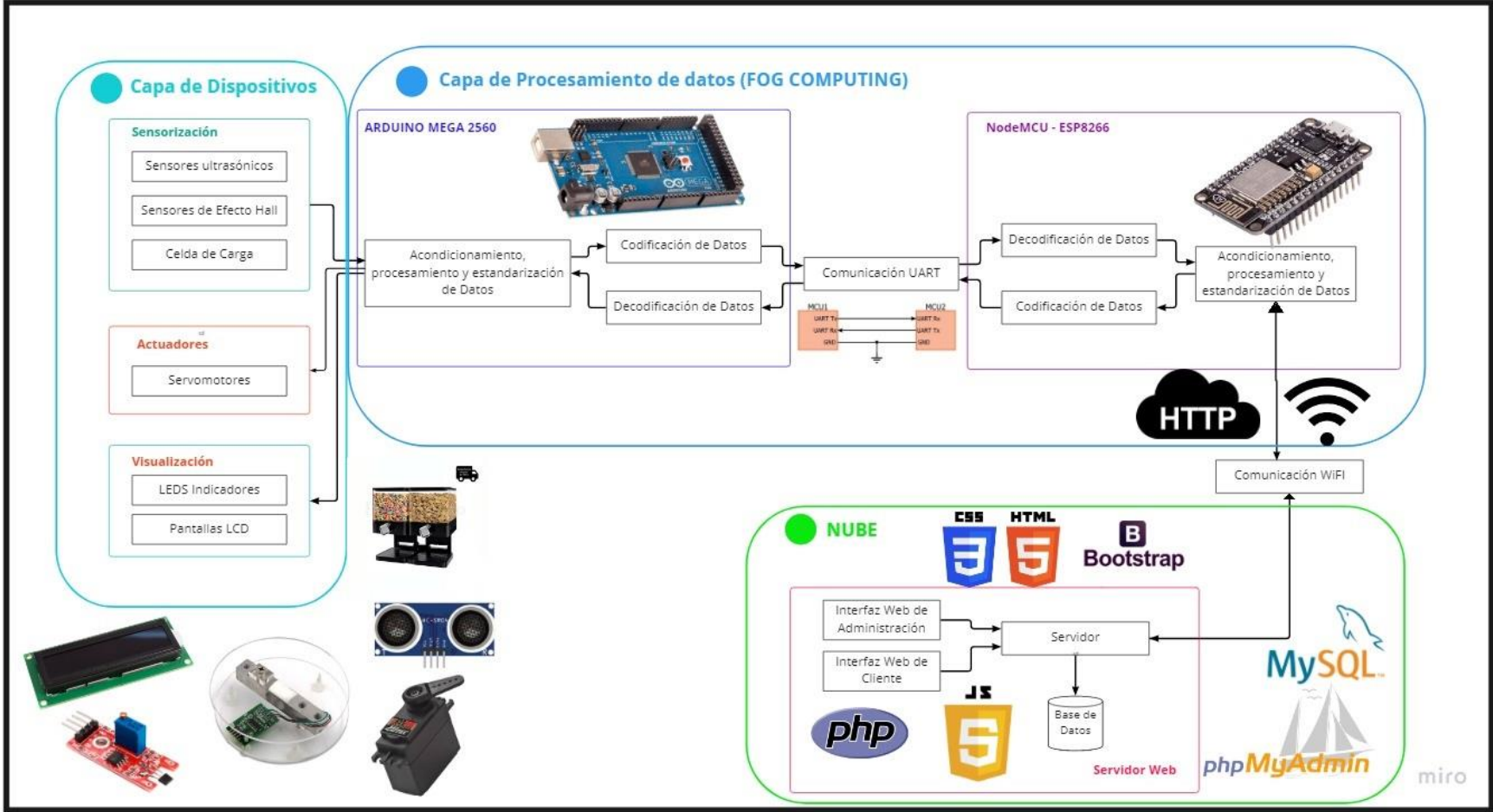


Figura 49. Bosquejo de la arquitectura Interna del Sistema

Elaborado por: el Investigador

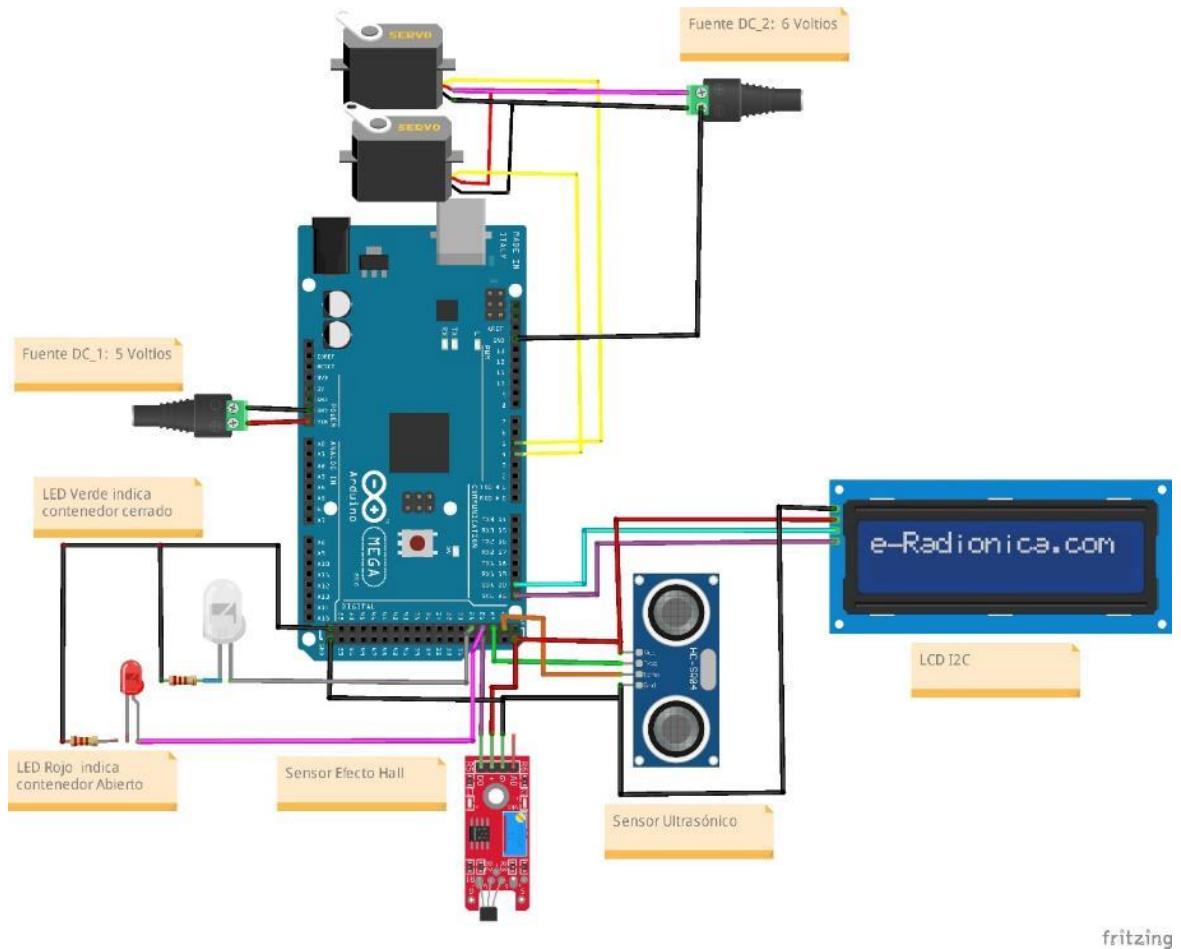
### 3.2.5.1 Capa de Dispositivos

Dentro de la capa de dispositivos se utilizan los sensores, actuadores y medios de visualización que fueron analizados y seleccionados en la sección 3.2.3. Los datos son recolectados por los sensores y botones, luego son enviados hacia las entradas del Microcontrolador Arduino Mega, éste a su vez, por medio de puertos de salida, envía señales a puertos de salida que son recibidos por servomotores, diodos LED y pantallas LCD de 16x2 posiciones.

En la **Figura 50**, se representa el esquema de conexiones de todos los sensores, actuadores y visualizadores que se conectan para un solo contenedor de frutos secos, es decir, para un mejor entendimiento, se representa un solo diagrama individual de los elementos interconectados, este mismo patrón de conexiones se repite para cada uno de los cuatro contenedores de frutos secos, ya que, todos ellos requieren de la misma cantidad de elementos que son:

- ✓ 1 Sensor de Efecto Hall (M44 o KY-024)
- ✓ 1 Sensor Ultrasónico HC-SR04
- ✓ 1 Diodo LED verde (Representa al contenedor en estado CERRADO)
- ✓ 1 Diodo LED rojo (Representa al contenedor en estado ABIERTO)
- ✓ 1 LCD I2C de 16x2 (Imprime los datos de los sensores)
- ✓ 1 Celda de carga de 5Kg junto con su controlador HX711
- ✓ 2 Servomotores 360° de giro continuo MG-995 (Alimentados por una fuente independiente)
- ✓ 2 Resistencias de 330  $\Omega$ .
- ✓ Conectores para Fuentes DC
- ✓ Cables Macho – Macho
- ✓ Cables Macho - Hembra

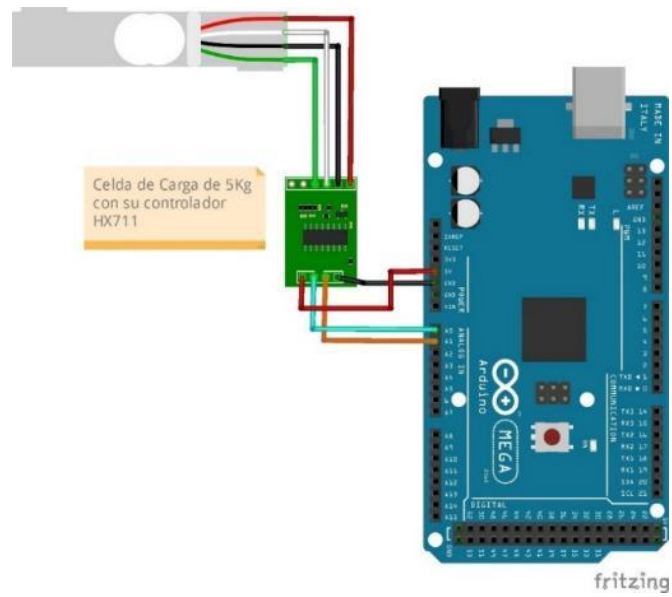




**Figura 50.** Esquema de conexión de un contenedor individual

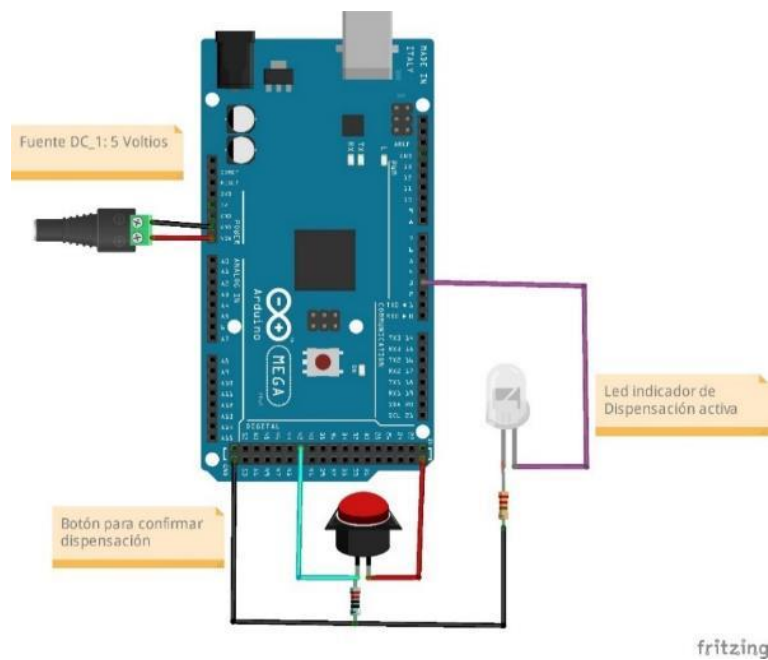
**Elaborado por:** El Investigador

En la capa de dispositivos también se incluye la celda de carga y el botón de confirmación de dispensación, dispositivos cuya funcionalidad fue detallada en la sección 3.2 junto con el resto de los elementos del proyecto. En la **Figura 51**, se muestra el esquema de conexión para la celda de carga de 5Kg, mientras que, en la **Figura 52**, se muestra el esquema de conexión para el botón de confirmación.



**Figura 51.** Esquema de conexión de la celda de carga

**Elaborado por:** El Investigador



**Figura 52.** Esquema de conexión de un contenedor individual

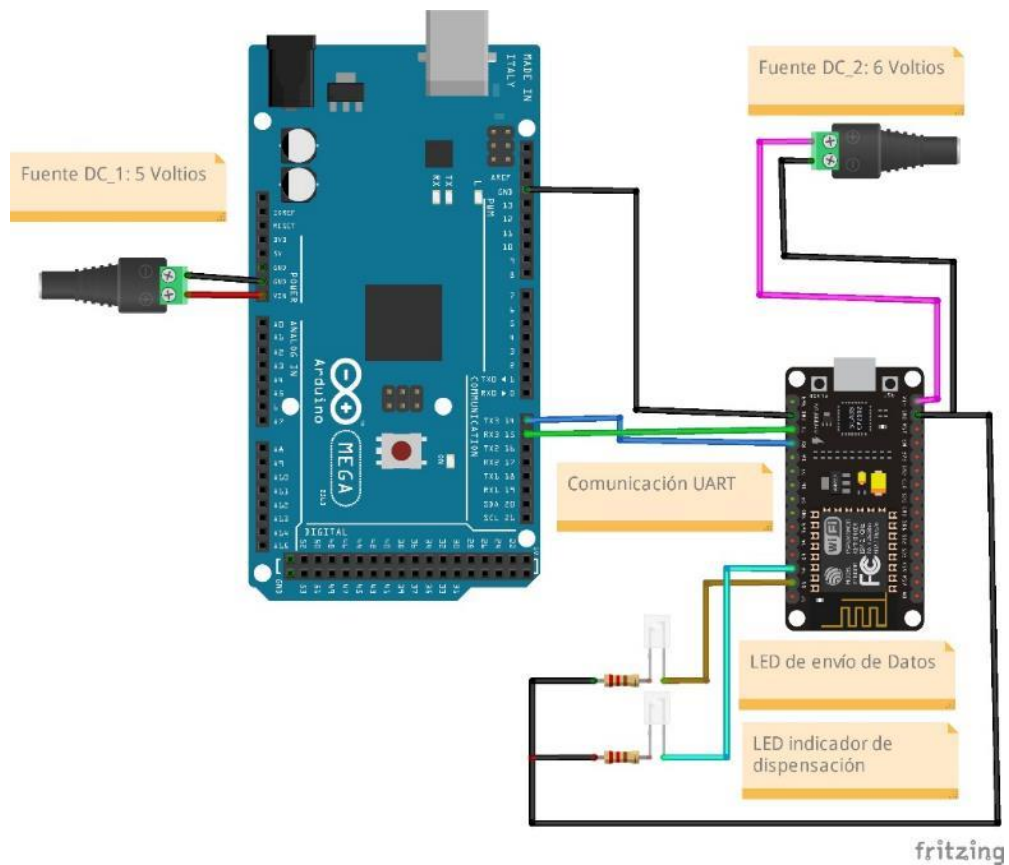
**Elaborado por:** El Investigador

### 3.2.5.2 Capa de Procesamiento de Datos (FOG COMPUTING)

La capa de procesamiento de datos escogida para el presente proyecto tendrá lugar en la niebla (o FOG computing), conceptos que fueron detallados en la sección 1.3.9, es

decir, los datos **no** serán procesados o validados por los sensores ya que estos no tienen la capacidad necesaria para decidir por sí solos, ni tampoco por el servidor en la nube, sino más bien, los datos se procesarán en un punto intermedio que es **el microcontrolador**, de esta manera, se reduce la latencia y se evitan las sobrecargas de datos innecesarios en el servidor, debido que el Arduino Mega tiene la capacidad necesaria para decidir si se envía o no un dato leído por un sensor.

En esta etapa, el microcontrolador principal es el Arduino Mega, siendo este el que se encarga de procesar todos los datos leídos por los sensores, imprimirlos en los medios de visualización y además activar o desactivar los actuadores; Mientras que, el microcontrolador NODEMCU es únicamente usado para enviar los datos al servidor por medio de su funcionalidad de conexión inalámbrica, misma que carece el Arduino Mega. Ambos microcontroladores se comunican de manera alámbrica usando el protocolo UART, estableciendo una comunicación semi-duplex. En la **Figura 53** se representa el esquema de conexión entre ambos microcontroladores.



**Figura 53.** Esquema de conexión entre Arduino Mega y NODEMCU

**Elaborado por:** El Investigador

### 3.2.5.3 Capa de Servicios en la Nube

Para el desarrollo de las interfaces web se emplearon conocimientos adquiridos a lo largo de la carrera universitaria, mismos que se detallan a continuación.

#### Front – End

- ✓ **HTML**, usado para formar todas las estructuras base de los sitios web.
- ✓ **CSS**, usado para los estilos de cada elemento creado.
- ✓ **JavaScript**, usado para crear funciones dinámicas que aumentan la interactividad entre el usuario y las páginas web.

Al tener como base conocimientos sólidos de estos tres lenguajes, se decidió hacer uso de plantillas desarrolladas en el Framework **BootStrap**, el cual, combina CSS y JavaScript para estandarizar funciones de suma utilidad al momento de dar estilos a los elementos de las páginas web.

#### Back – End

- ✓ **SQL**, lenguaje que se ejecuta para hacer múltiples consultas y actualizaciones a la base de datos, se usa en conjunto con PHP.
- ✓ **PHP**, utilizado para realizar todas las validaciones de Inicios de sesión, seguridad de datos, ejecución de sentencias SQL, renderización de elementos por medio de bucles que ejecutan lenguaje HTML en conjunto con datos extraídos de la base de Datos.

El conocimiento necesario en todos los lenguajes mencionados para el desarrollo del Front-End y Back-End permitieron emplear las librerías como:

- ✓ **JQUERY**, que permite la ejecución de peticiones AJAX para extraer y presentar datos de la Base en tiempo real sin necesidad de recargar manualmente ninguna sección de la página web empleando JavaScript y PHP simultáneamente.
- ✓ **HIGHCHARTS**, que permite el uso de gráficas interactivas desarrolladas en JavaScript, HTML y CSS.

## **Selección del Protocolo de mensajería IoT**

El protocolo usado para el desarrollo del presente proyecto es HTTP, debido a su alta compatibilidad con PHP y con varias librerías de Arduino para el envío de datos desde la NODEMCU. Además, por simplicidad de la arquitectura, dentro del Back-End no se utiliza lenguajes como Python, ni tampoco ningún entorno de Linux, requerimientos que son necesarios para ejecutar otro tipo de protocolos. En cuanto al envío de datos en la arquitectura, la conexión no siempre está abierta, ya que, el microcontrolador envía datos al servidor únicamente si existe algún dato distinto al último registrado en la Base, y solo si el cliente realiza algún pedido. Razones por las cuales se selecciona HTTP.

## **Selección modelo de servicio en la Nube**

El presente proyecto necesita de un servidor en la Nube. Debido a que el prototipo se lo desarrolla de forma local, sus archivos necesarios para su funcionamiento pueden ser alojados en la Nube en dos de los tres modelos de servicios existentes:

- ✓ **Plataforma como Servicio (Paas).** Hosting Web disponibles en el mercado que tengan instalado todos los componentes necesarios para correr la programación web, principalmente que permitan la administración de Bases de Datos MySQL por medio de PhpMyAdmin y servicios como FTP para subir todos los archivos desarrollados en el Proyecto. Algunos ejemplos de Paas son los Hosting Web 000WebHost, DreamHost, Hostgator, Hostinger, etc.
  
- ✓ **Infraestructura como Servicio (Iaas).** Un claro ejemplo de este modelo de servicio es Microsoft Azure, en el cuál, el usuario puede construir su servidor instalando todos los componentes necesarios para el proyecto, se requiere sólidos conocimientos en Linux para poder instalar, conectar y ejecutar todas las dependencias del servidor.

Tanto Paas como Iaas son modelos de servicio en la nube que se presentan en el mercado de forma gratuita y de pago, sin embargo, las versiones gratuitas poseen funciones y recursos muy limitados, razón por la cual, se desarrolla la ejecución web del prototipo de forma local, además, este va a ser utilizado en la red LAN de las

instalaciones de empresa Lula Organic. En el caso de que la empresa requiera desplegar el sistema a una red más amplia, se podrían utilizar los modelos de servicios mencionados anteriormente.

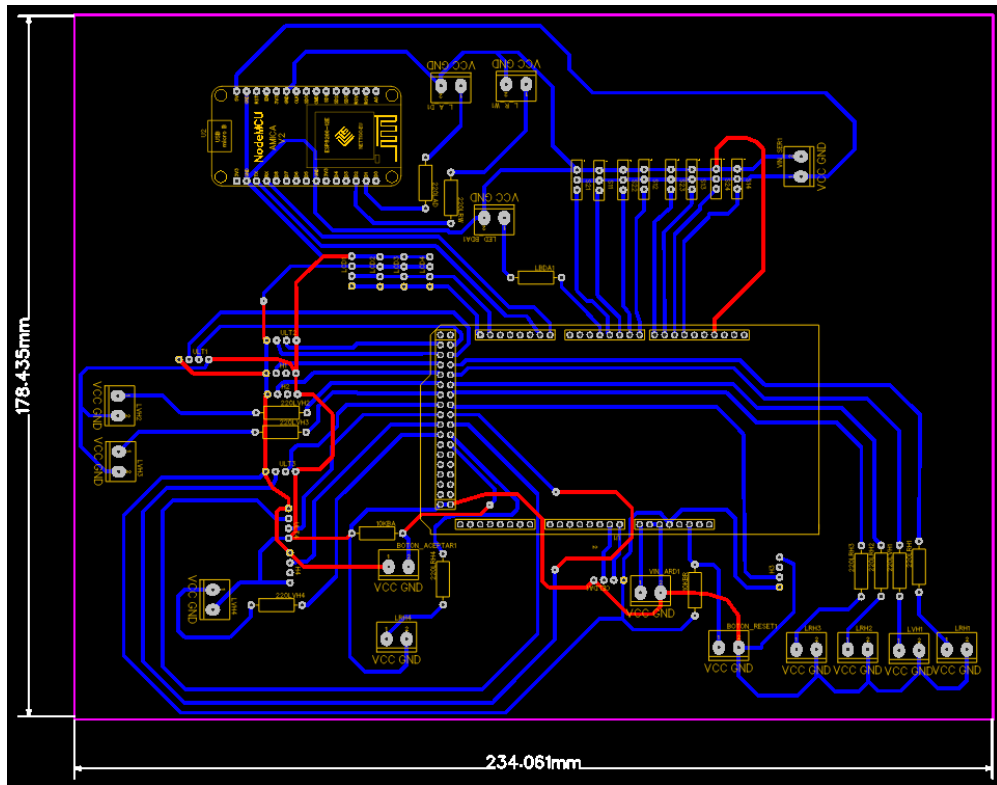
### **3.2.6 Diseño de la placa de circuito impreso (PCB)**

La placa que se usa para el dispensador de frutos secos consta de los siguientes componentes:

- ✓ 1 NodeMCU
- ✓ 1 Arduino Mega 2560
- ✓ 4 Pantallas LCD de 16x2
- ✓ 4 Sensores ultrasónicos HC-SR04
- ✓ 11 Diodos LED
- ✓ 4 Sensores de Efecto Hall
- ✓ 8 Servomotores Mg-995
- ✓ 2 Entradas de Alimentación
- ✓ 1 Botón
- ✓ 11 Resistencias de 330 o 220  $\Omega$
- ✓ 2 Resistencias de 10k  $\Omega$

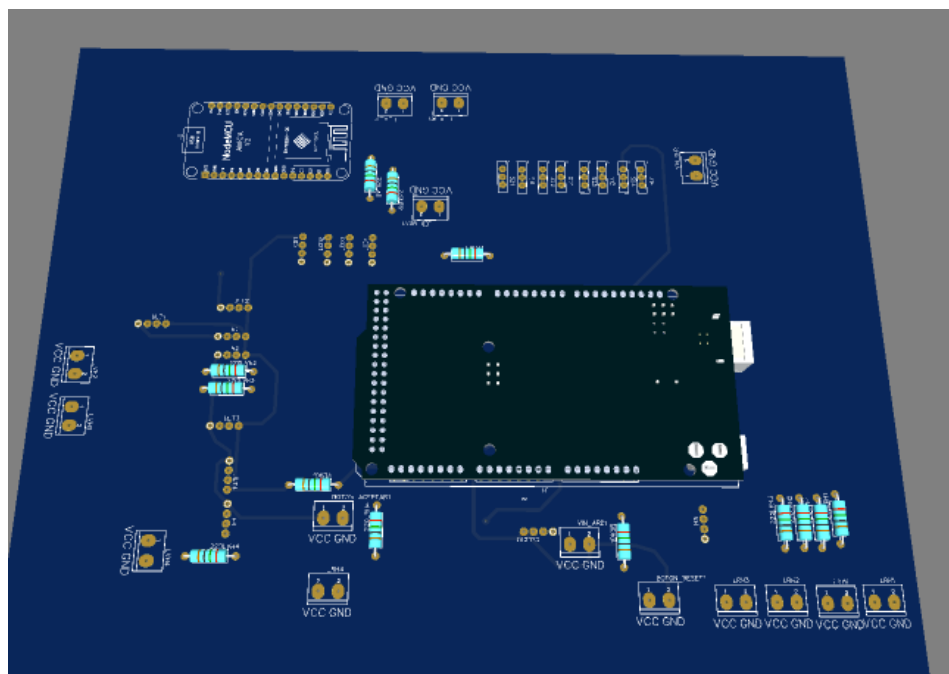
En el **Anexo 1** se muestra el diseño esquemático con todas las conexiones respectivas realizado en el software EasyEda, además de las borneras para cada elemento que se conecta de manera externa.

El Diseño impreso se lo realizó en dos capas, una inferior de color azul y una superior de color rojo, mismo que fue realizado también en el software EasyEda. En las **Figuras 54, 55 y 56** se muestra el diseño del circuito impreso mientras que en la **Figura 57** se muestra su implementación.



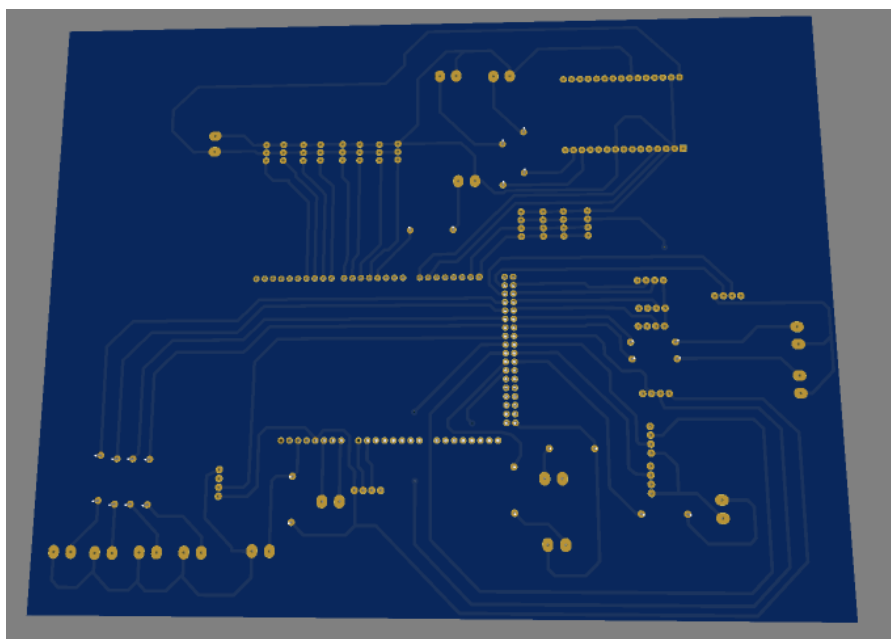
**Figura 54.** Diseño Circuito Impreso

Elaborado por: El Investigador



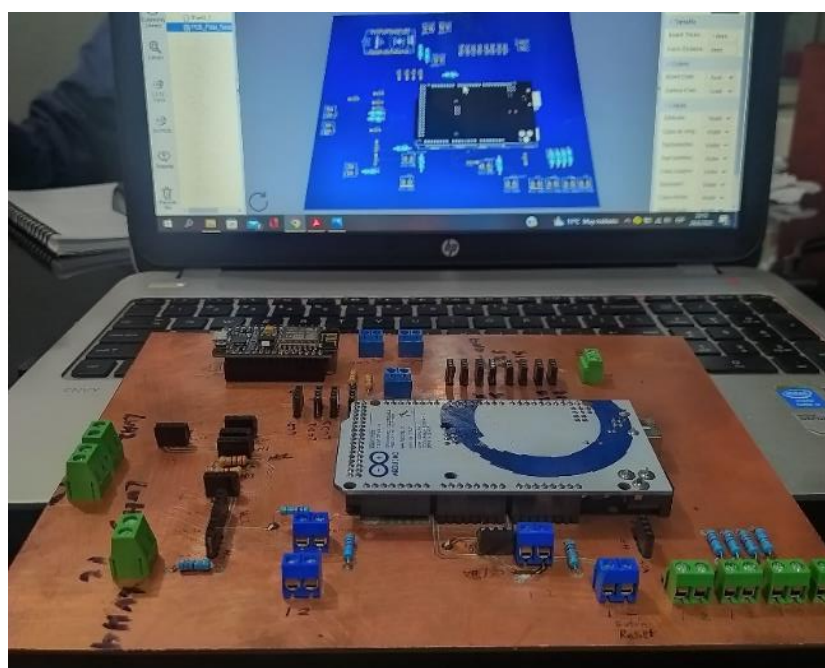
**Figura 55.** Diseño Circuito Impreso 3D – Capa Superior

Elaborado por: El Investigador



**Figura 56.** Diseño Circuito Impreso 3D – Capa Inferior

**Elaborado por:** El Investigador

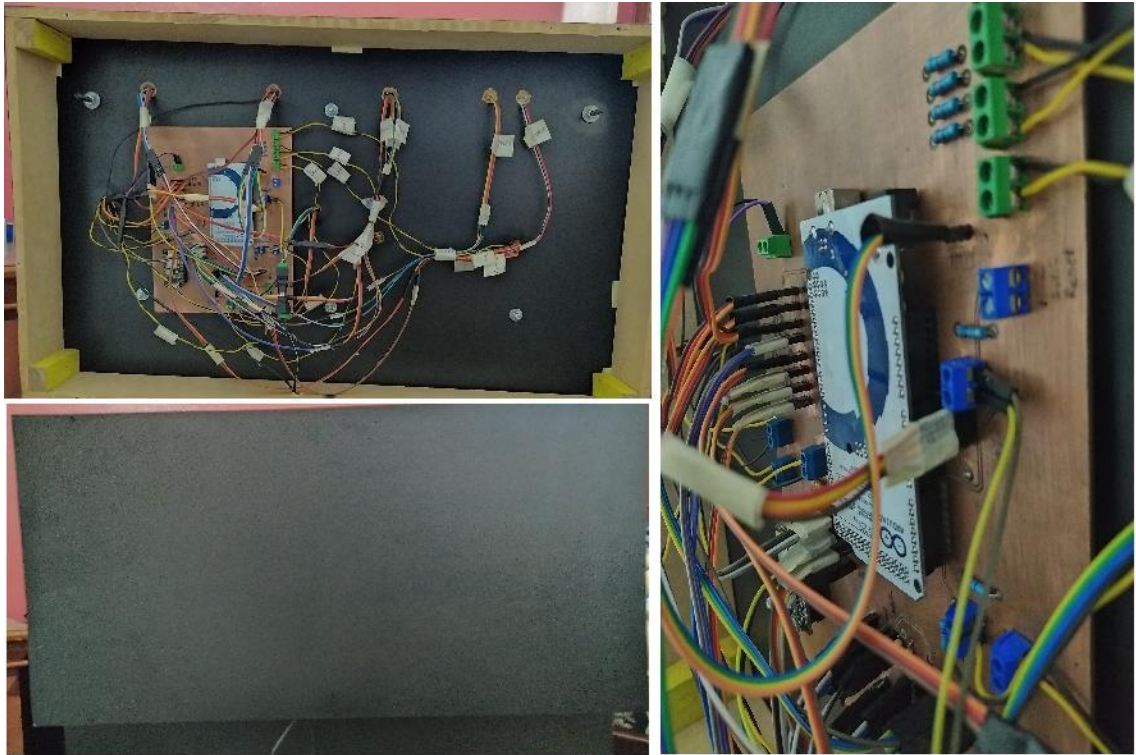


**Figura 57.** Circuito Impreso Implementado

**Elaborado por:** El Investigador

En la **Figura 58** se muestran las conexiones finales del circuito PCB ubicado en la parte posterior del case del dispensador.



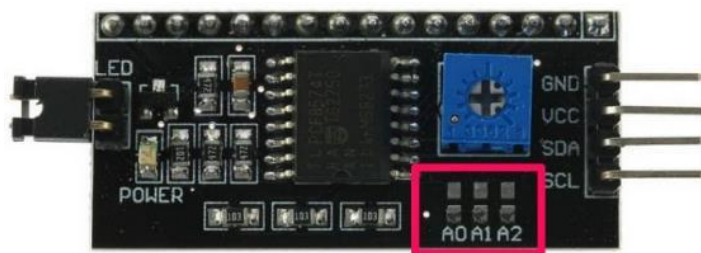


**Figura 58.** Conexiones del Sistema

**Elaborado por:** El Investigador

### 3.2.7 Direccionamiento I2C

Las pantallas LCD necesitan cuatro direcciones I2C diferentes para poder funcionar debido a que todas se encuentran conectadas a los mismos pines del Arduino Mega, estas serán definidas mediante el software IDE de Arduino y además deben ser soldados los pines respectivos según el direccionamiento establecido. En la **Tabla 8**, se presenta en una guía todas las posibles direcciones que se pueden ocupar junto con los pines que deberán modificarse, mientras que en la **Figura 59** se muestran los pines A0, A1 y A2 que deberán ser soldados de acuerdo con la dirección que se escoja.



**Figura 59.** Pines del Direccionamiento I2C

**Elaborado por:** El Investigador

**Tabla 8.** Guía para el Direccionamiento I2C

| A0 | A1 | A2 | Dirección |
|----|----|----|-----------|
| 1  | 1  | 1  | 0x27      |
| 0  | 1  | 1  | 0x26      |
| 1  | 0  | 1  | 0x25      |
| 0  | 0  | 1  | 0x24      |
| 1  | 1  | 0  | 0x23      |
| 0  | 1  | 0  | 0x22      |
| 1  | 0  | 0  | 0x21      |
| 0  | 0  | 0  | 0x20      |

**Elaborado por:** El Investigador

El direccionamiento escogido para las pantallas LCD se muestra en la **Tabla 9** y en la **Figura 60**.

**Tabla 9.** Direccionamiento I2C Seleccionado

| LCD    | A0 | A1 | A2 | Dirección |
|--------|----|----|----|-----------|
| LCD #1 | 1  | 1  | 1  | 0x27      |
| LCD #2 | 1  | 1  | 0  | 0x23      |
| LCD #3 | 1  | 0  | 1  | 0x25      |
| LCD #4 | 0  | 1  | 1  | 0x26      |

**Elaborado por:** El Investigador

```
// Pines y objeto pantalla LCD I2C =====  
LiquidCrystal_I2C lcd_grano1 (0x27, 2, 1, 0, 4, 5, 6, 7); // DIR, E, RW, RS, D4,  
LiquidCrystal_I2C lcd_grano2 (0x23, 2, 1, 0, 4, 5, 6, 7);  
LiquidCrystal_I2C lcd_grano3 (0x25, 2, 1, 0, 4, 5, 6, 7);  
LiquidCrystal_I2C lcd_grano4 (0x26, 2, 1, 0, 4, 5, 6, 7);  
//=====
```

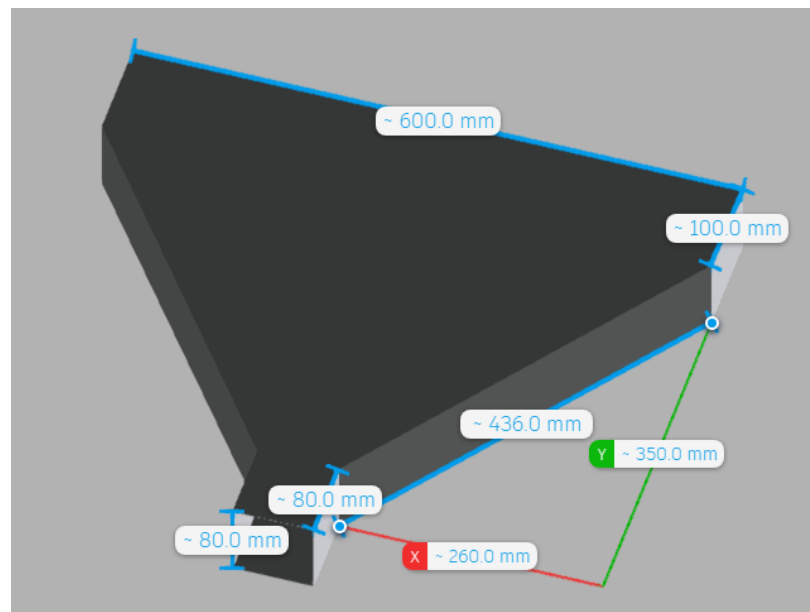
**Figura 60.** Direccionamiento I2C Establecido en Arduino

**Elaborado por:** El Investigador

Por lo tanto, en cada dirección que contiene un 0 lógico en su dirección, se procedió a soldar el pin correspondiente.

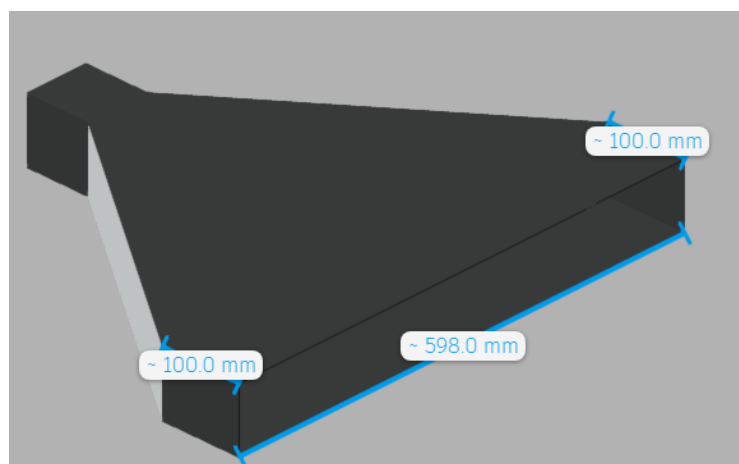
### 3.2.8 Mecanismo guía de la caída de los frutos secos

Los frutos secos dispensados por cada uno de los contenedores son guiados por una pieza diseñada a manera de tolva invertida, y fue construida en acero inoxidable, de esta manera, se asegura que todos los frutos se deslicen por el metal hasta llegar a un recipiente final. En las **Figuras 61 y 62** se muestra el diseño 3D de la tolva con sus respectivas medidas, mientras que en las **Figuras 63 y 64** se observa la tolva construida e implementada.



**Figura 61.** Diseño de la Tolva Invertida – Vista 1

**Elaborado por:** El Investigador



**Figura 62.** Diseño de la Tolva Invertida – Vista 2

**Elaborado por:** El Investigador



**Figura 63.** Tolva Invertida Implementada – Vista 1

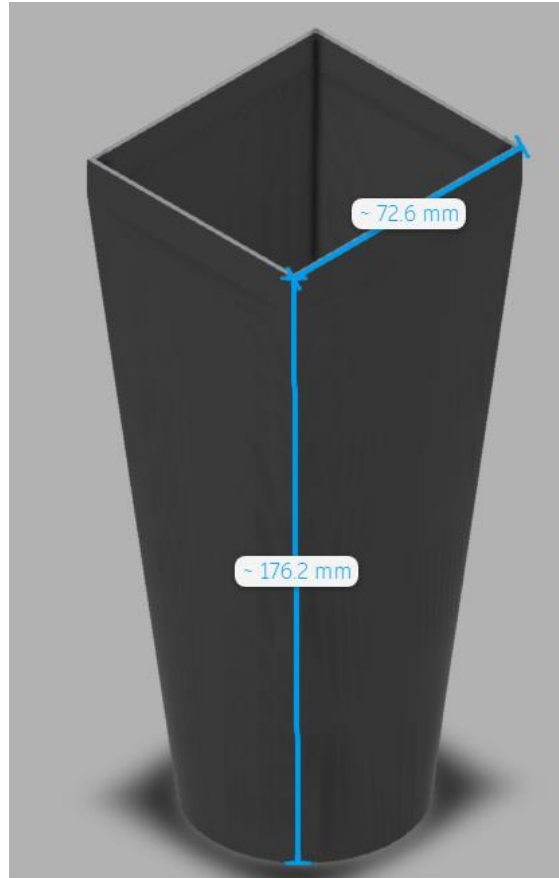
**Elaborado por:** El Investigador



**Figura 64.** Tolva Invertida Implementada – Vista 2

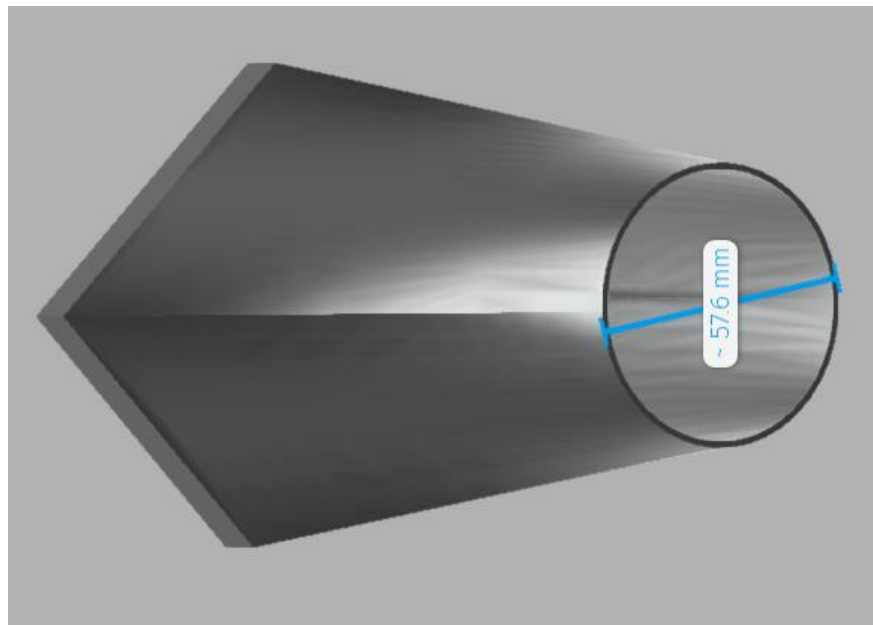
**Elaborado por:** El Investigador

Adicionalmente se diseña una boca final con el propósito de evitar caídas del producto en lugares fuera del recipiente. En las **Figuras 65 y 66** se muestra el diseño 3D de la pieza mencionada.



**Figura 65.** Boca final de la Tolva – Vista 1

**Elaborado por:** El Investigador

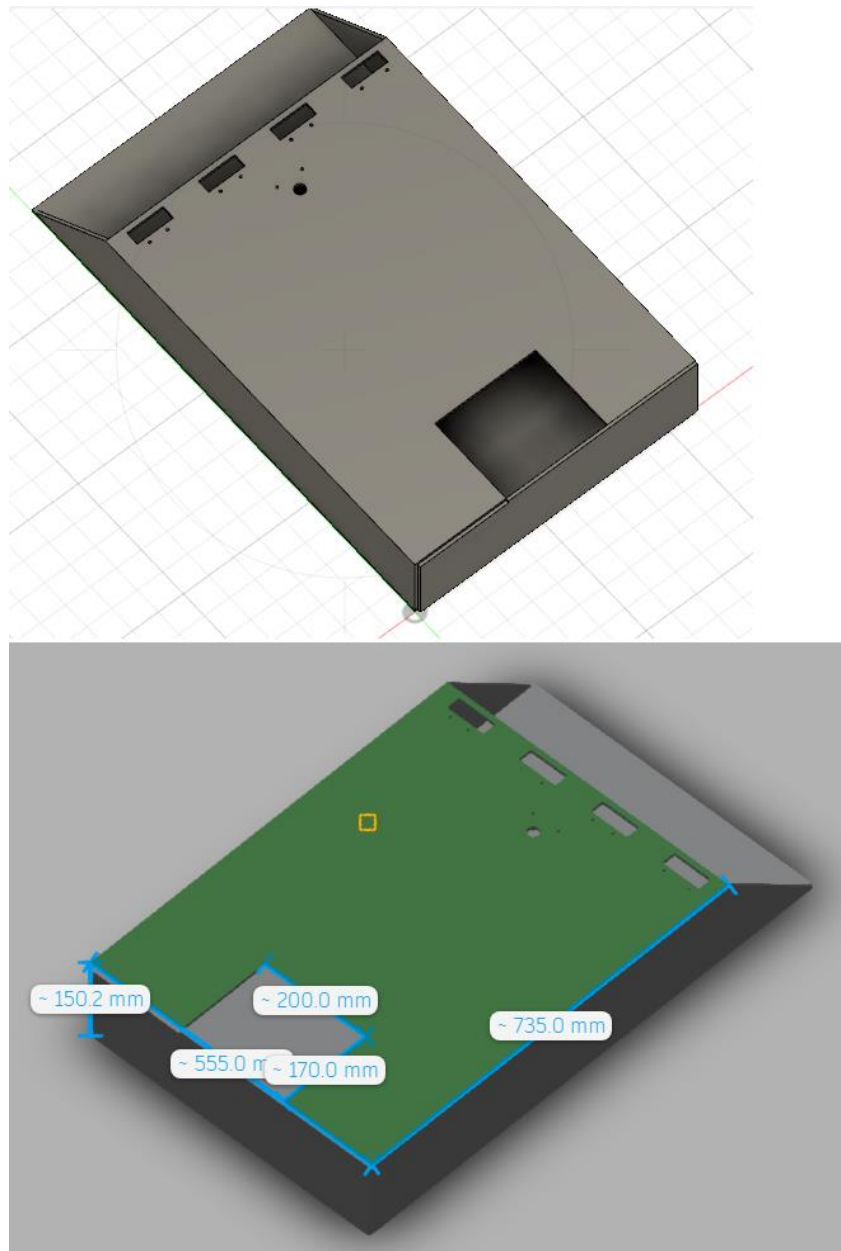


**Figura 66.** Boca final de la Tolva – Vista 2

**Elaborado por:** El Investigador

### 3.2.9 Esquema del Dispensador

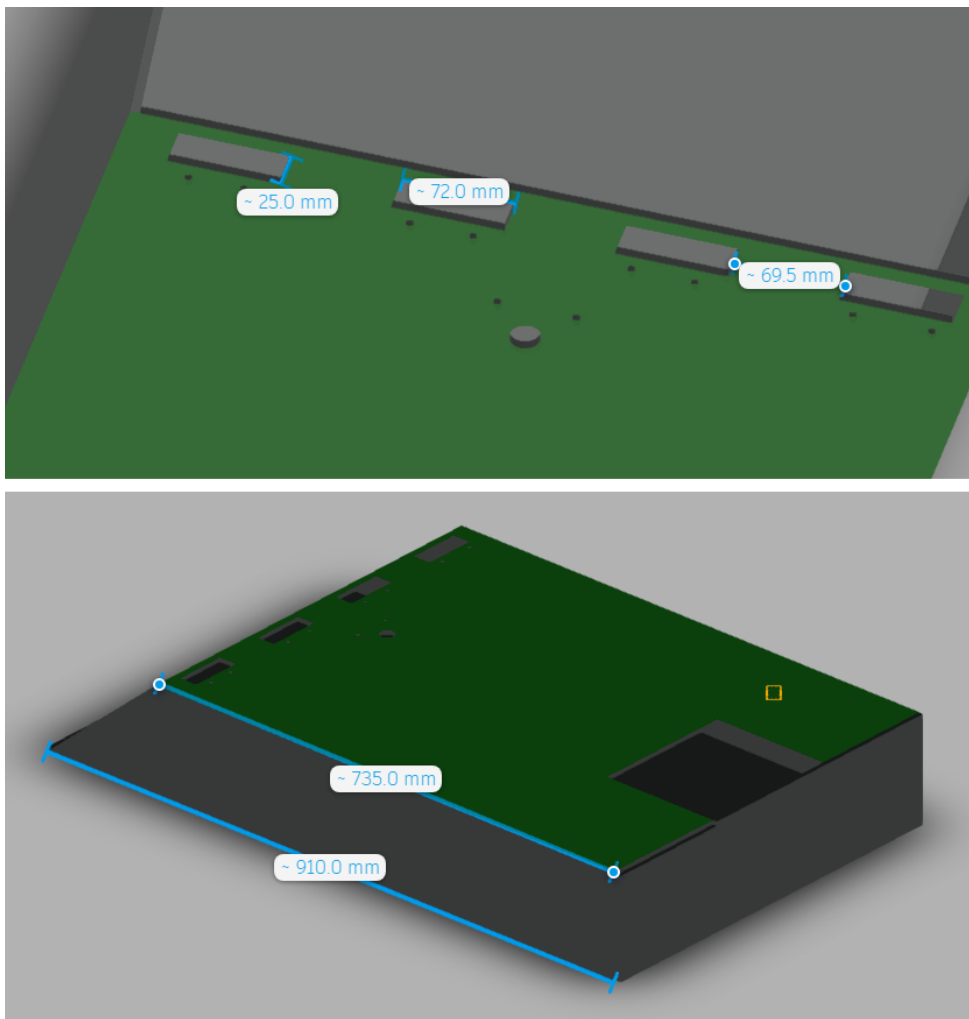
Debido a que se utilizan cuatro contenedores de frutos secos de hasta 1 Kg de capacidad cada uno, junto con la tolva invertida, y los demás elementos con sus respectivas conexiones, se diseña un case de tamaño considerable. Sin embargo, para reducir el peso de todo el prototipo, el material seleccionado para la construcción del case es de madera de tipo MDF. En las **Figuras 67** y **68** se muestra el diseño en 3D del case junto con sus medidas.



**Figura 67.** Diseño 3D del dispensador completo – Vista 1

**Elaborado por:** El Investigador





**Figura 68.** Diseño 3D del dispensador completo – Vista 2

**Elaborado por:** El Investigador

A continuación, en las **Figuras 69 y 70** se observa el case implementado.



**Figura 69.** Dispensador de frutos Secos Implementado – Vista 1

**Elaborado por:** El Investigador



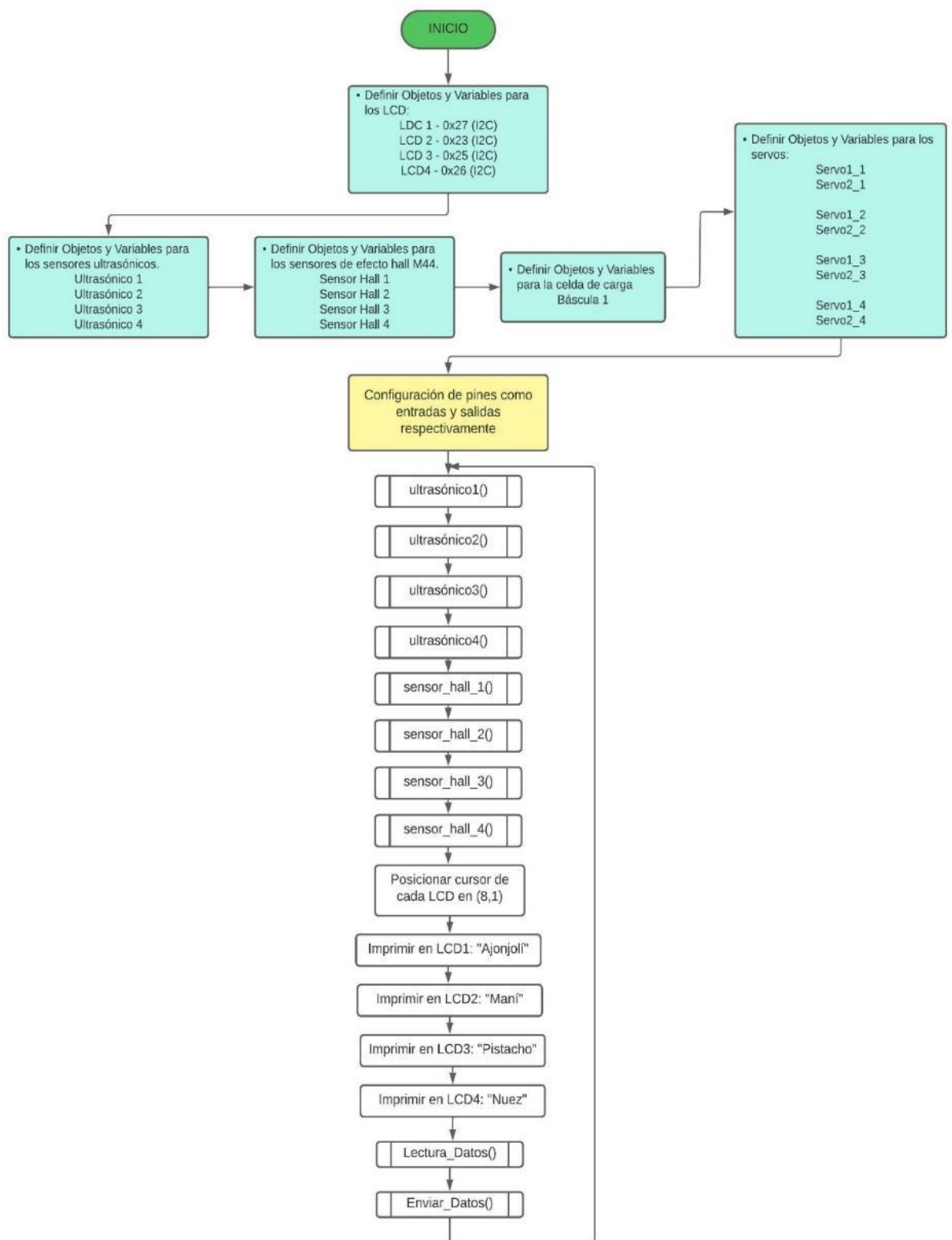
**Figura 70.** Dispensador de frutos Secos Implementado – Vista 2

**Elaborado por:** El Investigador

### 3.2.10 Diagrama de flujo para el Algoritmo de programación del Arduino Mega2560

En la **Figura 71** se puede visualizar el diagrama de flujo del algoritmo general programado en la placa Arduino Mega 2560, posterior a ésta, se detalla el algoritmo de cada subrutina utilizada en distintas figuras respectivamente.





**Figura 71.** Diagrama de Flujo General del Arduino Mega

**Elaborado por:** El Investigador

a. Diagrama de flujo para la Subrutina ultrasónico1()

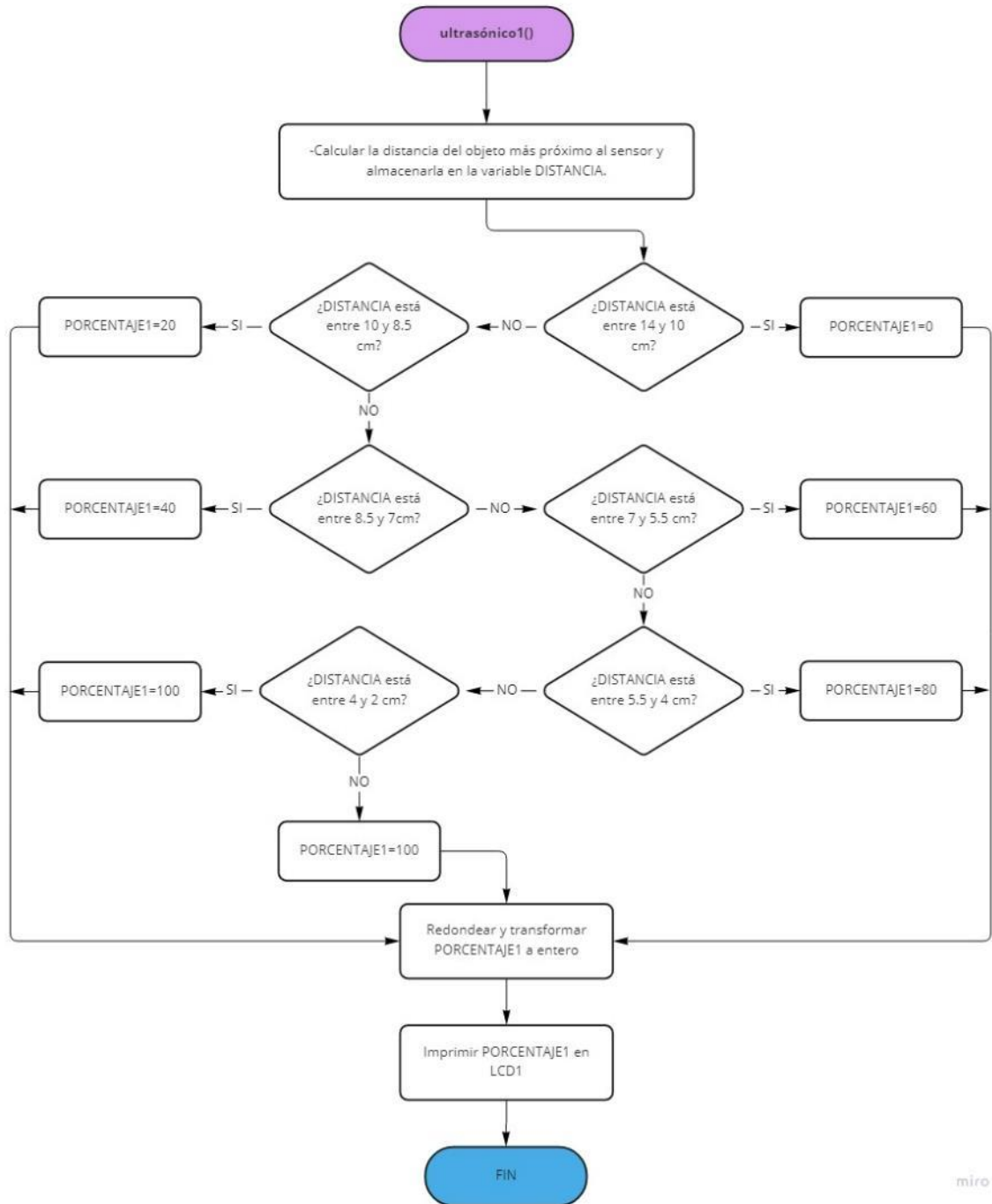
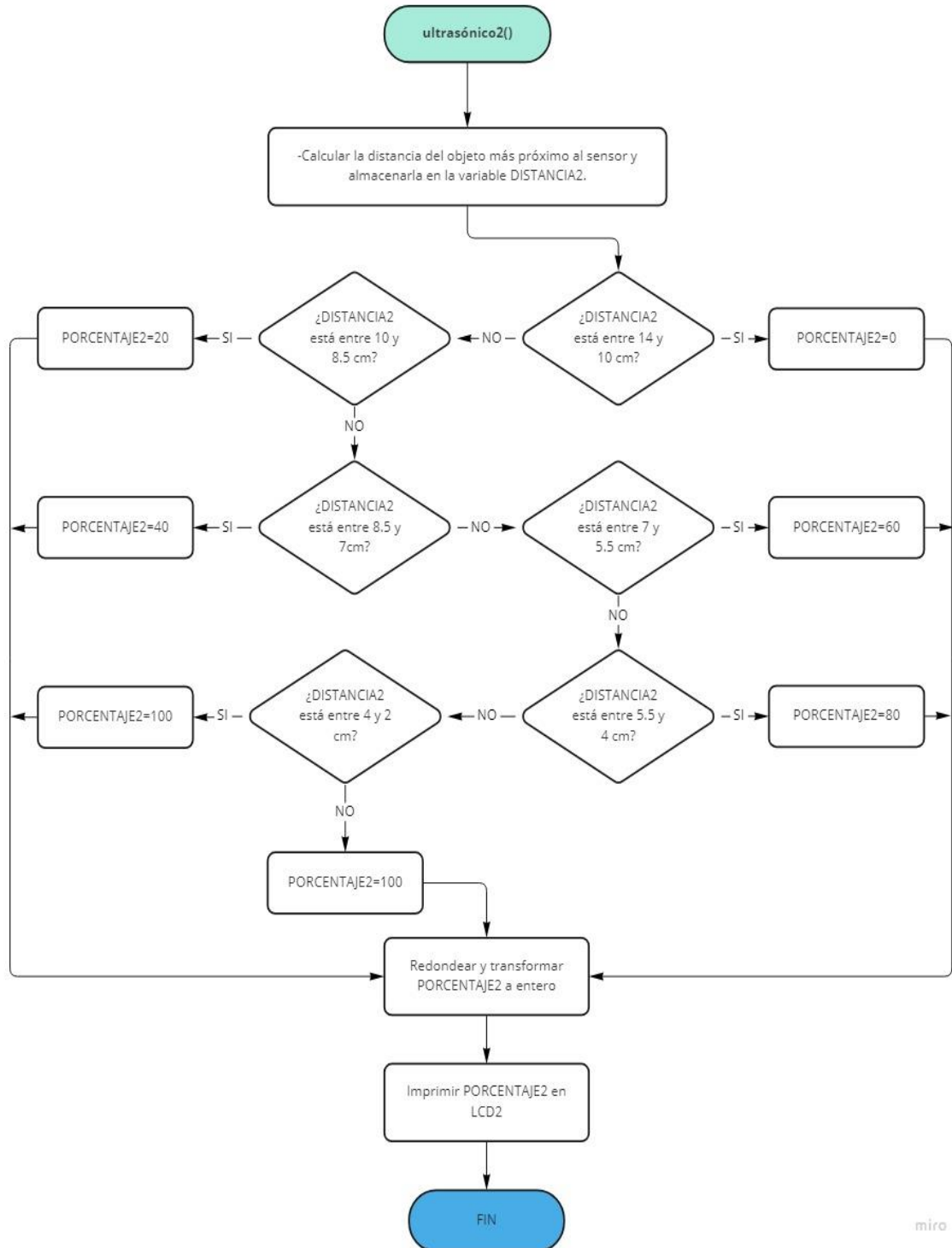


Figura 72. Diagrama de Flujo de la Subrutina ultrasónico1()

Elaborado por: El Investigador

**b. Diagrama de flujo para la Subrutina ultrasónico2()**



**Figura 73.** Diagrama de Flujo de la Subrutina ultrasónico2()

**Elaborado por:** el Investigador

c. Diagrama de flujo para la Subrutina ultrasónico3()

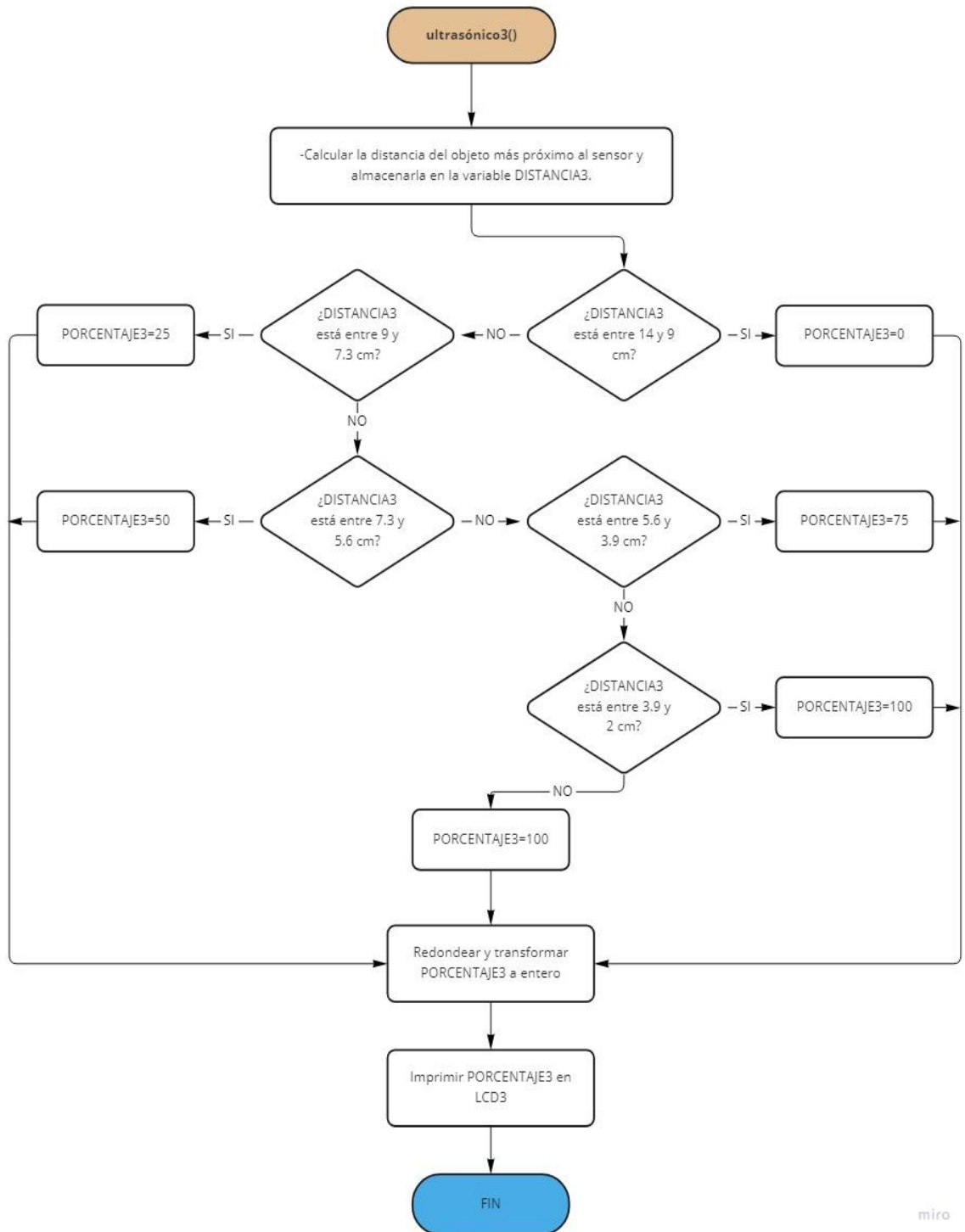


Figura 74. Diagrama de Flujo de la Subrutina ultrasónico3()

Elaborado por: el Investigador

d. Diagrama de flujo para la Subrutina ultrasónico4()

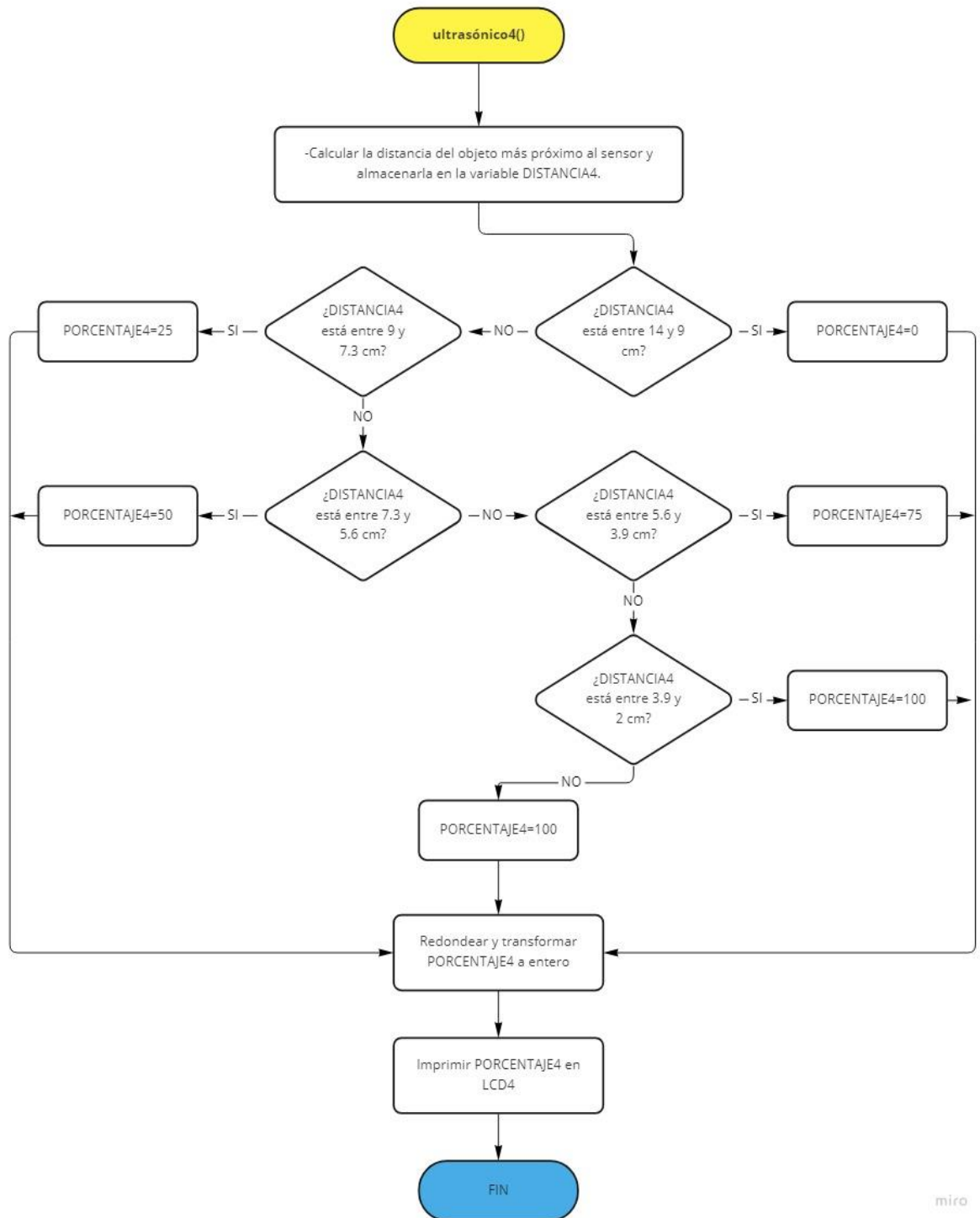
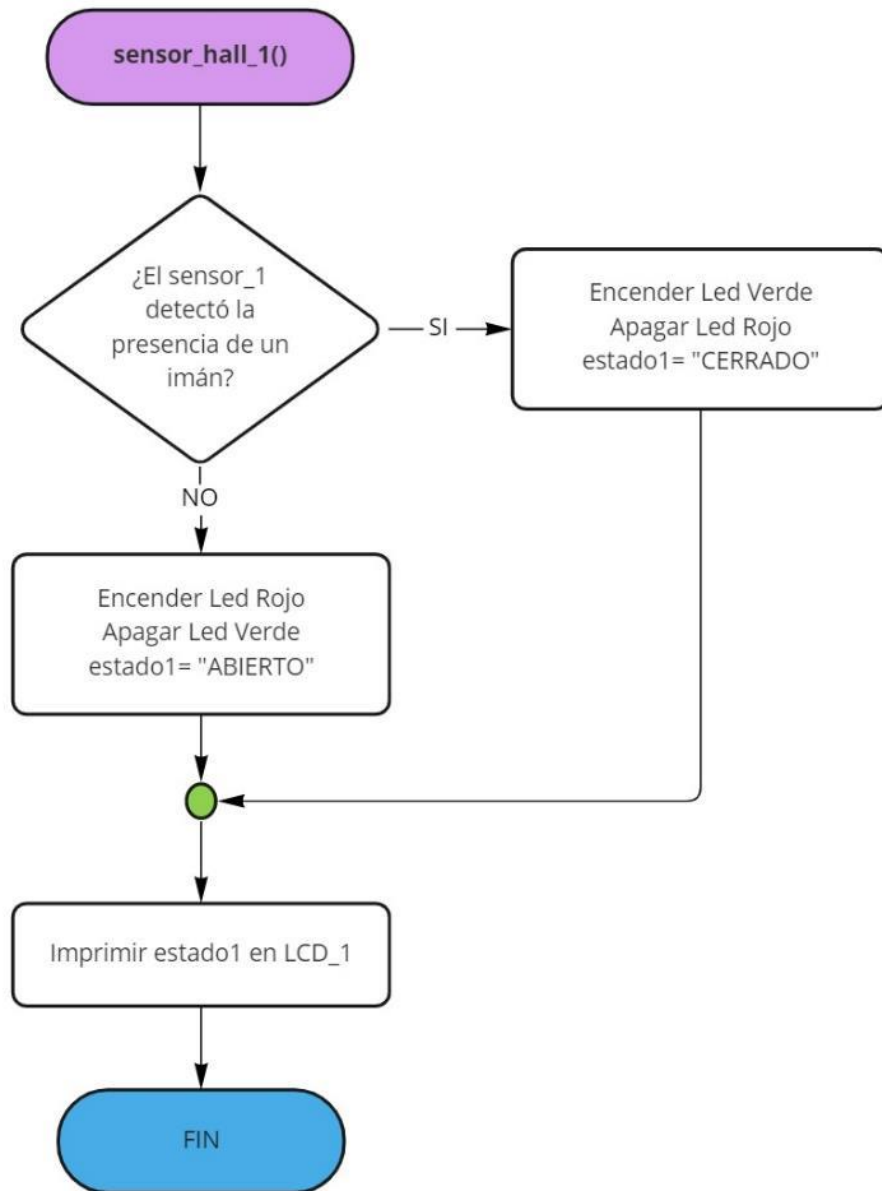


Figura 75. Diagrama de Flujo de la Subrutina ultrasónico4()

Elaborado por: el Investigador

e. Diagrama de flujo para la Subrutina sensor\_hall\_1()

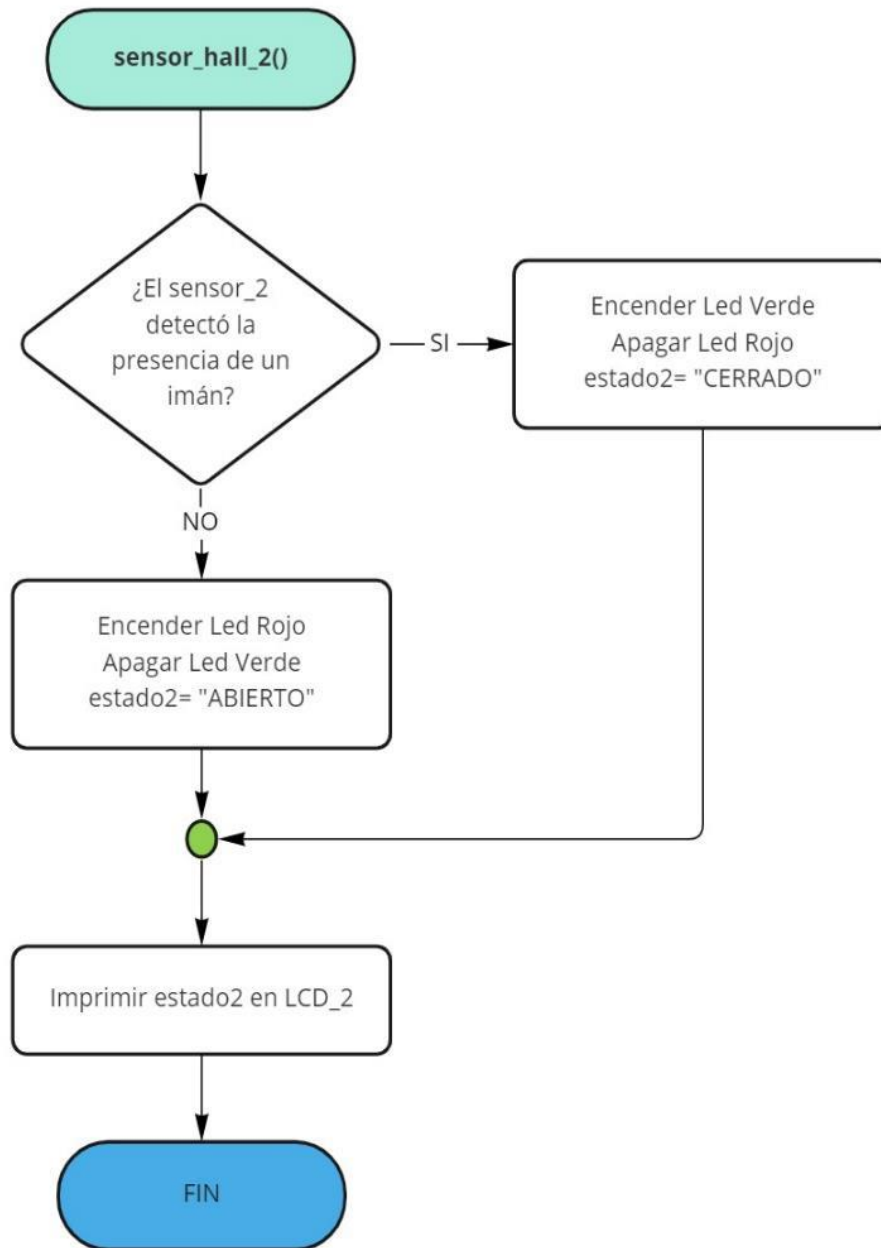


miro

**Figura 76.** Diagrama de Flujo de la Subrutina Sensor\_hall\_1()

**Elaborado por:** el Investigador

f. Diagrama de flujo para la Subrutina sensor\_hall\_2()



**Figura 77.** Diagrama de Flujo de la Subrutina Sensor\_hall\_2()

**Elaborado por:** el Investigador

g. Diagrama de flujo para la Subrutina sensor\_hall\_3()

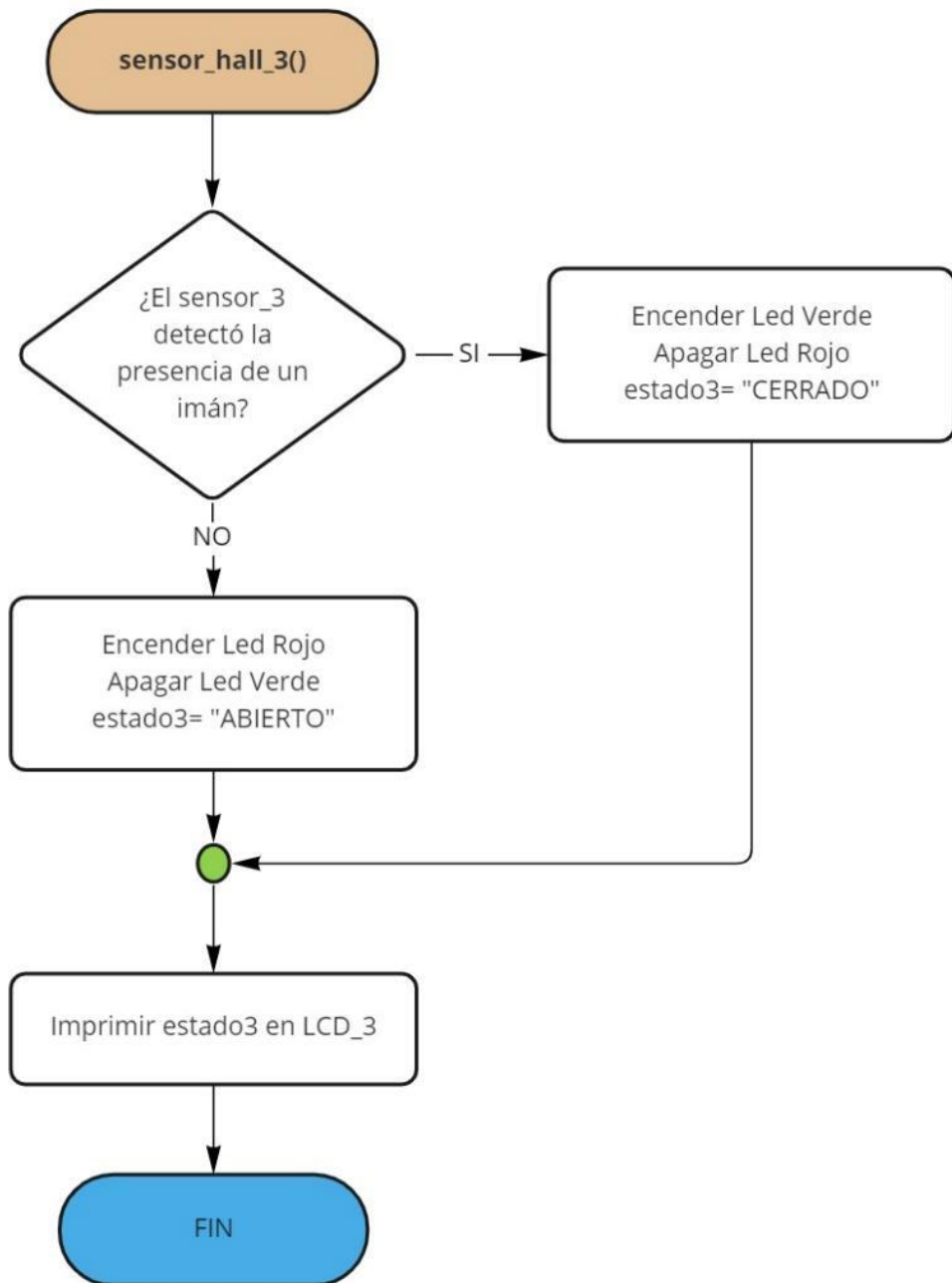
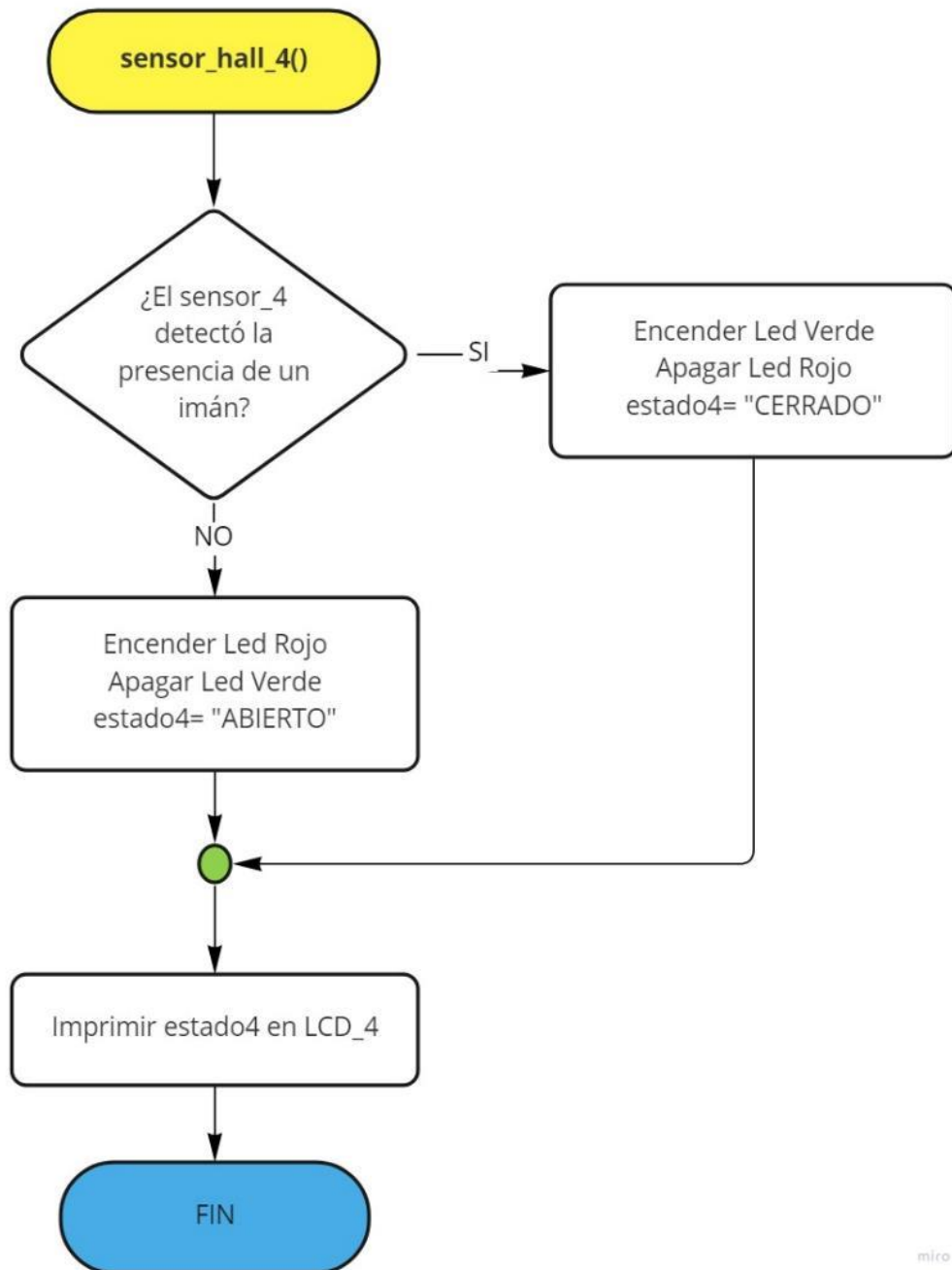


Figura 78. Diagrama de Flujo de la Subrutina Sensor\_hall\_3()

Elaborado por: el Investigador



**h. Diagrama de flujo para la Subrutina sensor\_hall\_4()**



**Figura 79.** Diagrama de Flujo de la Subrutina Sensor\_hall\_4()

**Elaborado por:** el Investigador

i. Diagrama de flujo para la Subrutina Lectura\_Datos()

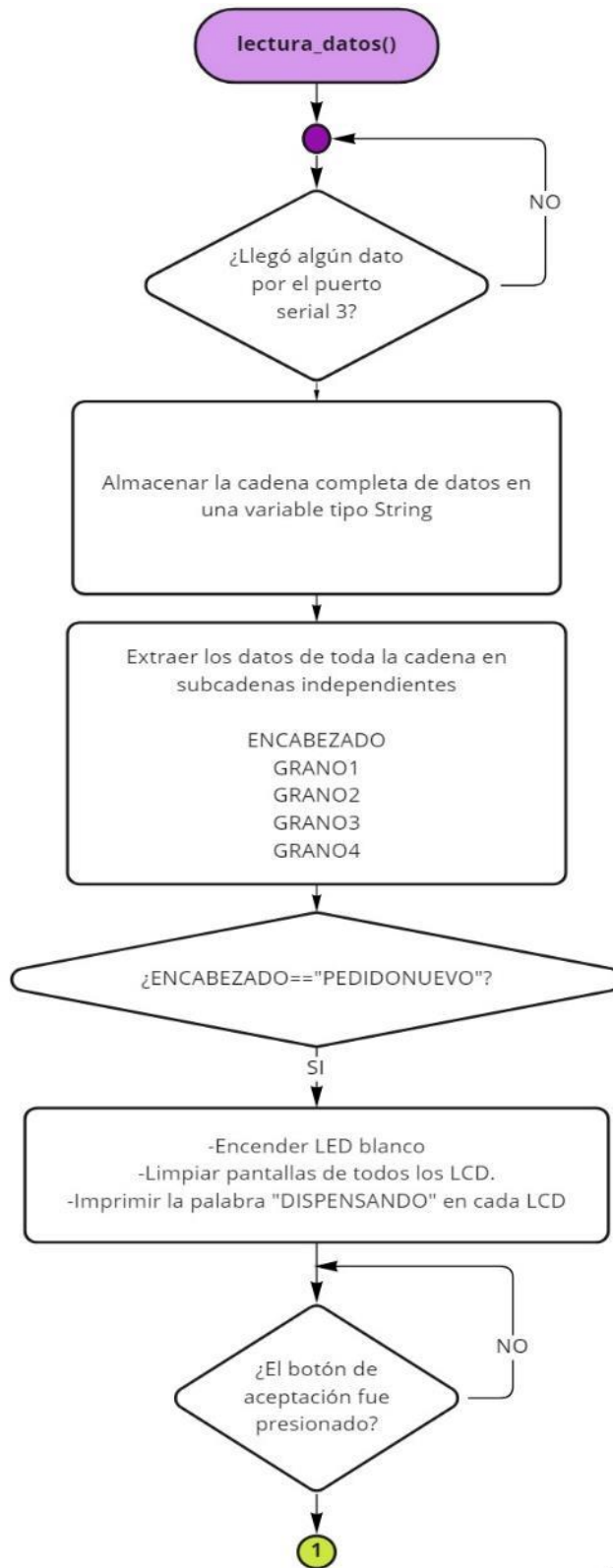
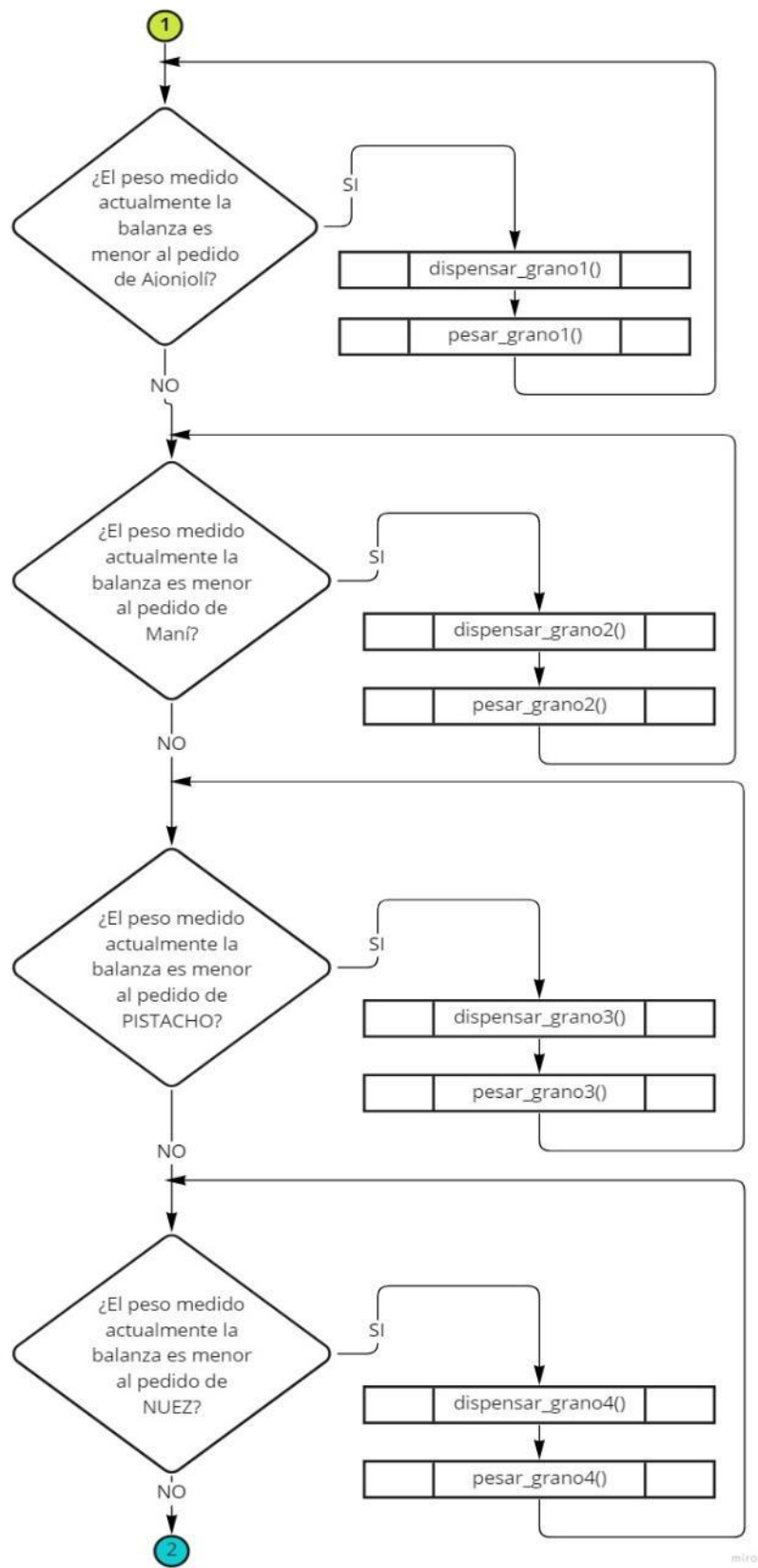


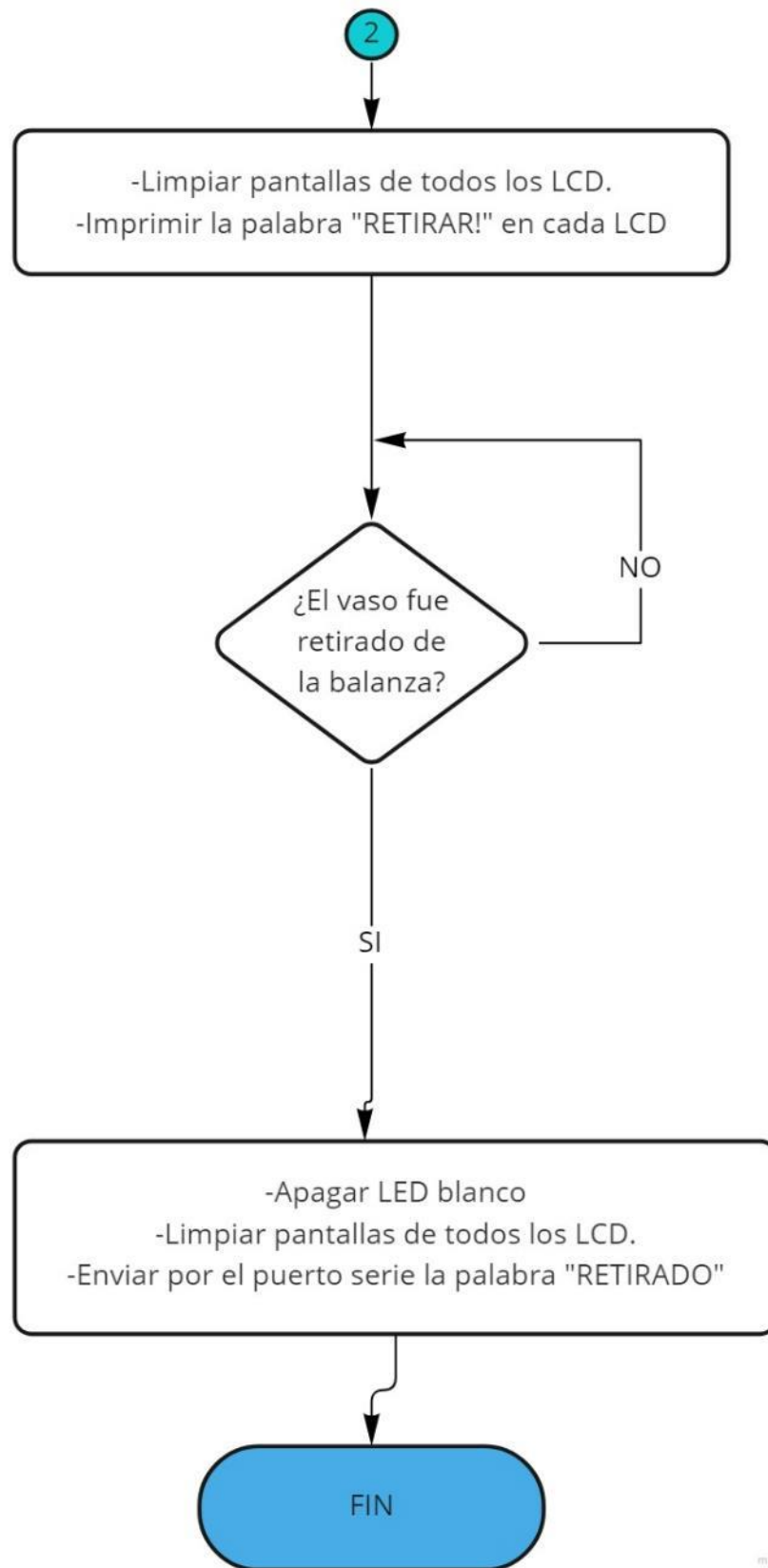
Figura 80. Diagrama de Flujo de la Subrutina Lectura\_datos() – Parte 1/3

Elaborado por: el Investigador



**Figura 81.** Diagrama de Flujo de la Subrutina Lectura\_datos() – Parte 2/3

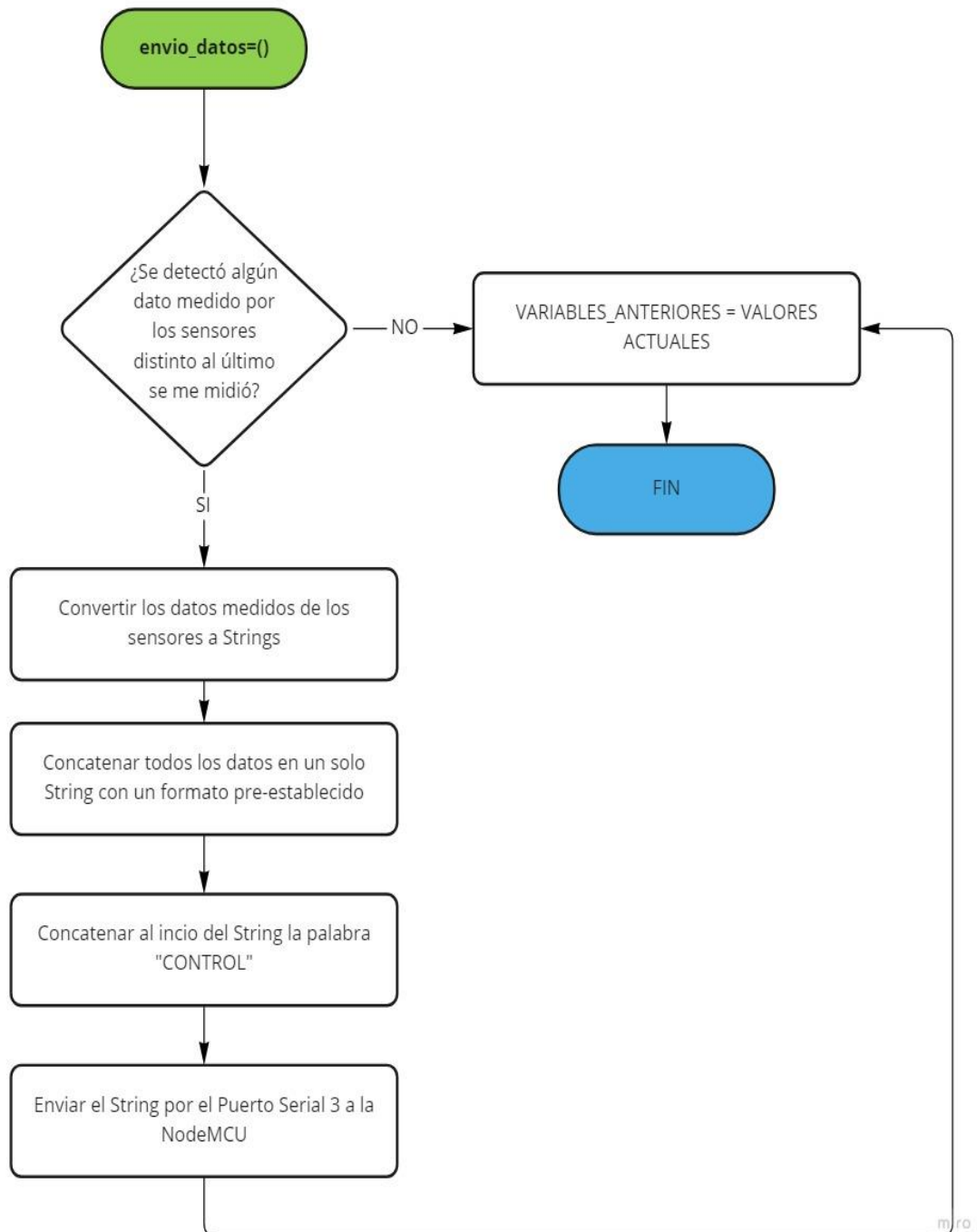
**Elaborado por:** el Investigador



**Figura 82.** Diagrama de Flujo de la Subrutina Lectura\_datos() – Parte 3/3

**Elaborado por:** el Investigador

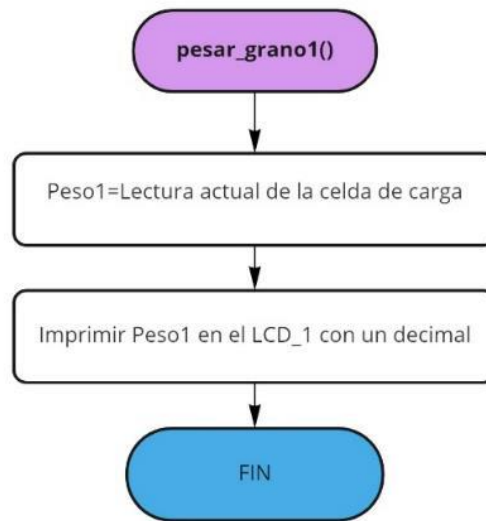
**j. Diagrama de flujo para la Subrutina Envío\_Datos()**



**Figura 83.** Diagrama de Flujo de la Subrutina Envío\_datos()

**Elaborado por:** el Investigador

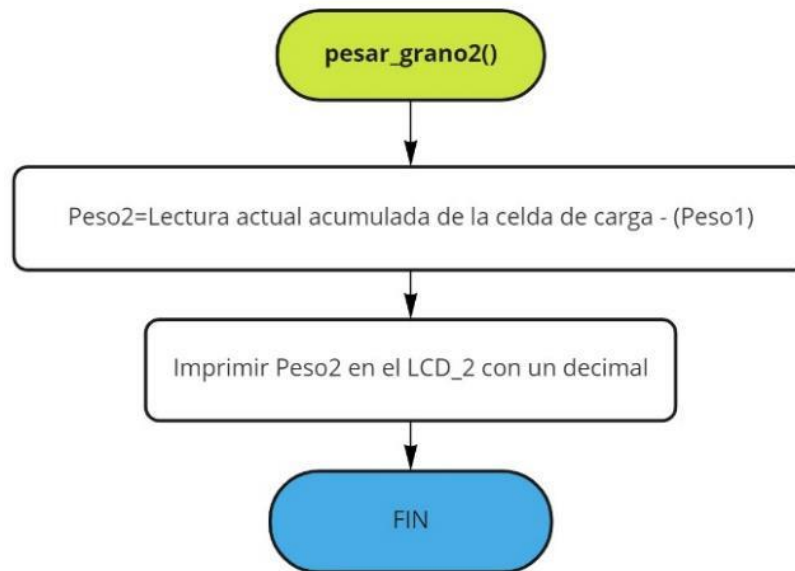
**k. Diagrama de flujo para la Subrutina pesar\_grano1()**



**Figura 84.** Diagrama de Flujo de la Subrutina Pesar\_Grano1()

**Elaborado por:** el Investigador

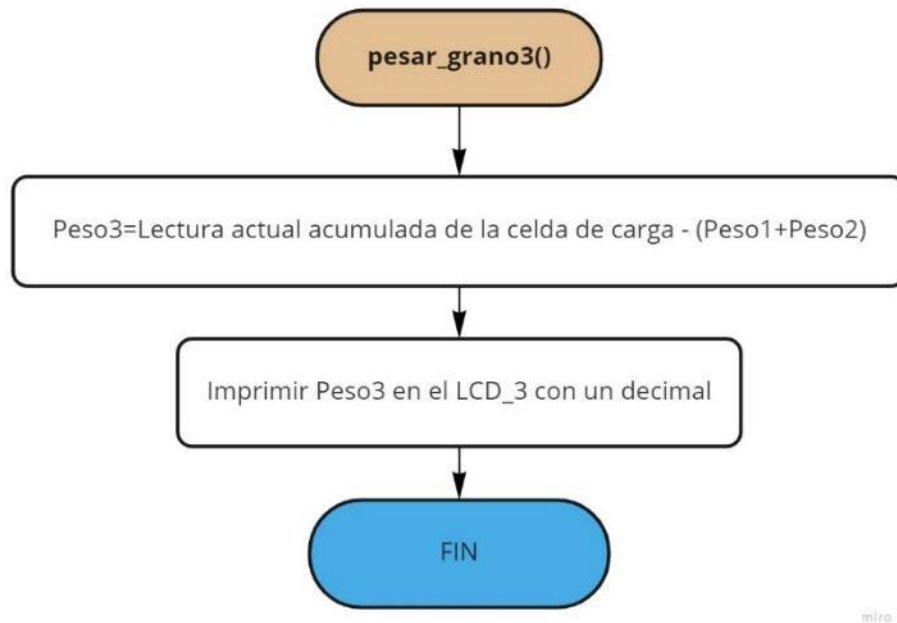
**l. Diagrama de flujo para la Subrutina pesar\_grano2()**



**Figura 85.** Diagrama de Flujo de la Subrutina Pesar\_Grano2()

**Elaborado por:** el Investigador

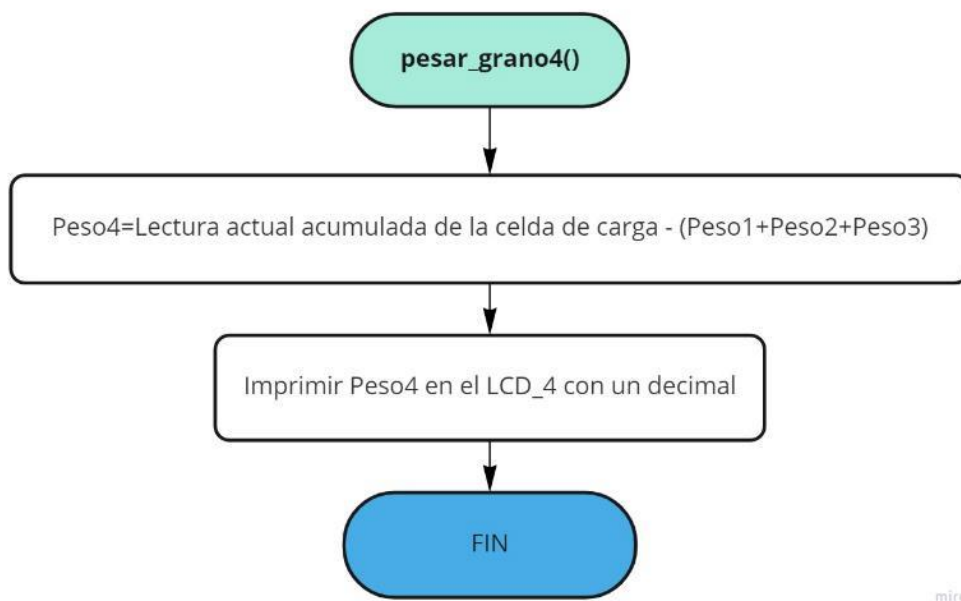
**m. Diagrama de flujo para la Subrutina pesar\_grano3()**



**Figura 86.** Diagrama de Flujo de la Subrutina Pesar\_Grano3()

**Elaborado por:** el Investigador

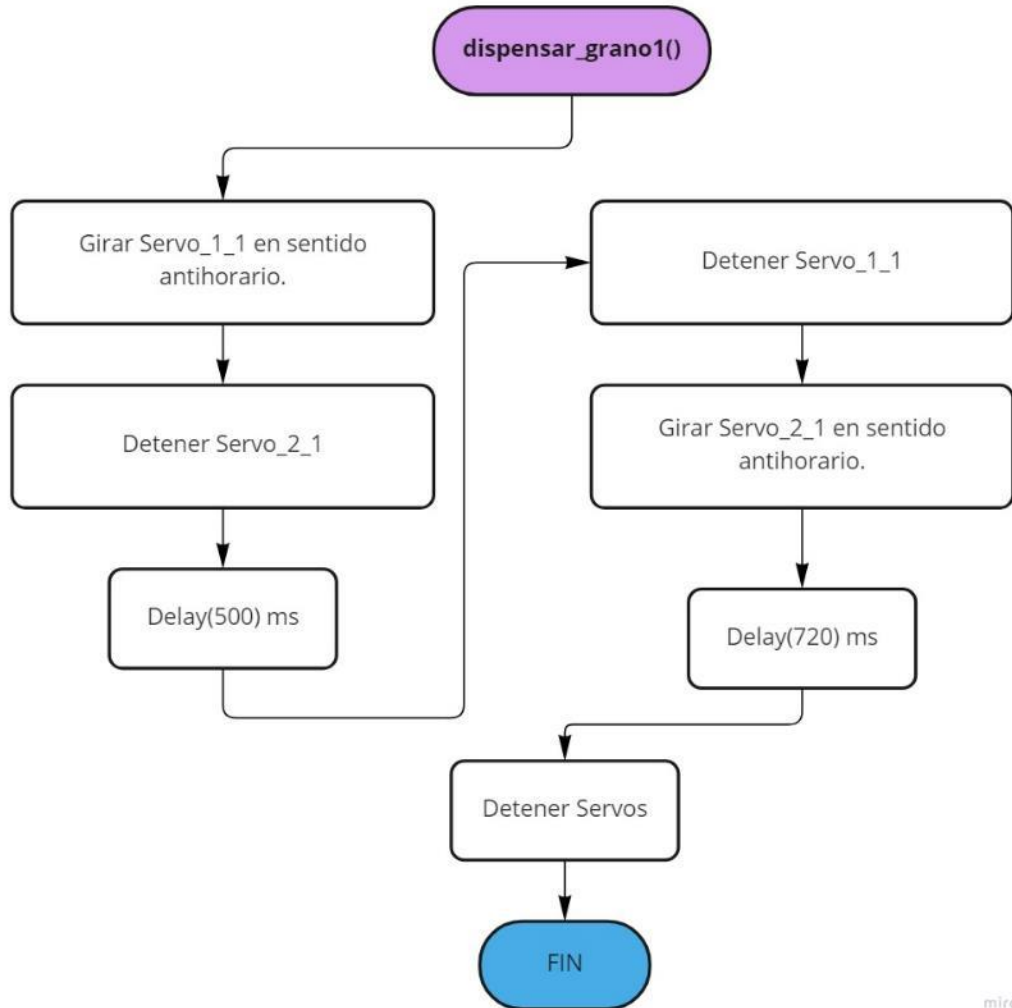
**n. Diagrama de flujo para la Subrutina pesar\_grano4()**



**Figura 87.** Diagrama de Flujo de la Subrutina Pesar\_Grano4()

**Elaborado por:** el Investigador

**o. Diagrama de flujo para la Subrutina Dispensar\_grano1()**



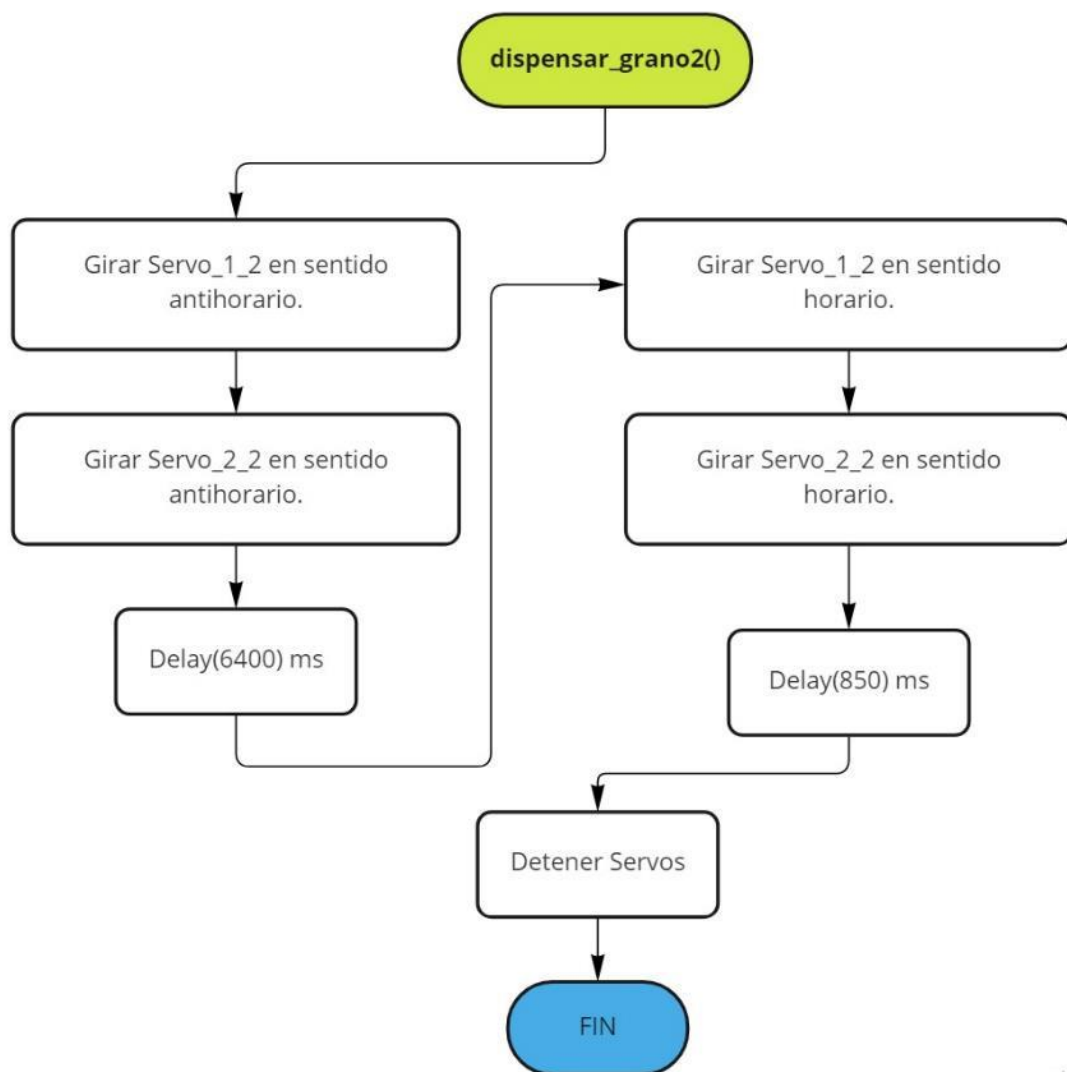
**Figura 88.** Diagrama de Flujo de la Subrutina Dispensar\_Grano1()

**Elaborado por:** el Investigador

**p. Diagrama de flujo para la Subrutina Dispensar\_grano2()**

El diagrama de flujo correspondiente para el algoritmo de programación de la Subrutina Dispensar\_grano2() se muestra en la **Figura 89.**





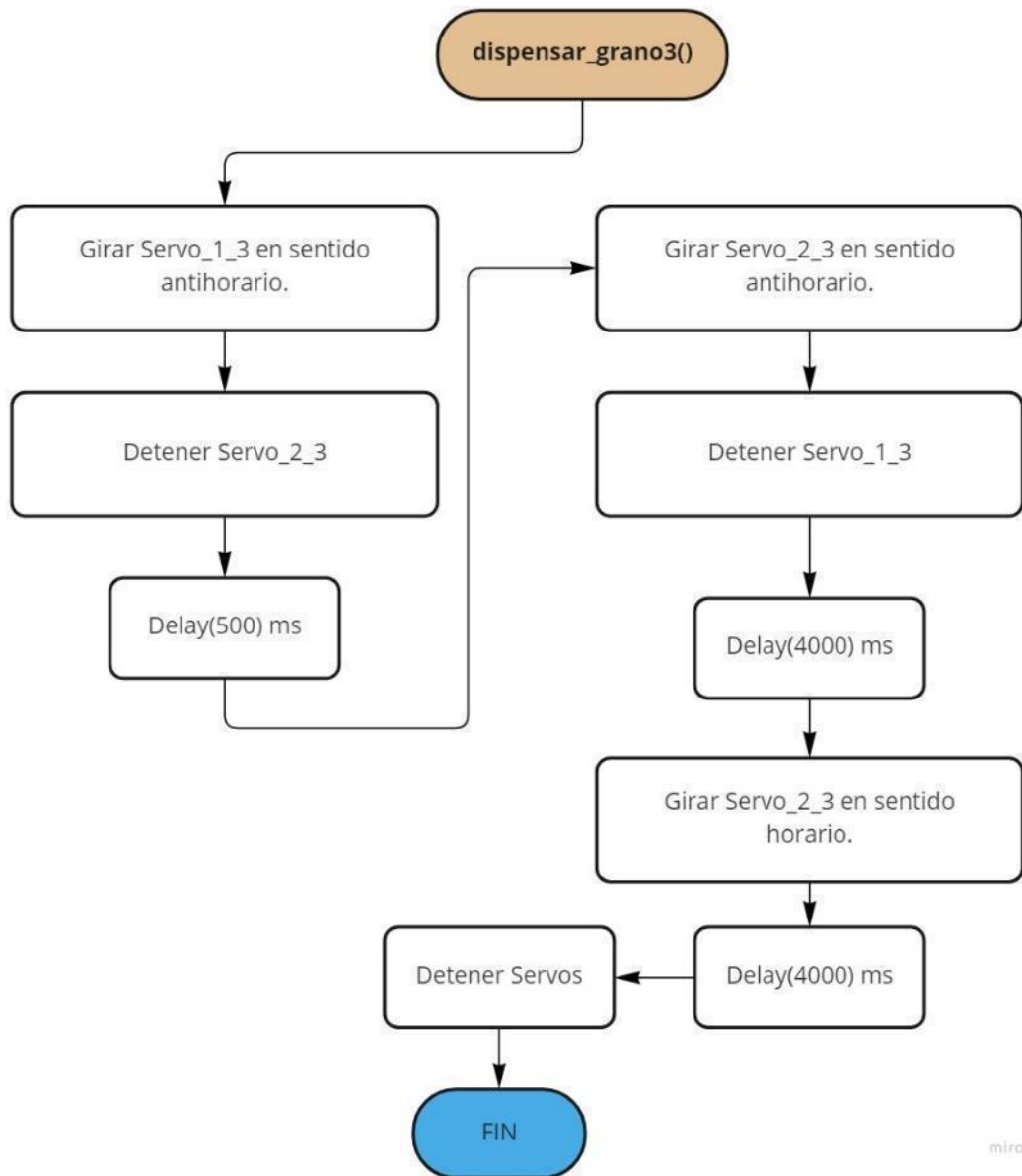
miro

**Figura 89.** Diagrama de Flujo de la Subrutina Dispensar\_Grano2()

**Elaborado por:** el Investigador

**q. Diagrama de flujo para la Subrutina Dispensar\_grano3()**

El diagrama de flujo correspondiente para el algoritmo de programación de la Subrutina Dispensar\_grano3() se muestra en la **Figura 90.**

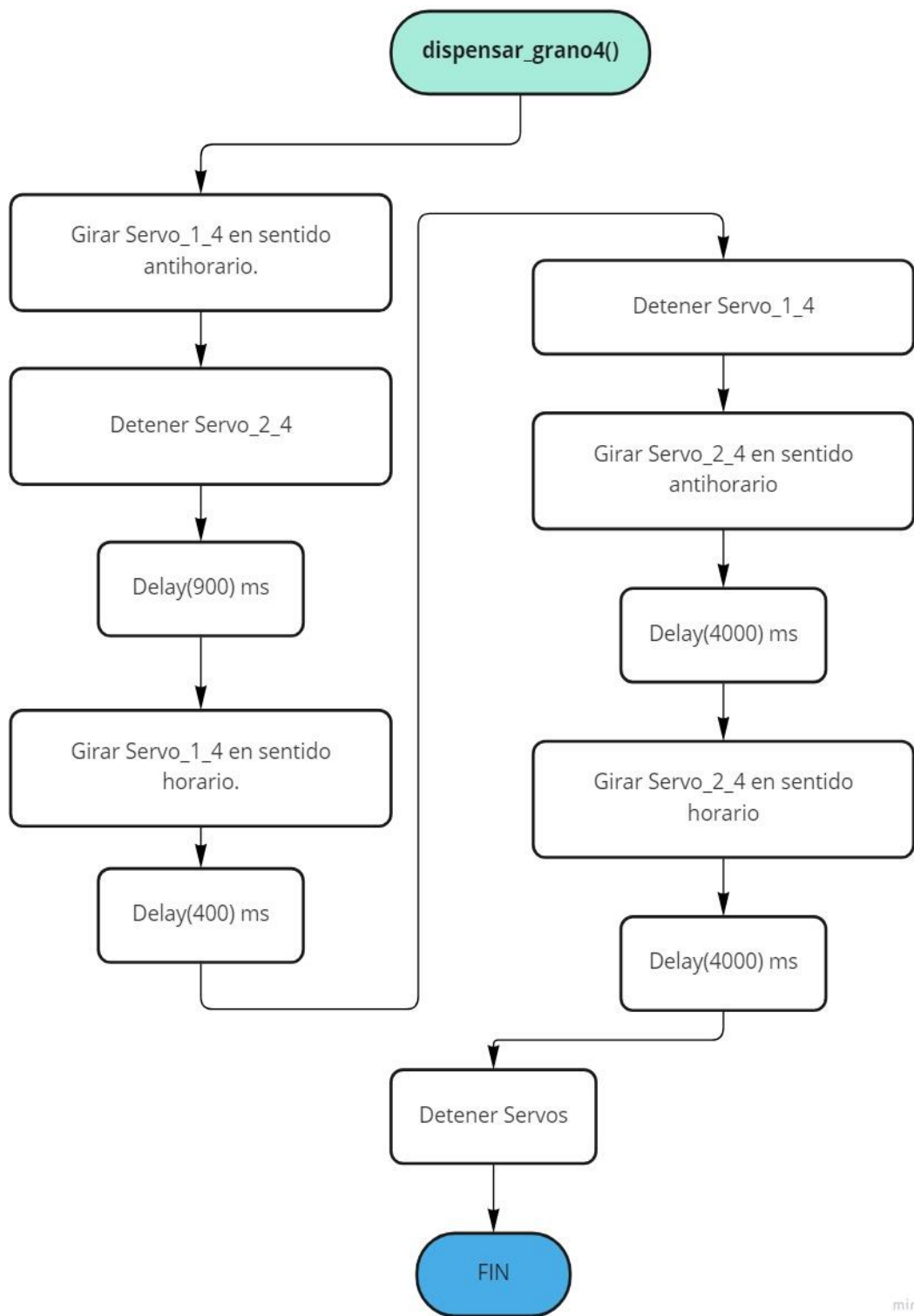


**Figura 90.** Diagrama de Flujo de la Subrutina Dispensar\_Grano3()

**Elaborado por:** el Investigador

**r. Diagrama de flujo para la Subrutina Dispensar\_grano4()**

El diagrama de flujo correspondiente para el algoritmo de programación de la Subrutina Dispensar\_grano4() se muestra en la **Figura 91.**



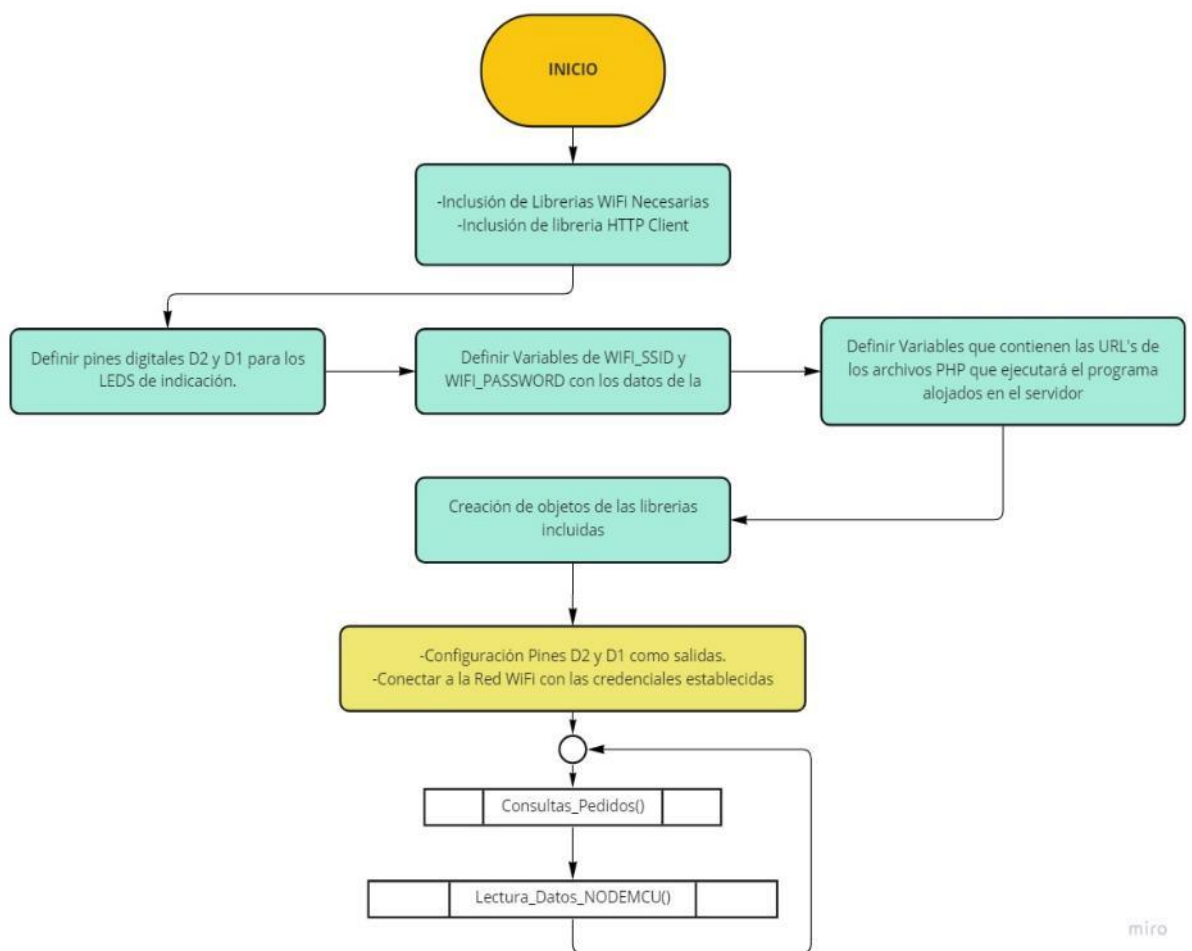
**Figura 91.** Diagrama de Flujo de la Subrutina Dispensar\_Grano4()

**Elaborado por:** el Investigador

El código fuente de programación del Arduino Mega se encuentra descrita en el **Anexo 2** del presente documento.

### 3.2.11 Diagrama de flujo para el Algoritmo de programación de la NodeMCU

En la **Figura 92**, se puede visualizar el diagrama de flujo del algoritmo general programado en la placa NodeMCU ESP8266, posterior a ésta, se detalla el algoritmo de cada subrutina utilizada en distintas figuras respectivamente.



**Figura 92.** Diagrama de Flujo General de la NodeMCU

**Elaborado por:** el Investigador

a. Diagrama de flujo para la Subrutina Consultas\_Pedidos()

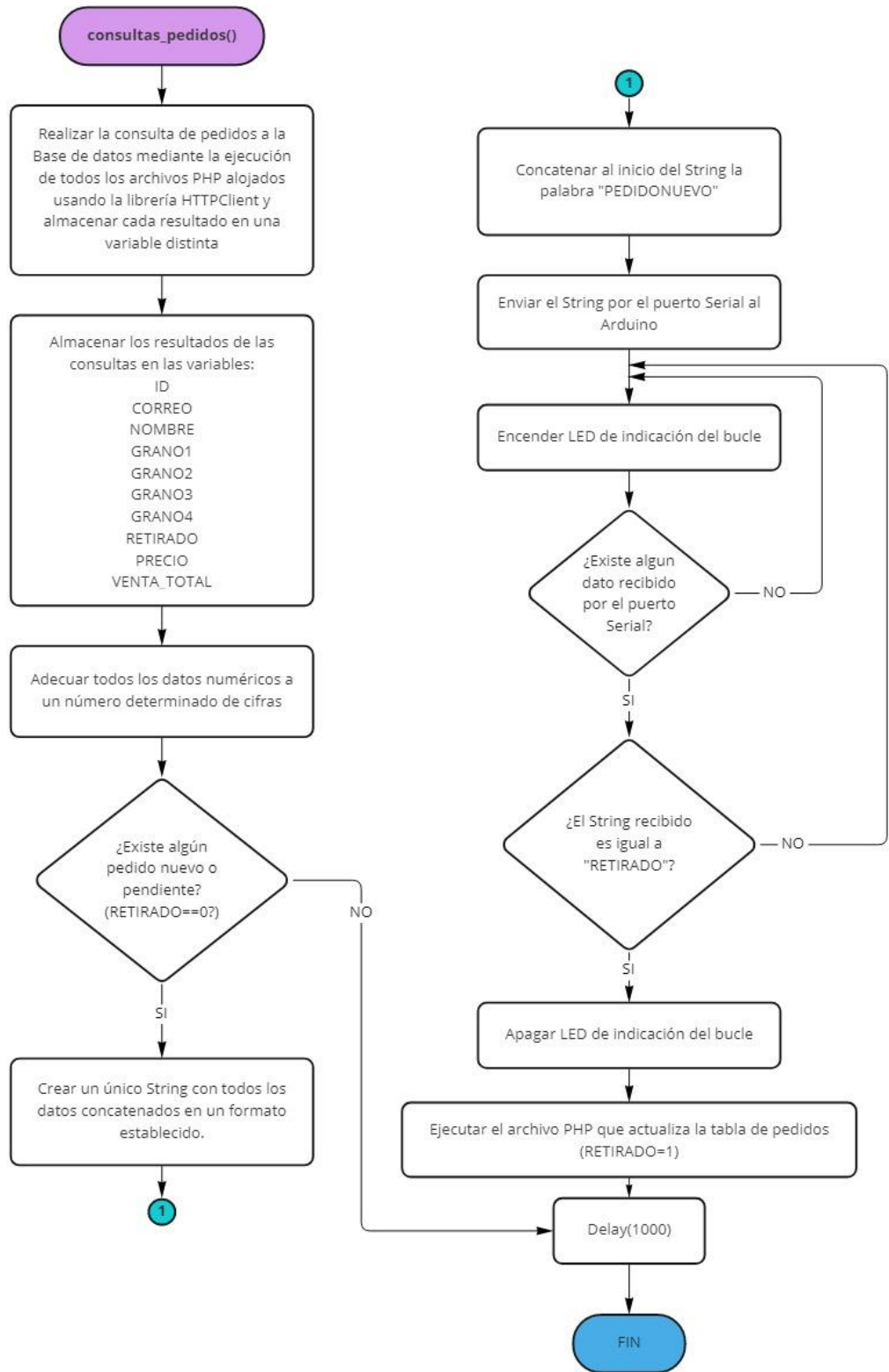
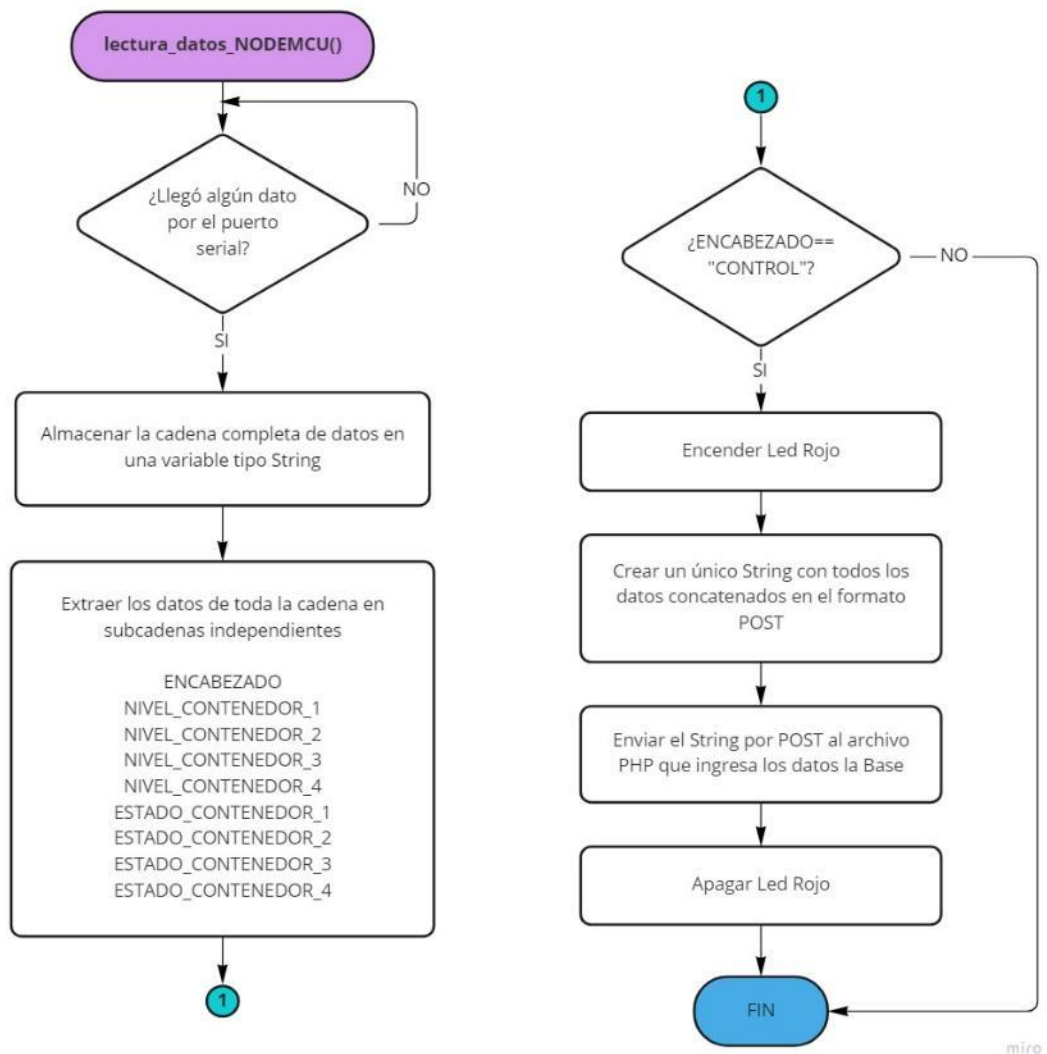


Figura 93. Diagrama de Flujo de la Subrutina Consultas\_Pedidos()

Elaborado por: el Investigador

**b. Diagrama de flujo para la Subrutina Lectura\_Datos\_NODEMCU()**



**Figura 94.** Diagrama de Flujo de la Subrutina Lectura\_Datos\_NODEMCU()

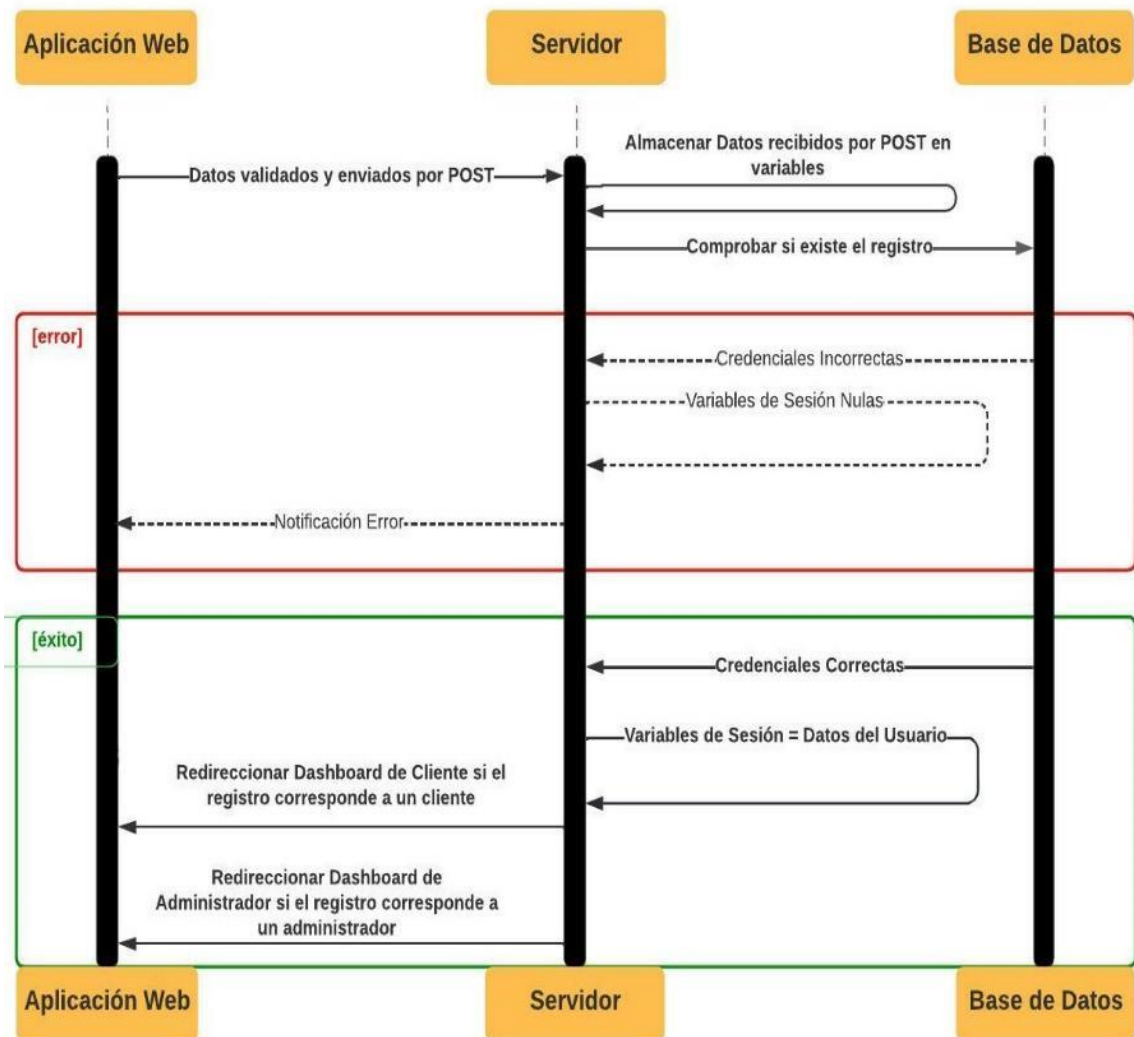
**Elaborado por:** el Investigador

El código fuente de programación del microcontrolador NodeMCU se encuentra descrito en el **Anexo 3** del presente documento.

### 3.2.12 Desarrollo de la Interfaz Web de Inicio de Sesión

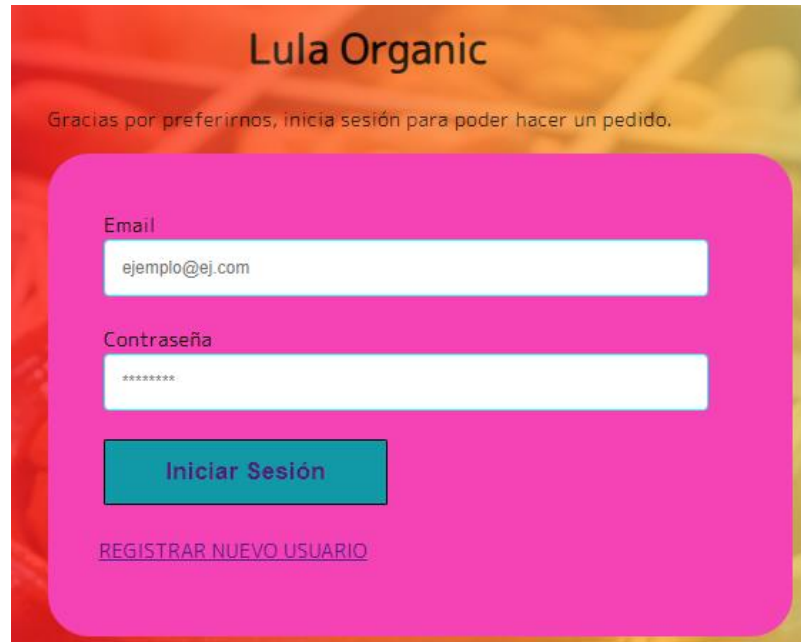
Esta corresponde a la interfaz web que se direcciona por defecto al acceder al servidor, permite el acceso a las Interfaces Web de administrador o de cliente según correspondan las credenciales ingresadas, la misma que, funciona estableciendo valores en variables de tipo SESSION de PHP, cabe destacar que por medio de estas variables, se controla el acceso a las distintas páginas web del proyecto, es decir, un

usuario registrado como cliente jamás podrá acceder a ninguna página web correspondiente a administradores, además, para poder acceder a cualquier sitio web, es estrictamente necesario un previo inicio de Sesión. En la **Figura 95** se muestra un diagrama de secuencia del Back-End programado para el sitio mencionado, mientras que en las **Figuras 96 y 97** se muestra su Interfaz Web. El código Fuente se lo puede observar en el **Anexo 4**.



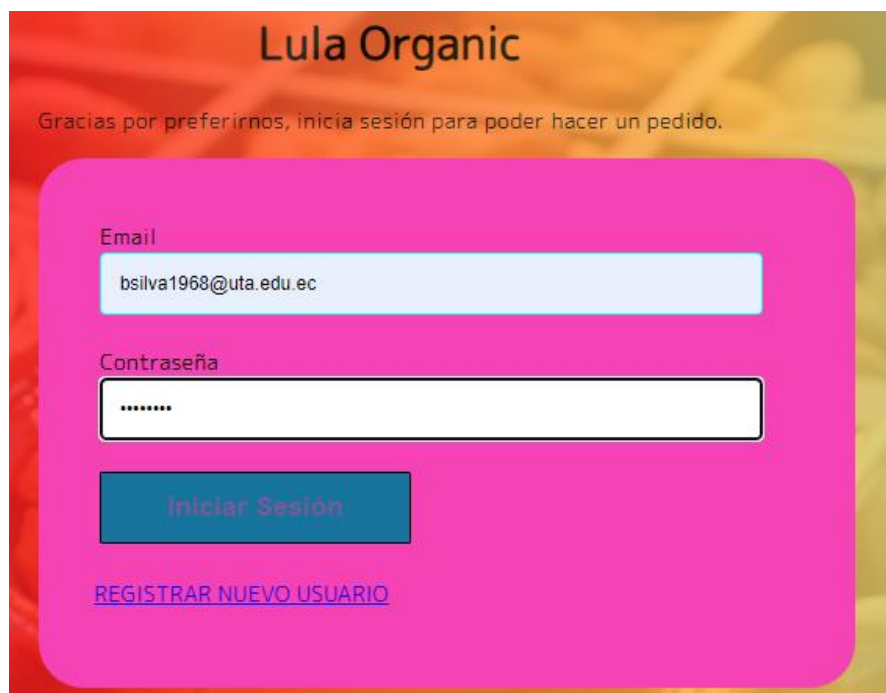
**Figura 95.** Diagrama de Secuencia “Log In”

**Elaborado por:** el Investigador



**Figura 96.** Interfaz Web “Log In”

**Elaborado por:** el Investigador



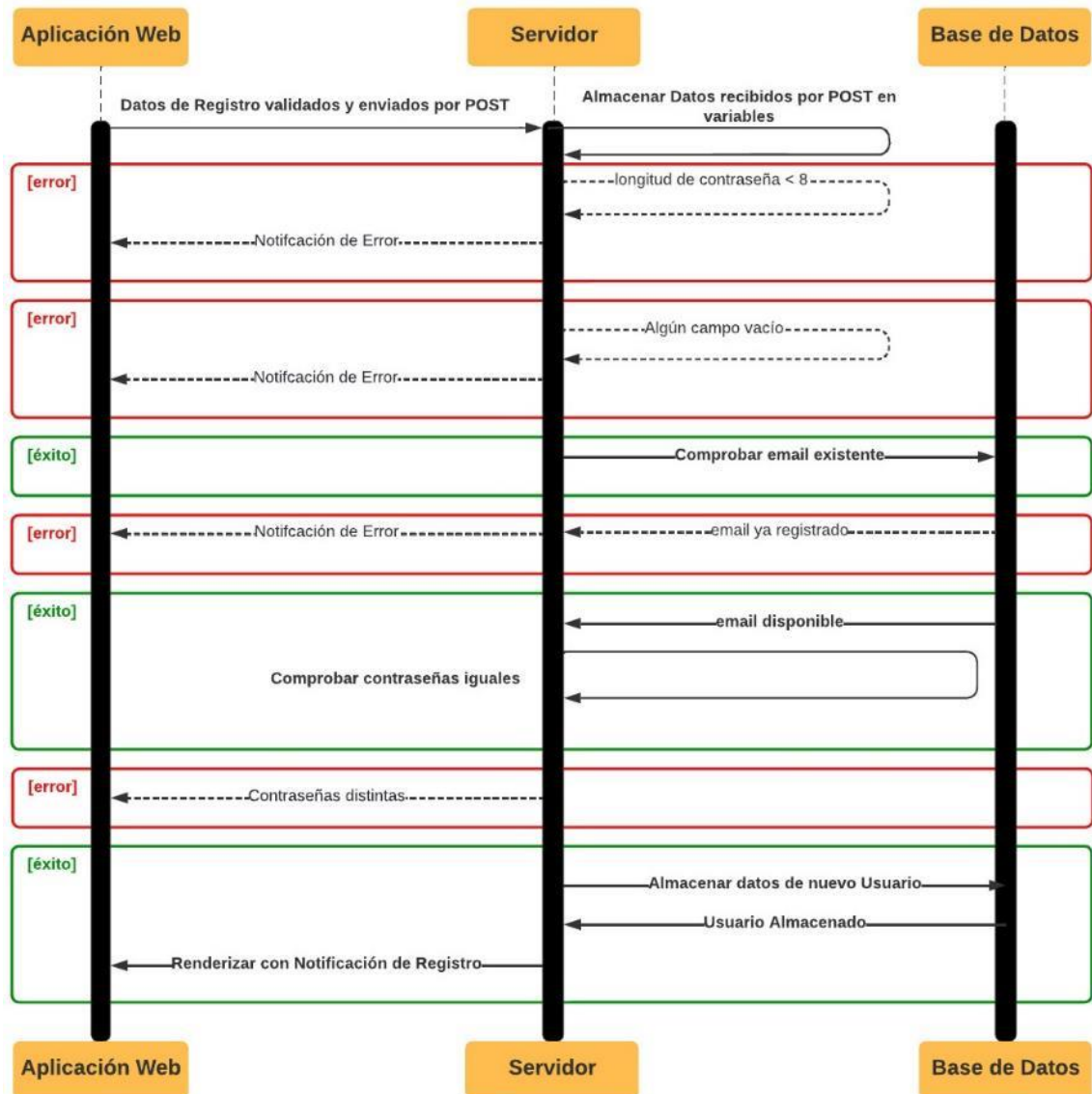
**Figura 97.** Interfaz Web “Log In”

**Elaborado por:** el Investigador



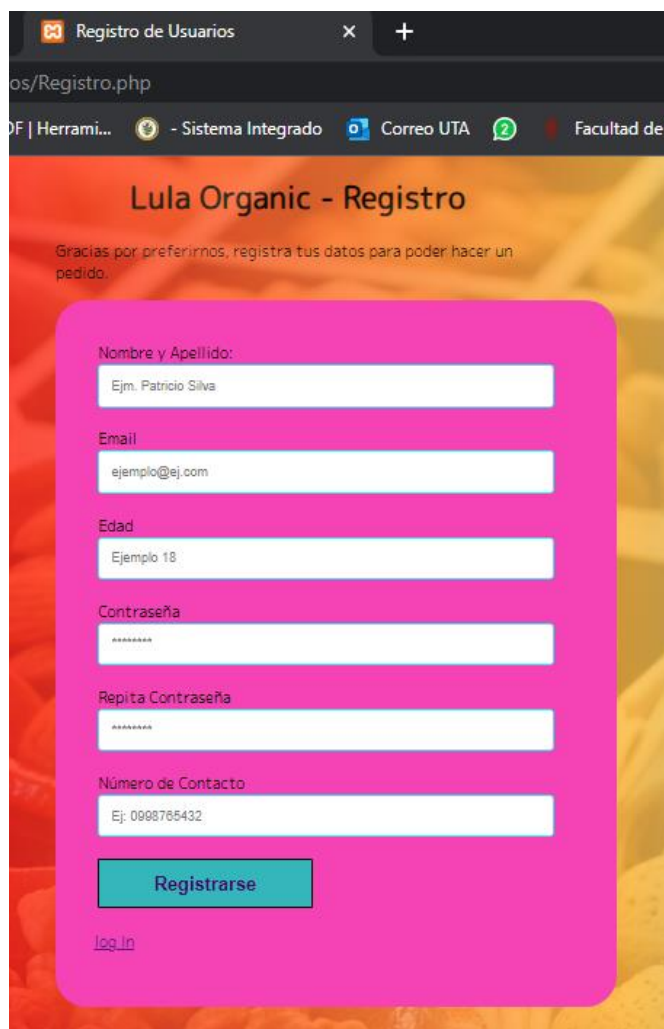
### 3.2.13 Desarrollo de la Interfaz Web de Registro de Usuarios

Esta Interfaz web permite el registro de usuarios nuevos, todos los usuarios registrados por esta página web se los considera de tipo “cliente” por defecto. En la **Figura 98**, se muestra un diagrama de secuencia del Back-End programado para el sitio mencionado, mientras que en la **Figura 99** se muestra su Interfaz Web. El código Fuente se lo puede observar en el **Anexo 5**.



**Figura 98.** Diagrama de Secuencia “Registrar Usuario”

**Elaborado por:** el Investigador



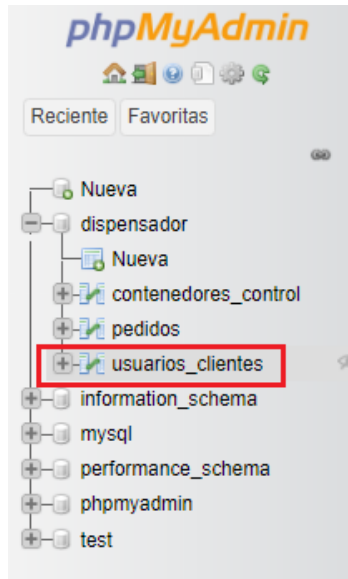
**Figura 99.** Interfaz Web “Registrar Usuario”

**Elaborado por:** el Investigador

Todas las páginas web del presente proyecto cuentan con una opción de Cerrar Sesión, su programación Web se encuentra detallada en el **Anexo 6**.

### **3.2.14 Registro de un Administrador**

Para poder registrar un nuevo administrador, es necesario que el usuario se registre primero como si fuese un cliente normal, posterior a esto se accede a la base de datos utilizando PhpMyAdmin y en el atributo del tipo de usuario se lo cambia de “cliente” a “admin”, de esta manera, el usuario beneficiado ya podrá acceder automáticamente a la Interfaz Web de Administración cada vez que inicie sesión. En las **Figuras 100 y 101**, se ilustra el proceso mencionado.



**Figura 100.** Registro de un Administrador

**Elaborado por:** el Investigador

The image shows the phpMyAdmin interface displaying the 'usuarios\_clientes' table. A red arrow points to the 'admin' user in the table. The table has columns: id, fecha, nombre, correo, password, edad, telefono, ultimo acceso, and tipo.

|                          | id | fecha               | nombre         | correo                  | password                                  | edad | telefono  | ultimo acceso    | tipo    |
|--------------------------|----|---------------------|----------------|-------------------------|---|------|-----------|------------------|---------|
| <input type="checkbox"/> | 1  | 2022-06-13 01:11:14 | Marcos Erazo   | merazo9003@uta.edu.ec   | 7c222fb2927d828af22f592134e89332480637c0d | 25   | 998141177 | 2022-06-16 17:20 | cliente |
| <input type="checkbox"/> | 2  | 2022-06-13 01:11:35 | Kenneth Silva  | kenneth@gmail.com       | 7c222fb2927d828af22f592134e89332480637c0d | 21   | 992747485 | 2022-07-14 42:57 | cliente |
| <input type="checkbox"/> | 3  | 2022-06-13 01:11:58 | Patricio Silva | bsilva1968@uta.edu.ec   | 7c222fb2927d828af22f592134e89332480637c0d | 26   | 960651514 | 2022-07-21 01:59 | admin   |
| <input type="checkbox"/> | 4  | 2022-07-03 22:19:04 | Soraya Naranjo | naranjosoraya@gmail.com | 7372c359f733b6dfe68b1700e91dd1d7173bc71a  | 48   | 998348029 | 2022-07-22 19:50 | cliente |

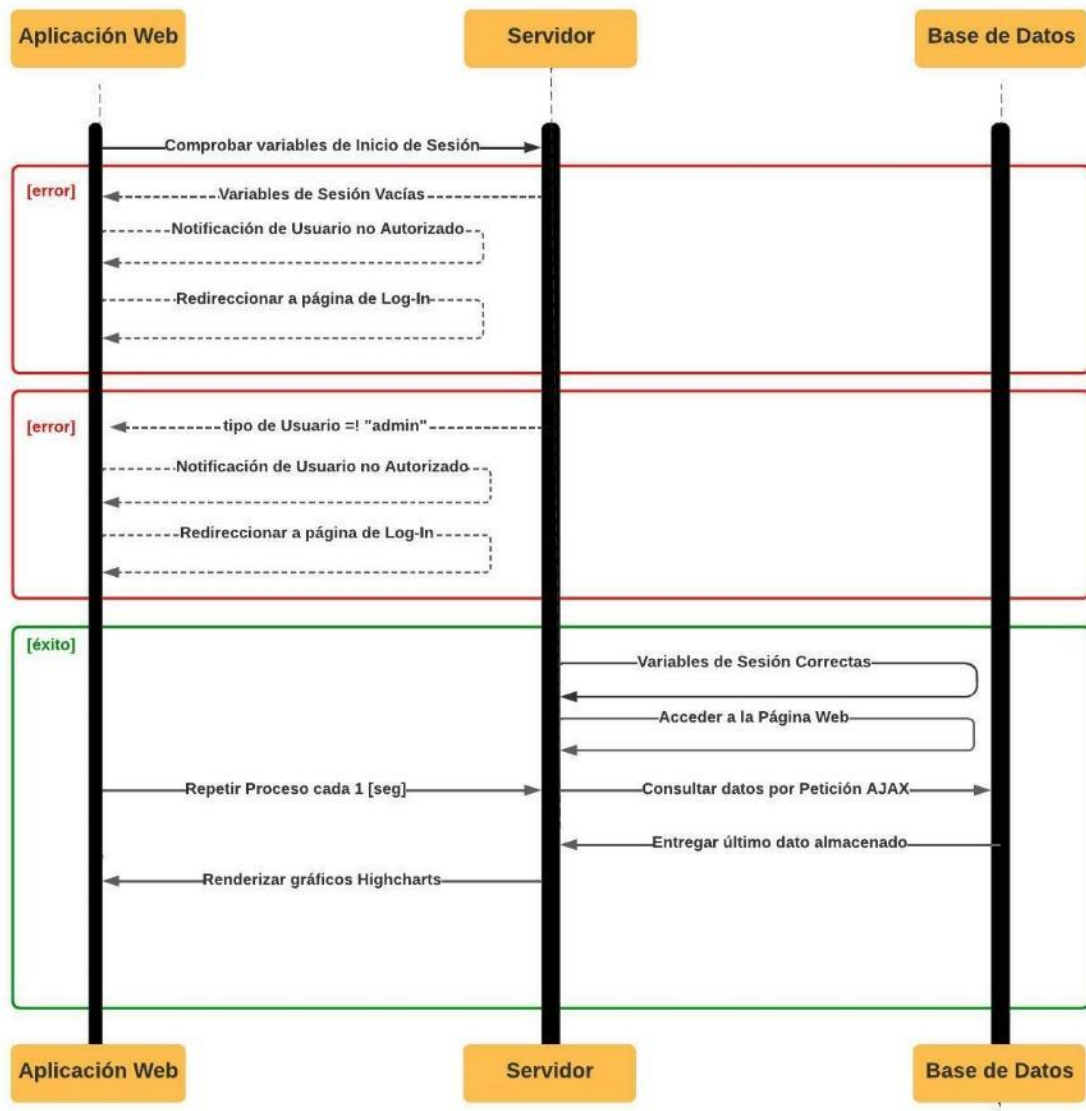
**Figura 101.** Registro de un Administrador

**Elaborado por:** el Investigador

### 3.2.15 Desarrollo de la Interfaz Web de Administrador

La interfaz Web de Administrador consta de múltiples páginas web relacionadas entre sí, estas siguen tres algoritmos distintos de programación web según sea el caso. Para

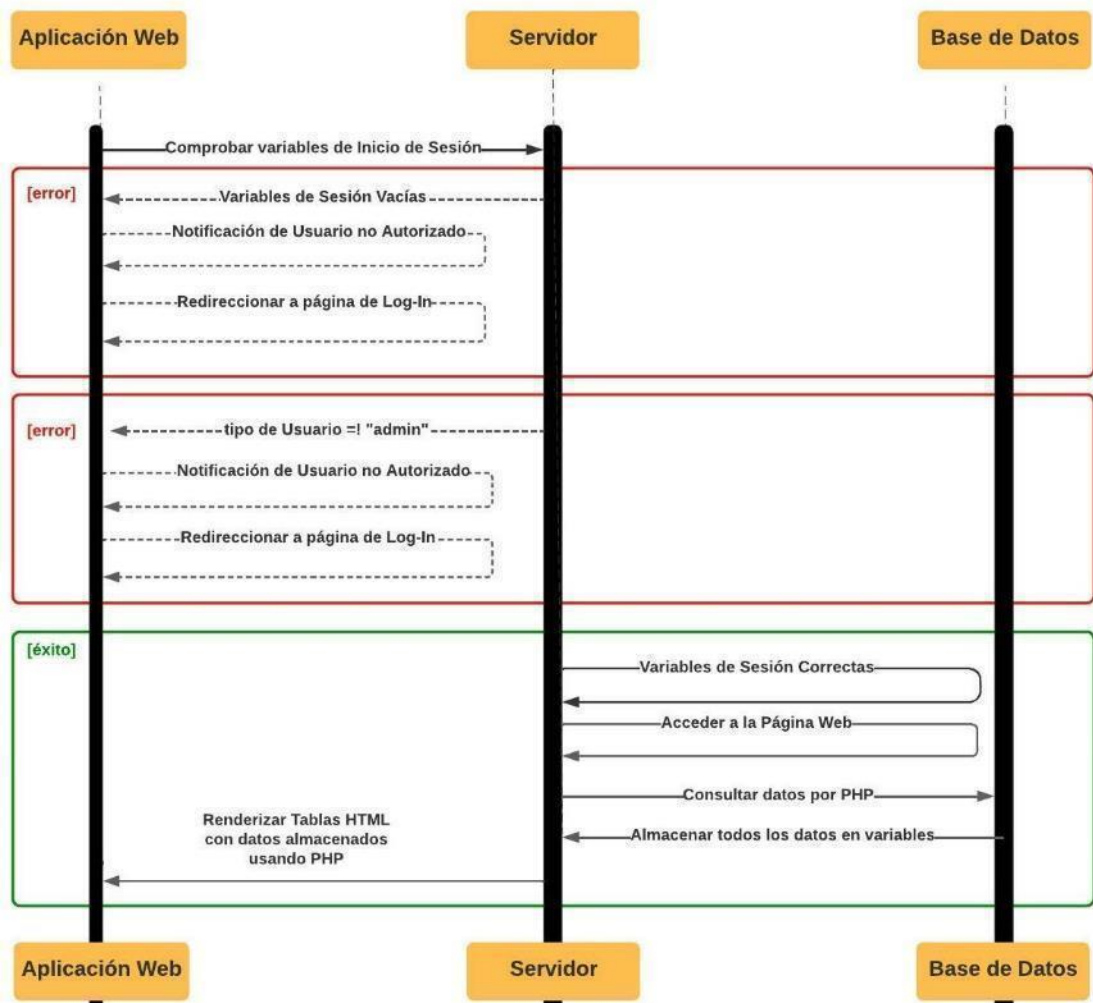
una mejor apreciación de los algoritmos, se presentan figuras con éstos representados en diagramas secuenciales. En la **Figura 102**, se muestra un diagrama secuencial para el caso de las páginas web que poseen únicamente gráficos dinámicos que se muestran datos en tiempo real.



**Figura 102.** Diagrama de Secuencia de las Páginas Dinámicas de Administrador

**Elaborado por:** el Investigador

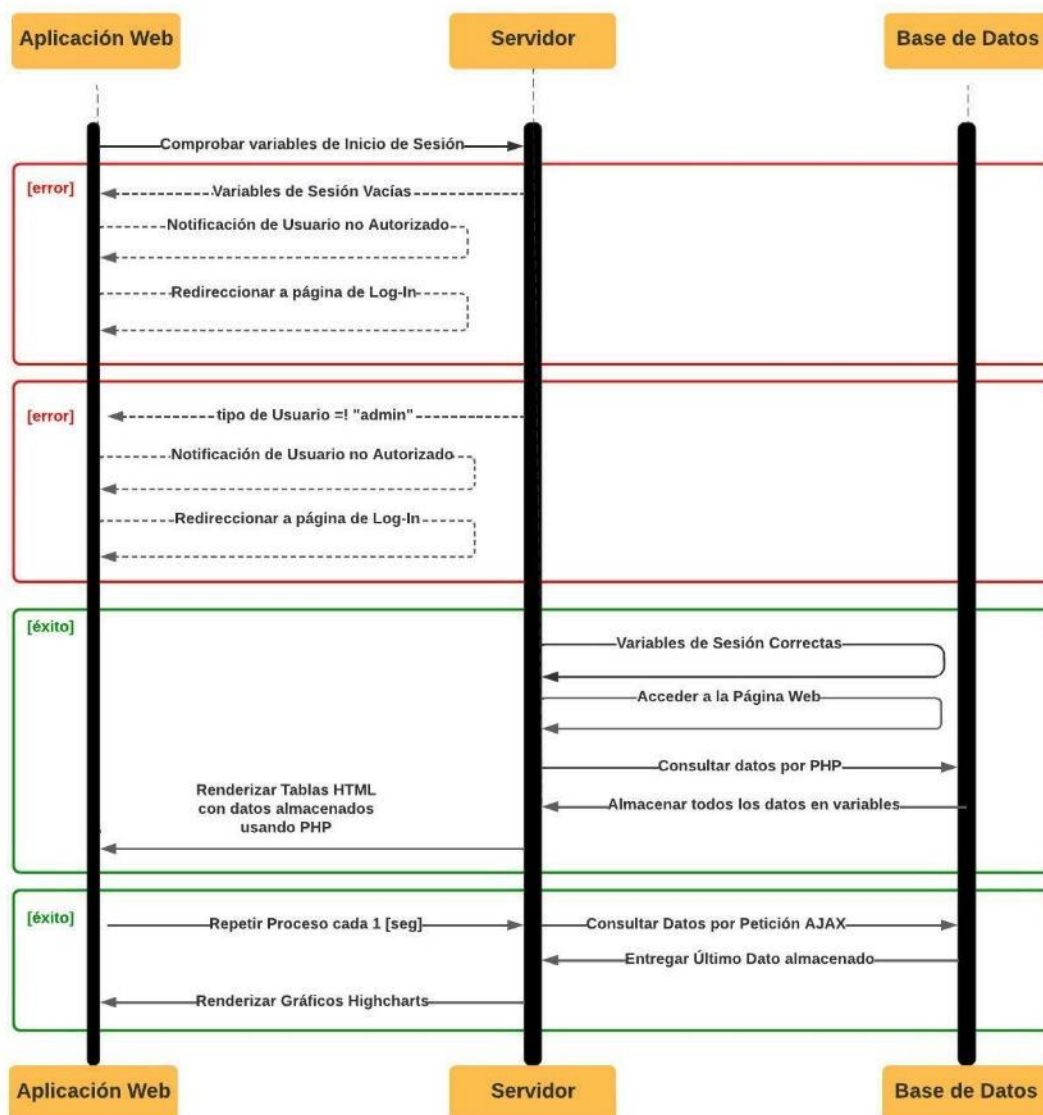
Por otra parte, en la **Figura 103**, se muestra el diagrama secuencial para el caso de las páginas web que muestran datos estáticos realizando una única consulta en una sola renderización.



**Figura 103.** Diagrama de Secuencia de las Páginas Estáticas de Administrador

**Elaborado por:** el Investigador

Mientras que, en la **Figura 104.** se muestra el diagrama secuencial para el caso de las páginas web que presentan en conjunto tanto gráficos en tiempo real, como datos estáticos. Cabe mencionar que, para que los datos estáticos presentados en tablas se actualicen, es necesario refrescar la página web, mientras que, para los gráficos dinámicos, este proceso no es necesario.



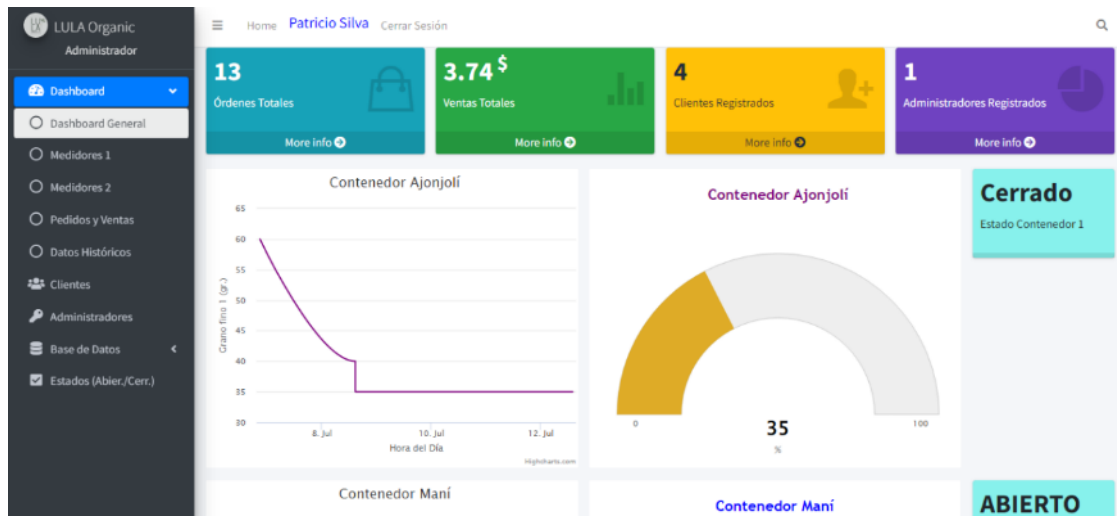
**Figura 104.** Diagrama de Secuencia de las Páginas Mixtas de Administrador

**Elaborado por:** el Investigador

### Presentación de Páginas Web para el Usuario Administrador

En la **Figura 105**, se presenta el diseño de la página web correspondiente al Dashboard General de Administrador, en ésta se representa un entorno de control general de todos los cuatro contenedores de frutos secos, representando sus datos en gráficos dinámicos que se actualizan en tiempo real según el último dato registrado en la Base de datos, además, se puede visualizar datos de interés como el número de Pedidos totales,

número de usuarios registrados y el monto total de las ventas dispensadas por el prototipo.



**Figura 105.** Dashboard General de Administrador

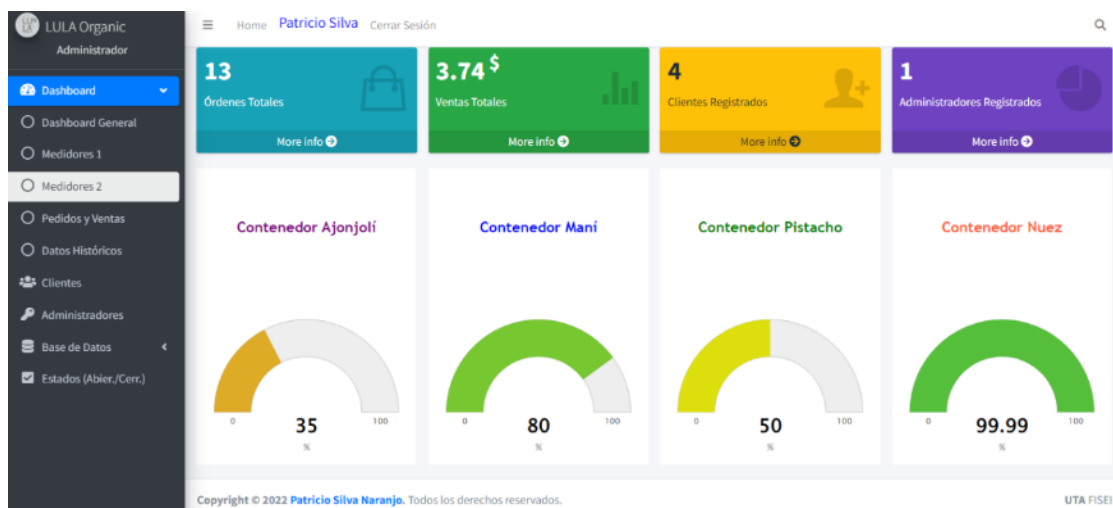
**Elaborado por:** el Investigador

El segundo y tercer Dashboard presentados en las **Figuras 106 y 107** respectivamente, corresponden a páginas web dinámicas que se actualizan, de igual forma, en tiempo real, con la única diferencia que presentan distintos gráficos de highcharts en cada una, con la finalidad de que el usuario escoja el medio de visualización que mejor se adapte a sus necesidades.



**Figura 106.** Segundo Dashboard de Administrador

**Elaborado por:** el Investigador



**Figura 107.** Tercer Dashboard de Administrador

**Elaborado por:** el Investigador

La siguiente página web corresponde a una presentación de datos estáticos en forma de Tabla, correspondientes a los pedidos y ventas registradas en la base de Datos. Dicha página se la puede observar en la **Figura 108.**

The table displays the following data:

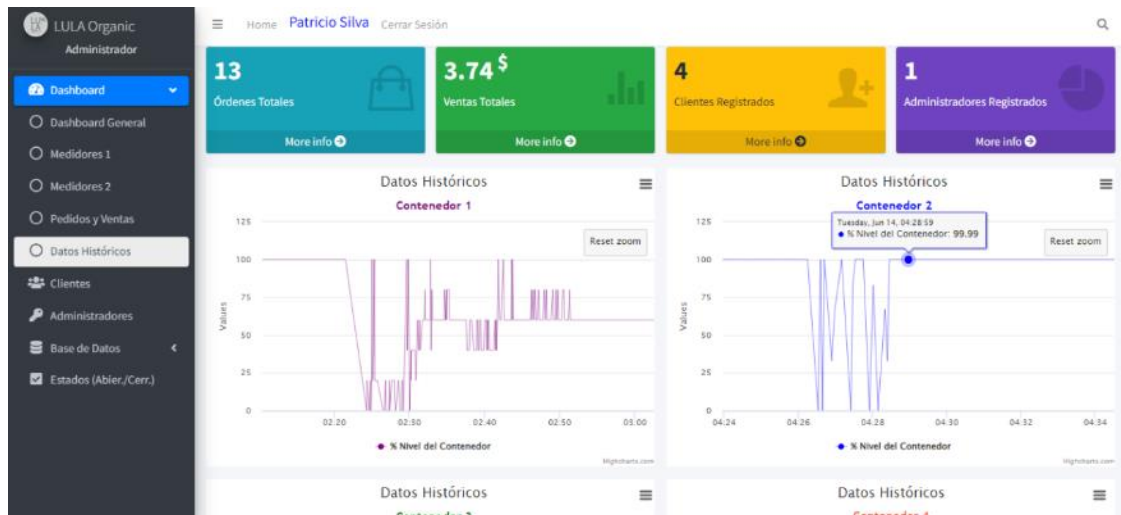
| ID | Fecha               | Correo            | Nombre        | Ajonjoli (gr.) | Mani (gr.) | Grano 3 (gr.) | Grano 4 (gr.) | Precio (\$ USD) | Venta acumulad |
|----|---------------------|-------------------|---------------|----------------|------------|---------------|---------------|-----------------|----------------|
| 1  | 2022-06-21 23:41:26 | abcd@gmail.com    | sda           | 2              | 0          | 0             | 0             | 1               | 1              |
| 2  | 2022-06-21 23:52:47 | kenneth@gmail.com | Kenneth Silva | 2              | 6          | 0             | 0             | 0.07            | 1.07           |
| 3  | 2022-06-21 23:56:33 | abcd@gmail.com    | asd3          | 3              | 0          | 0             | 0             | 1               | 2              |
| 4  | 2022-06-22 00:01:54 | sdf               | sdf           | 3              | 0          | 4             | 0             | 1               | 1              |
| 5  | 2022-06-22 00:03:44 | sdf               | sdf           | 0              | 4          | 4             | 0             | 1               | 1              |
| 6  | 2022-07-01 02:22:40 | afadsf            | dfsdf         | 4              | 3          | 0             | 0             | 1               | 1              |
| 7  | 2022-07-01 02:51:26 | kenneth@gmail.com | Kenneth Silva | 6              | 14         | 0             | 0             | 0.17            | 1.17           |
| 8  | 2022-07-01 05:19:38 | kenneth@gmail.com | Kenneth Silva | 9              | 15         | 10            | 0             | 0.5             | 1.67           |
| 9  | 2022-07-01 09:41:57 | kenneth@gmail.com | Kenneth Silva | 20             | 10         | 10            | 10            | 0.6             | 2.27           |

**Figura 108.** Dashboard “Pedidos y Ventas” de Administrador

**Elaborado por:** el Investigador

La página web correspondiente a Datos históricos presenta gráficos dinámicos de Highcharts que indican todos los datos almacenados de cada contenedor desde la fecha que empezó a funcionar el sistema hasta la actualidad, de esta forma, el administrador o propietario puede tener un mayor conocimiento sobre los casos que se presentaron en fechas pasadas.





**Figura 109.** Dashboard “Datos Históricos” de Administrador

**Elaborado por:** el Investigador

Se desarrollan dos páginas web destinadas a la visualización de los datos de los usuarios registrados por medio de tablas. En la **Figura 110**, se muestra la página de visualización de clientes, mientras que en la **Figura 111**, se observa la página de visualización de administradores.

| ID | Fecha               | Nombre         | Correo                      | Edad | Teléfono  | Últi |
|----|---------------------|----------------|-----------------------------|------|-----------|------|
| 1  | 2022-06-13 01:11:14 | Marcos Erazo   | merazo9003@uta.edu.ec       | 25   | 998141177 | 2022 |
| 2  | 2022-06-13 01:11:35 | Kenneth Silva  | kenneth@gmail.com           | 21   | 992747485 | 2022 |
| 4  | 2022-07-03 22:19:04 | Soraya Naranjo | naranjosoraya@gmail.com     | 48   | 998348029 | 2022 |
| 5  | 2022-07-03 22:30:41 | Danna Silva    | silva.dannalorena@gmail.com | 18   | 998348029 | 2022 |

**Figura 110.** Dashboard “Clientes Registrados” de Administrador

**Elaborado por:** el Investigador

| ID | Fecha               | Nombre         | Correo                | Edad | Teléfono  | Últi |
|----|---------------------|----------------|-----------------------|------|-----------|------|
| 3  | 2022-06-13 01:11:58 | Patricio Silva | bsilva1968@uta.edu.ec | 26   | 960651514 | 2022 |

**Figura 111.** Dashboard “Administradores Registrados” de Administrador

**Elaborado por:** el Investigador

Se desarrolla una sección de páginas en las cuales, se muestra tablas de datos ordenados de forma ascendente o descendente según el usuario lo requiera, en las **Figuras 112 y 113** se observa un par de casos de datos ordenados de forma ascendente y descendente respectivamente.

| ID | Fecha               | Nivel Ajonjoli (%) | Nivel Maní (%) | Nivel Grano 3 (%) | Nivel Grano 4 (%) |
|----|---------------------|--------------------|----------------|-------------------|-------------------|
| 1  | 2022-06-09 04:05:07 | 99.99              | 99.99          | 99.99             | 25.00             |
| 2  | 2022-06-09 04:05:33 | 99.99              | 99.99          | 99.99             | 99.99             |
| 3  | 2022-06-09 04:06:58 | 99.99              | 99.99          | 99.99             | 99.99             |
| 4  | 2022-06-09 04:09:55 | 99.99              | 99.99          | 99.99             | 99.99             |
| 5  | 2022-06-09 04:12:30 | 99.99              | 99.99          | 99.99             | 0.00              |
| 6  | 2022-06-09 04:12:36 | 99.99              | 99.99          | 99.99             | 0.00              |
| 7  | 2022-06-09 04:16:10 | 99.99              | 99.99          | 99.99             | 99.99             |

**Figura 112.** Tabla de Datos Ascendentes por ID

**Elaborado por:** el Investigador

| ID  | Fecha               | Nivel Ajonjoli (%) | Nivel Maní (%) | Nivel Grano 3 (%) | Nivel Grano 4 (%) |
|-----|---------------------|--------------------|----------------|-------------------|-------------------|
| 256 | 2022-06-21 22:03:35 | 99.99              | 99.99          | 99.99             | 50.00             |
| 1   | 2022-06-09 04:05:07 | 99.99              | 99.99          | 99.99             | 25.00             |
| 2   | 2022-06-09 04:05:33 | 99.99              | 99.99          | 99.99             | 99.99             |
| 258 | 2022-06-21 22:03:49 | 99.99              | 99.99          | 99.99             | 99.99             |
| 3   | 2022-06-09 04:06:58 | 99.99              | 99.99          | 99.99             | 99.99             |
| 259 | 2022-06-21 22:04:02 | 99.99              | 99.99          | 99.99             | 50.00             |
| 4   | 2022-06-09 04:09:55 | 99.99              | 99.99          | 99.99             | 99.99             |
| 260 | 2022-06-21 22:04:13 | 99.99              | 99.99          | 99.99             | 50.00             |
| 5   | 2022-06-09 04:12:30 | 99.99              | 99.99          | 99.99             | 0.00              |

**Figura 113.** Tabla de Datos Descendentes por Nivel de Grano 3

**Elaborado por:** el Investigador

La última página web correspondiente a la interfaz web de administradores, presenta los datos de los estados de los contenedores organizados por tablas en orden ascendente y descendente, de esta forma, el usuario puede saber en qué fechas exactamente fue destapado alguno de los cuatro contenedores. En la **Figura 114**, se muestra una captura de pantalla de la página mencionada.

| Estado Abierto Grano 1 |                     |                       | Estado Cerrado Grano 1 |                     |                       |
|------------------------|---------------------|-----------------------|------------------------|---------------------|-----------------------|
| ID                     | Fecha               | Estado Ajonjolí (A/C) | ID                     | Fecha               | Estado Ajonjolí (A/C) |
| 1                      | 2022-06-09 04:05:07 | ABIERTO               | 22                     | 2022-06-09 04:53:45 | Cerrado               |
| 2                      | 2022-06-09 04:05:33 | ABIERTO               | 24                     | 2022-06-09 04:54:00 | Cerrado               |
| 3                      | 2022-06-09 04:06:58 | ABIERTO               | 34                     | 2022-06-09 04:55:20 | CERRADO               |
| 4                      | 2022-06-09 04:09:55 | ABIERTO               | 37                     | 2022-06-14 04:24:35 | Cerrado               |

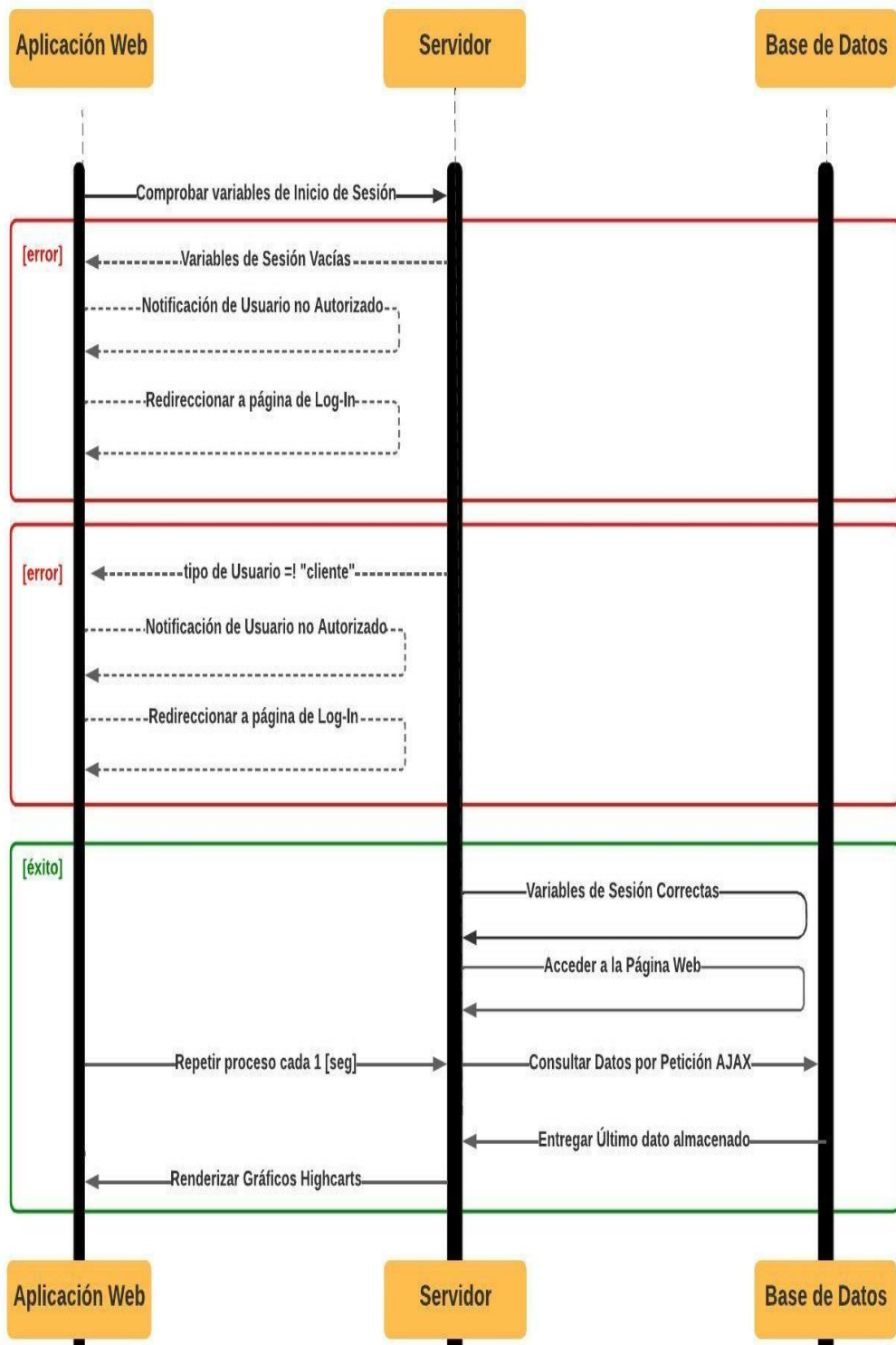
| Estado Abierto Grano 2 |                     |                   | Estado Cerrado Grano 2 |                     |                   |
|------------------------|---------------------|-------------------|------------------------|---------------------|-------------------|
| ID                     | Fecha               | Estado Maní (A/C) | ID                     | Fecha               | Estado Maní (A/C) |
| 1                      | 2022-06-09 04:05:07 | ABIERTO           | 26                     | 2022-06-09 04:54:10 | Cerrado           |
| 2                      | 2022-06-09 04:05:33 | ABIERTO           | 34                     | 2022-06-09 04:55:20 | CERRADO           |
| 3                      | 2022-06-09 04:06:58 | ABIERTO           | 344                    | 2022-06-30 03:15:15 | Cerrado           |

**Figura 114.** Tabla de Datos De estados de los contenedores

**Elaborado por:** el Investigador

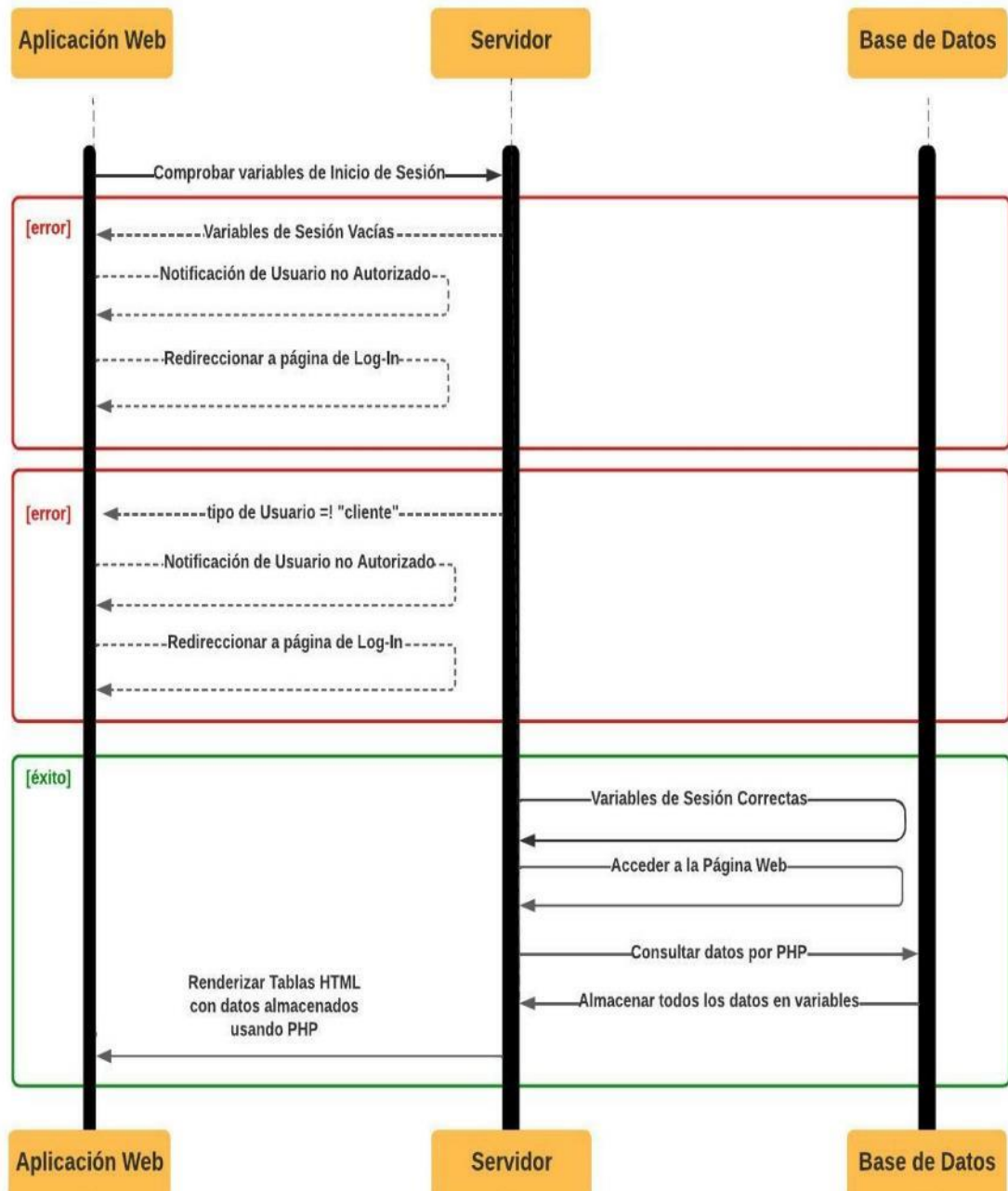
### 3.2.16 Desarrollo de la Interfaz Web de Cliente

La interfaz Web de Cliente consta de múltiples páginas web relacionadas entre sí, estas, a diferencia de las interfaces descritas anteriormente, siguen dos algoritmos distintos de programación web según sea el caso. Para una mejor apreciación de los algoritmos, se presentan figuras con éstos representados en diagramas secuenciales. En la **Figura 115**, se observa un diagrama secuencial para el caso de la única página web que posee gráficos dinámicos que muestran datos en tiempo real; mientras que en la **Figura 116**, se presenta el diagrama secuencial para el caso de las páginas web que muestran datos estáticos en tablas.



**Figura 115.** Diagrama de Secuencia de las Páginas Dinámicas de Clientes

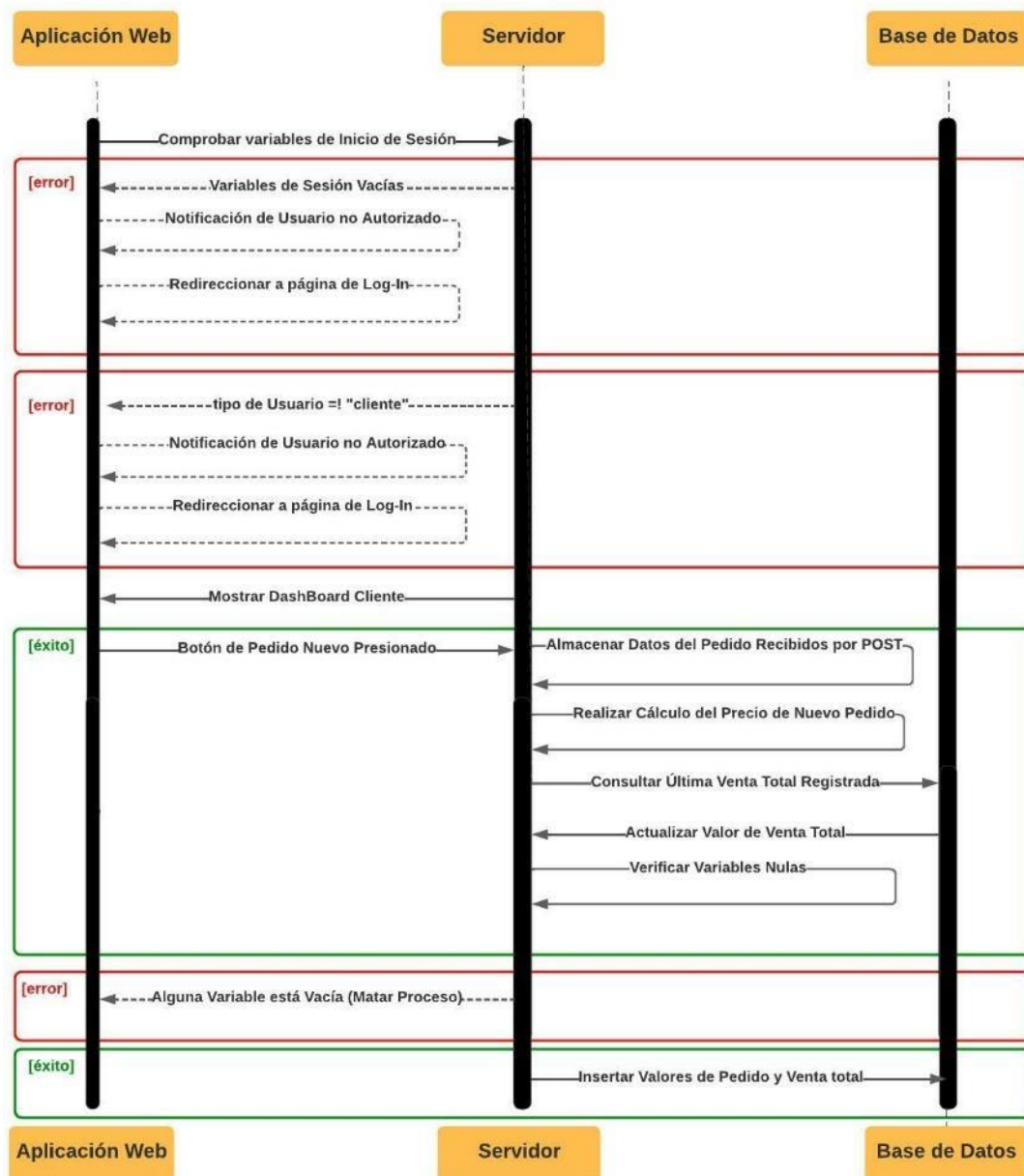
**Elaborado por:** el Investigador



**Figura 116.** Diagrama de Secuencia de las Páginas Estáticas de Administrador

**Elaborado por:** el Investigador

Dentro del conjunto de páginas web que forman la Interfaz Web del cliente, existe una en particular, en la cual, el usuario realiza el pedido personalizado de los frutos secos. En la **Figura 117**, se muestra un diagrama secuencial del algoritmo web programado en la página en cuestión.

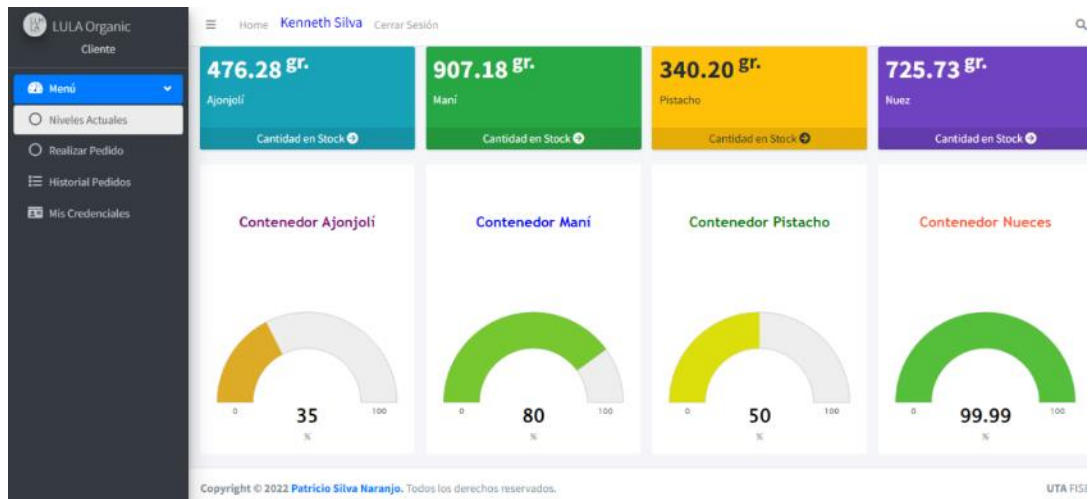


**Figura 117.** Diagrama de Secuencia “Registrar Pedido” Cliente

**Elaborado por:** el Investigador

### **Presentación de Páginas Web para el Usuario Cliente**

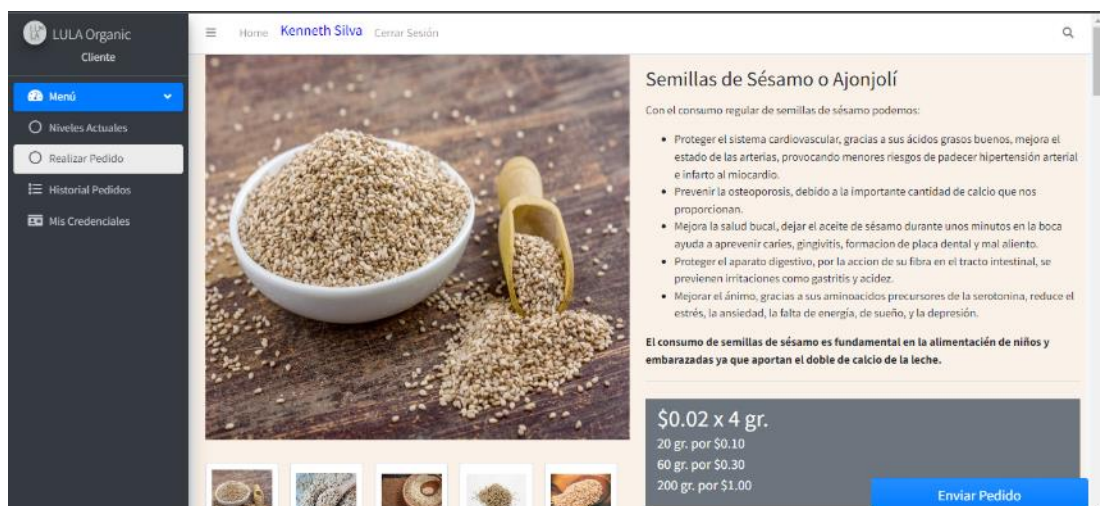
En la **Figura 118**, se presenta el diseño de la página web, a la cuál, el cliente accederá por defecto después de haber iniciado sesión con sus credenciales, se trata de un entorno con gráficos dinámicos que se actualizan en tiempo real sobre la cantidad de producto total en gramos que existe de momento en cada uno de los cuatro contenedores.



**Figura 118.** Dashboard Inicial de Cliente

**Elaborado por:** el Investigador

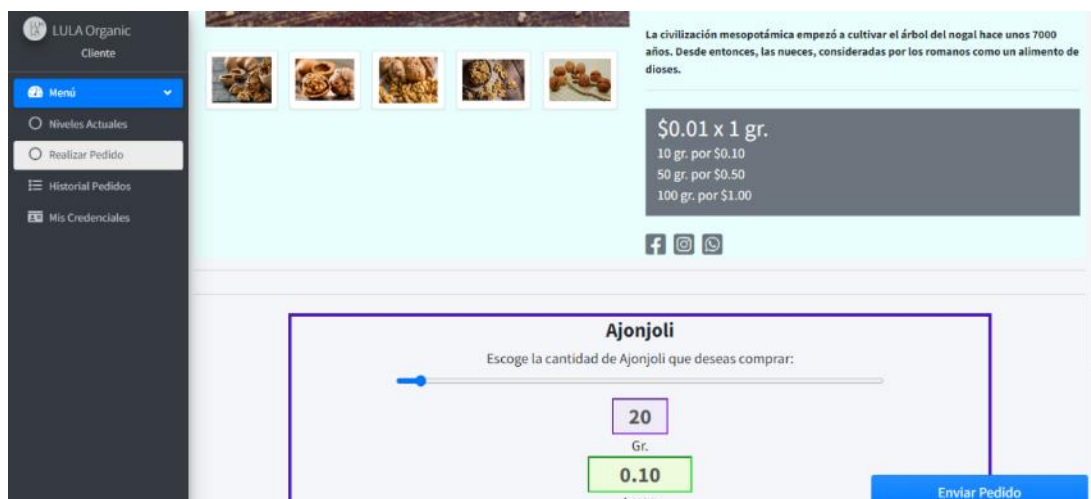
La siguiente página web que forma parte de la Interfaz de cliente, es la más importante, pues esta es el medio de comunicación entre el Usuario y el dispensador de frutos secos. Esta página presenta un panorama comercial y amigable al cliente brindando información destacada del producto ofertado y, además, el usuario escoge la cantidad de su pedido en gramos mediante sliders dinámicos y finalmente son enviados al servidor mediante la acción de presionar un botón. En las **Figuras 119, 120 y 121** se visualiza capturas de la página web mencionada.



**Figura 119.** Página Web “Realizar Pedido” – Vista 1

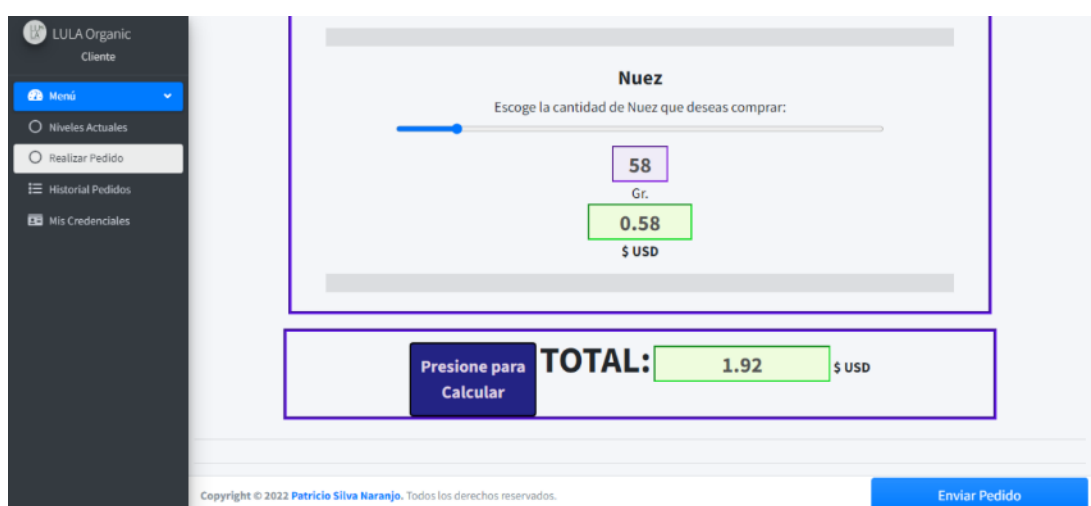
**Elaborado por:** el Investigador





**Figura 120.** Página Web “Realizar Pedido” – Vista 2

**Elaborado por:** el Investigador



**Figura 121.** Página Web “Realizar Pedido” – Vista 3

**Elaborado por:** el Investigador

Las últimas dos páginas web diseñadas en beneficio del cliente presentan tablas de datos de interés para el mismo. Una de ellas sirve para visualizar el historial de pedidos realizados por el usuario, mientras que la otra sirve para visualizar las credenciales con las que el cliente fue registrado y son mostradas en las **Figuras 122 y 123** respectivamente.



| ID | Fecha               | Nombre        | Ajonjolí | Maní | Pistacho | Nuéz |
|----|---------------------|---------------|----------|------|----------|------|
| 2  | 2022-06-21 23:52:47 | Kenneth Silva | 2        | 6    | 0        | 0    |
| 7  | 2022-07-01 02:51:26 | Kenneth Silva | 6        | 14   | 0        | 0    |
| 8  | 2022-07-01 05:19:38 | Kenneth Silva | 9        | 15   | 10       | 0    |
| 9  | 2022-07-01 09:41:57 | Kenneth Silva | 20       | 10   | 10       | 10   |
| 10 | 2022-07-03 22:00:30 | Kenneth Silva | 5        | 6    | 10       | 10   |
| 13 | 2022-07-08 14:43:29 | Kenneth Silva | 20       | 10   | 10       | 10   |

**Figura 122.** Página Web “Historial de Pedidos”

**Elaborado por:** el Investigador

| ID | Fecha Registro      | Nombre        | Correo            | Edad | Telf.     | Último a  |
|----|---------------------|---------------|-------------------|------|-----------|-----------|
| 2  | 2022-06-13 01:11:35 | Kenneth Silva | kenneth@gmail.com | 21   | 992747485 | 2022-07-1 |

**Figura 123.** Página Web “Credenciales de Cliente”

**Elaborado por:** el Investigador

El código fuente de las páginas web tanto de administrador como de cliente, se encuentran detalladas en el **Anexo 7**.

### 3.2.17 Instalación de herramientas y librerías necesarias del prototipo

Gran parte de las dependencias que necesita el proyecto como: JQuery, Bootstrap, Frameworks de íconos, Google Fonts etc. Ya se encuentran incluidos dentro de los archivos de programación web del presente proyecto, sin embargo, existen algunas herramientas adicionales que deben ser instaladas externamente para poder compilar la programación.

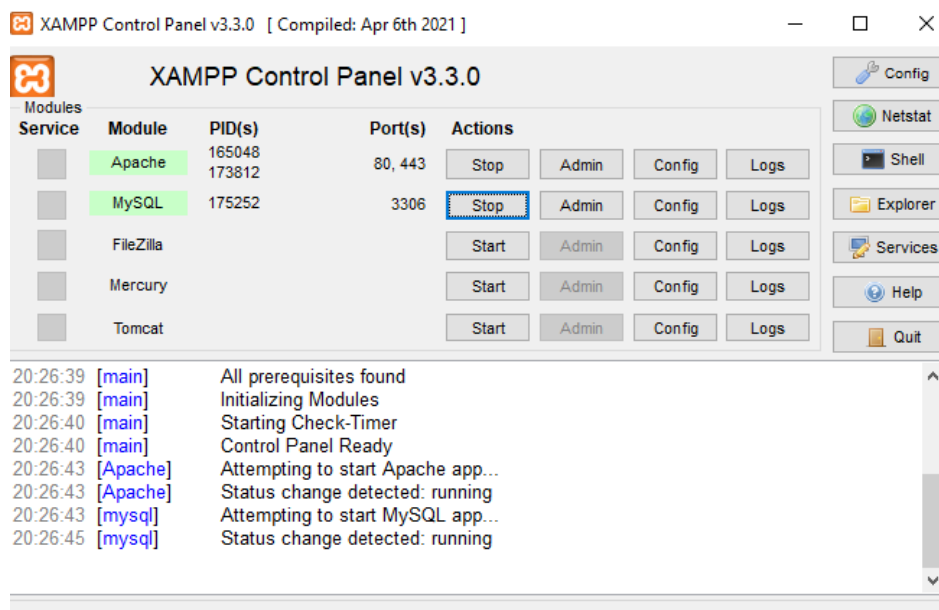
#### Instalación de la Base de Datos

Para poder instalar la base de datos seleccionada, es necesario que el usuario instale en un computador un software libre que consiste en la gestión independiente de varios

servicios Web, de esta manera, dicho computador se comportará como un servidor local y permitirá probar todas las interfaces web programadas en el presente proyecto. Para el sistema operativo Windows, el software más simple de acuerdo con la experiencia del investigador es XAMP, el mismo que permite activar servicios como Apache, MySQL, FileZilla, Mercury y Tomcat. Se lo puede descargar de su sitio web oficial: <https://www.apachefriends.org/es/index.html> y el proceso de instalación es muy simple:

- ✓ Descargar el archivo .exe.
- ✓ Ejecutar el archivo.
- ✓ Instalarlo usando el asistente de Windows de acuerdo con las necesidades del usuario.
- ✓ Inicializar los módulos de activación de servicios de Apache y MySQL.

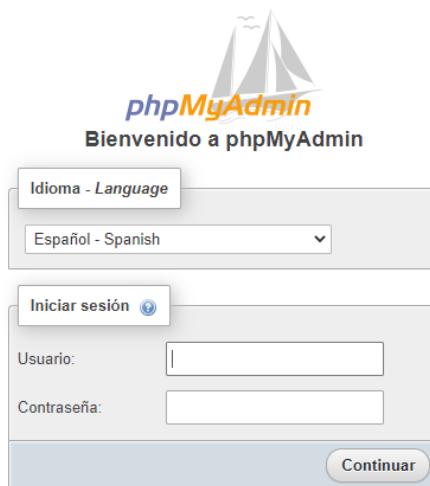
En la **Figura 124**, se muestra el panel de control de XAMP junto con los módulos necesarios activados.



**Figura 124.** Panel de Control de XAMP activado

**Elaborado por:** el Investigador

Para comprobar el correcto funcionamiento e instalación de la base de datos, se procede a abrir el navegador web e ingresar la dirección IP local: <http://localhost/phpmyadmin> e ingresar a PhpMyAdmin para la gestión y creación de las bases de datos. En la **Figura 125**, se muestra la interfaz de ingreso a PhpMyAdmin.



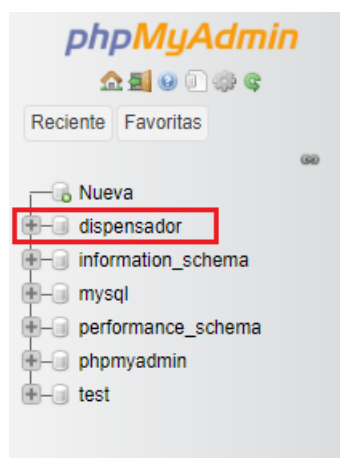
**Figura 125.** Ingreso a PhpMyAdmin

**Elaborado por:** el Investigador

De existir el caso en que el usuario desee instalar la base de datos usando otro tipo de software online o local, se sugiere escoger los modelos seleccionados en la **Sección 3.2.5.3** (Selección modelo de servicio en la Nube).

### **Creación de las Tablas de Datos**

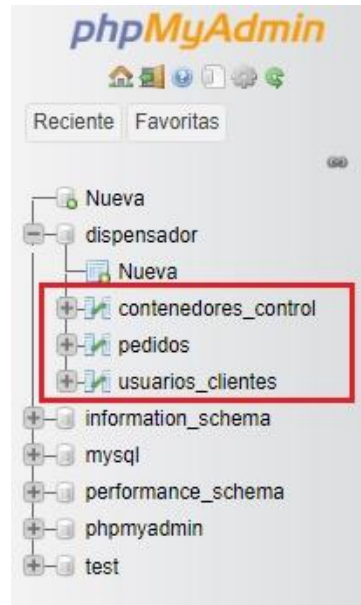
El usuario por defecto para ingresar al entorno PhpMyAdmin d XAMP es “root” sin contraseña, sin embargo, se puede crear usuarios con su respectiva contraseña según sea necesario desde el mismo gestor PhpMyAdmin. Una vez ingresado, se debe crear una base de datos nueva denominada “dispensador”, como se muestra en la **Figura 126**.



**Figura 126.** Base de Datos creada

**Elaborado por:** el Investigador

La Base de datos denominada “dispensador”, necesita contener a tres tablas de datos para que el presente proyecto pueda funcionar correctamente, las mismas que pueden ser observadas en la **Figura 127**.



**Figura 127.** Tablas de Datos

**Elaborado por:** el Investigador

La tabla denominada “**contenedores\_control**”, se encargará de almacenar las variables medidas por los sensores que son enviadas por el microcontrolador y consta de diez campos que son:

- Id
- Fecha
- Nivel del contenedor 1
- Nivel del contenedor 2
- Nivel del contenedor 3
- Nivel del contenedor 4
- Estado del Contenedor 1
- Estado del Contenedor 2
- Estado del Contenedor 3
- Estado del Contenedor 4

La estructura de la tabla se muestra en las **Figuras 128 y 129**.

| Estructura de tabla      |                  | Vista de relaciones |                    |           |      |                     |             |                |  |  |
|--------------------------|------------------|---------------------|--------------------|-----------|------|---------------------|-------------|----------------|--|--|
| #                        | Nombre           | Tipo                | Cotejamiento       | Atributos | Nulo | Predeterminado      | Comentarios | Extra          |  |  |
| <input type="checkbox"/> | 1 id             | int(11)             |                    |           | No   | Ninguna             |             | AUTO_INCREMENT |  |  |
| <input type="checkbox"/> | 2 fecha          | timestamp           |                    |           | No   | current_timestamp() |             |                |  |  |
| <input type="checkbox"/> | 3 nivel_cont_1   | double(4,2)         |                    |           | No   | Ninguna             |             |                |  |  |
| <input type="checkbox"/> | 4 nivel_cont_2   | double(4,2)         |                    |           | No   | Ninguna             |             |                |  |  |
| <input type="checkbox"/> | 5 nivel_cont_3   | double(4,2)         |                    |           | No   | Ninguna             |             |                |  |  |
| <input type="checkbox"/> | 6 nivel_cont_4   | double(4,2)         |                    |           | No   | Ninguna             |             |                |  |  |
| <input type="checkbox"/> | 7 estado_cont_1  | text                | utf8mb4_general_ci |           | No   | Ninguna             |             |                |  |  |
| <input type="checkbox"/> | 8 estado_cont_2  | text                | utf8mb4_general_ci |           | No   | Ninguna             |             |                |  |  |
| <input type="checkbox"/> | 9 estado_cont_3  | text                | utf8mb4_general_ci |           | No   | Ninguna             |             |                |  |  |
| <input type="checkbox"/> | 10 estado_cont_4 | text                | utf8mb4_general_ci |           | No   | Ninguna             |             |                |  |  |

**Figura 128.** Estructura de la Tabla “contenedores\_control”

**Elaborado por:** el Investigador

| id   | fecha                  | nivel_cont_1 | nivel_cont_2 | nivel_cont_3 | nivel_cont_4 | estado_cont_1 | estado_cont_2 | estado_cont_3 | estado_cont_4 |
|------|------------------------|--------------|--------------|--------------|--------------|---------------|---------------|---------------|---------------|
| 1201 | 2022-07-06<br>20:48:02 | 60.00        | 80.00        | 50.00        | 99.99        | Cerrado       | ABIERTO       | Cerrado       | Cerrado       |
| 1202 | 2022-07-06<br>20:48:08 | 60.00        | 80.00        | 50.00        | 99.99        | Cerrado       | Cerrado       | Cerrado       | Cerrado       |
| 1203 | 2022-07-06<br>20:49:10 | 60.00        | 99.99        | 50.00        | 99.99        | Cerrado       | Cerrado       | Cerrado       | Cerrado       |
| 1204 | 2022-07-06<br>20:49:16 | 60.00        | 99.99        | 50.00        | 99.99        | ABIERTO       | Cerrado       | Cerrado       | Cerrado       |

**Figura 129.** Datos de la Tabla “contenedores\_control”

**Elaborado por:** el Investigador

La siguiente tabla de datos denominada “**pedidos**”, se encargará de almacenar los pedidos realizados por todos los clientes junto al valor de venta total acumulada, esta tabla cuenta con once campos que se muestran en las **Figuras 130 y 131**. Cabe recalcar que, los datos booleanos almacenados en el campo “retirado” sirven para que el microcontrolador sepa que algún pedido está pendiente o ya fue dispensado.

|                          |        | Estructura de tabla |              | Vista de relaciones |      |                     |             |       |  |                |
|--------------------------|--------|---------------------|--------------|---------------------|------|---------------------|-------------|-------|--|----------------|
| #                        | Nombre | Tipo                | Cotejamiento | Atributos           | Nulo | Predeterminado      | Comentarios | Extra |  |                |
| <input type="checkbox"/> | 1      | id                  | int(11)      |                     | No   | Ninguna             |             |       |  | AUTO_INCREMENT |
| <input type="checkbox"/> | 2      | fecha               | timestamp    |                     | No   | current_timestamp() |             |       |  |                |
| <input type="checkbox"/> | 3      | correo              | varchar(60)  | utf8mb4_general_ci  | No   | Ninguna             |             |       |  |                |
| <input type="checkbox"/> | 4      | nombre              | text         | latin1_general_ci   | No   | Ninguna             |             |       |  |                |
| <input type="checkbox"/> | 5      | grano1              | int(4)       |                     | No   | Ninguna             |             |       |  |                |
| <input type="checkbox"/> | 6      | grano2              | int(4)       |                     | No   | Ninguna             |             |       |  |                |
| <input type="checkbox"/> | 7      | grano3              | int(4)       |                     | No   | Ninguna             |             |       |  |                |
| <input type="checkbox"/> | 8      | grano4              | int(4)       |                     | No   | Ninguna             |             |       |  |                |
| <input type="checkbox"/> | 9      | retirado            | tinyint(1)   |                     | No   | 0                   |             |       |  |                |
| <input type="checkbox"/> | 10     | precio              | double       |                     | No   | Ninguna             |             |       |  |                |
| <input type="checkbox"/> | 11     | venta_total         | double       |                     | No   | Ninguna             |             |       |  |                |

**Figura 130.** Estructura de la Tabla “pedidos”

**Elaborado por:** el Investigador

| id | fecha               | correo                | nombre        | grano1 | grano2 | grano3 | grano4 | retirado | precio | venta_total |
|----|---------------------|-----------------------|---------------|--------|--------|--------|--------|----------|--------|-------------|
| 1  | 2022-07-14 00:20:38 | kenneth@gmail.com     | Kenneth Silva | 27     | 19     | 33     | 33     | 1        | 1.65   | 1.65        |
| 2  | 2022-07-14 00:20:54 | kenneth@gmail.com     | Kenneth Silva | 4      | 49     | 40     | 32     | 1        | 2.03   | 3.68        |
| 3  | 2022-07-14 00:21:34 | merazo9003@uta.edu.ec | Marcos Erazo  | 0      | 17     | 17     | 59     | 1        | 1.27   | 4.95        |
| 4  | 2022-07-14 00:21:45 | merazo9003@uta.edu.ec | Marcos Erazo  | 20     | 38     | 87     | 50     | 1        | 3.59   | 8.54        |

**Figura 131.** Datos de la Tabla “pedidos”

**Elaborado por:** el Investigador

La tercera y última tabla de datos, denominada “**usuarios\_clientes**”, se encarga de almacenar los registros de todos los usuarios, tanto administradores como clientes y consta de nueve campos que se muestran en las **Figuras 132 y 133**. Todos los registros nuevos están programados por defecto para ser de tipo “cliente”, sin embargo, para registrar un nuevo administrador basta con cambiar desde PhpMyAdmin el campo “tipo” de “cliente” a “admin”.

| Estructura de tabla      |   | Vista de relaciones |             | #                  | Nombre | Tipo | Cotejamiento | Atributos | Nulo | Predeterminado      | Comentarios | Extra          |
|--------------------------|---|---------------------|-------------|--------------------|--------|------|--------------|-----------|------|---------------------|-------------|----------------|
| <input type="checkbox"/> | 1 | id                  | int(11)     |                    |        |      |              |           | No   | Ninguna             |             | AUTO_INCREMENT |
| <input type="checkbox"/> | 2 | fecha               | timestamp   |                    |        |      |              |           | No   | current_timestamp() |             |                |
| <input type="checkbox"/> | 3 | nombre              | varchar(60) | latin1_general_ci  |        |      |              |           | No   | Ninguna             |             |                |
| <input type="checkbox"/> | 4 | correo              | varchar(60) | latin1_general_ci  |        |      |              |           | No   | Ninguna             |             |                |
| <input type="checkbox"/> | 5 | password            | varchar(60) | latin1_general_ci  |        |      |              |           | No   | Ninguna             |             |                |
| <input type="checkbox"/> | 6 | edad                | int(3)      |                    |        |      |              |           | No   | Ninguna             |             |                |
| <input type="checkbox"/> | 7 | telefono            | int(11)     |                    |        |      |              |           | No   | Ninguna             |             |                |
| <input type="checkbox"/> | 8 | ultimo acceso       | timestamp   |                    |        |      |              |           | No   | current_timestamp() |             |                |
| <input type="checkbox"/> | 9 | tipo                | text        | utf8mb4_general_ci |        |      |              |           | No   | 'cliente'           |             |                |

**Figura 132.** Estructura de la Tabla “usuarios\_clientes”

**Elaborado por:** el Investigador

| id | fecha               | nombre         | correo                      | password                                 | edad | telefono  | ultimo acceso       | tipo    |
|----|---------------------|----------------|-----------------------------|--|------|-----------|---------------------|---------|
| 1  | 2022-06-13 01:11:14 | Marcos Erazo   | merazo9003@uta.edu.ec       | 7c222fb2927d828af22f592134e8932480637c0d | 25   | 998141177 | 2022-07-14 00:21:18 | cliente |
| 2  | 2022-06-13 01:11:35 | Kenneth Silva  | kenneth@gmail.com           | 7c222fb2927d828af22f592134e8932480637c0d | 21   | 992747485 | 2022-07-14 00:25:17 | admin   |
| 3  | 2022-06-13 01:11:58 | Patricio Silva | bsilva1968@uta.edu.ec       | 7c222fb2927d828af22f592134e8932480637c0d | 26   | 960651514 | 2022-07-13 23:46:58 | admin   |
| 4  | 2022-07-03 22:19:04 | Soraya Naranjo | naranjosoraya@gmail.com     | 7372c359f733b6dfe68b1700e91dd1d7173bc71a | 48   | 998348029 | 2022-07-03 22:19:50 | cliente |
| 5  | 2022-07-03 22:30:41 | Danna Silva    | silva.dannalorena@gmail.com | be86557a8a4b7491d5068ae8c57fc1454f57673f | 18   | 998348029 | 2022-07-03 22:31:05 | cliente |

**Figura 133.** Datos de la Tabla “usuarios\_clientes”

**Elaborado por:** el Investigador

### Instalación de las librerías en Arduino IDE

Debido a que la programación de los microcontroladores Arduino Mega y NodeMCU se las desarrolla en el software libre Arduino IDE, se necesita en algunos casos la instalación manual de librerías que podrían no estar instaladas por defecto dependiendo la versión del software que se disponga, de ser este el caso, todas las librerías que se necesita tener instaladas para compilar los programas del presente proyecto son:

```

// Librerías pantalla LCD I2C =====
#include <Wire.h>      // librería de comunicación por I2C
#include <LiquidCrystal_I2C.h>  // librería para LCD por I2C

//=====

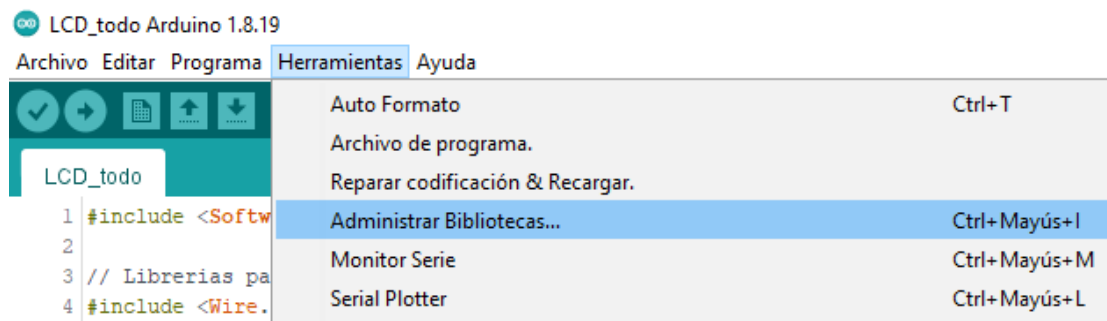
// Librerías para celda de carga =====
#include <HX711_ADC.h>
#include <EEPROM.h>
//=====

// Librería para los servomotores=====
#include <Servo.h>
//=====

// Librerías para la NODEMCU
=====
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
//=====

```

El proceso de instalación para cada una de las librerías es exactamente el mismo, dentro del Software Arduino IDE hay que dirigirse a la pestaña Herramientas – Administrar Bibliotecas, proceso que es ilustrado en la **Figura 134**.

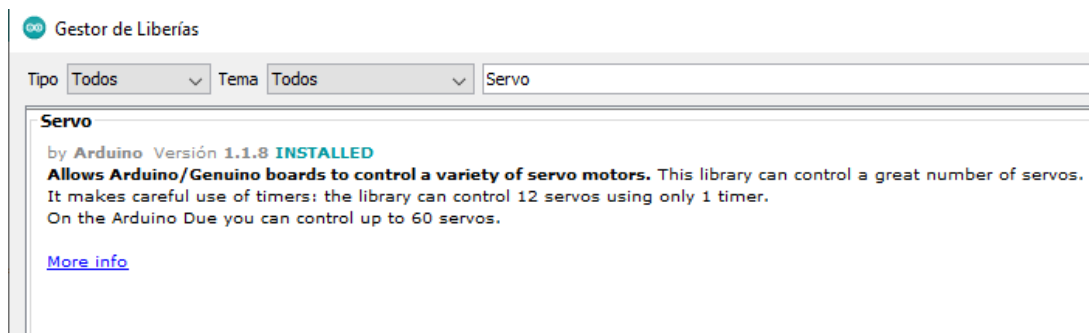


**Figura 134.** Gestor de Librerías

**Elaborado por:** el Investigador

Se abre el panel de Gestor de Librerías y dentro de este se escribe el nombre de la librería en cuestión, finalmente se presiona en instalar. El proceso se repite para cada una de las librerías que se requiera incluir. En la **Figura 135** se observa un ejemplo, en el que se instala la librería “servo”.



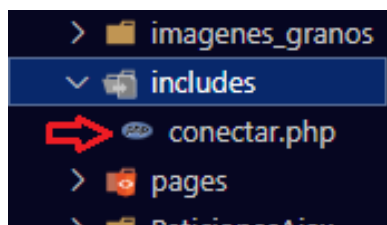


**Figura 135.** Instalación de Librería en Arduino

**Elaborado por:** el Investigador

### Conexión de la Base de Datos con el Servidor

Dentro de los archivos de la programación web del proyecto existe una carpeta denominada “**includes**”, en la cual, está contenido un único archivo con el nombre “**conectar.php**”, este archivo es el responsable de la conexión a la base de datos que requieren todas las páginas web del presente proyecto junto con los archivos que se ejecuta desde el microcontrolador, por lo tanto, se debe editar este archivo con las credenciales de la Base de Datos creada. En la **Figura 136** se muestra la carpeta y posteriormente el código fuente de la conectividad con la Base de datos.



**Figura 136.** Carpeta Includes

**Elaborado por:** el Investigador

```
<?php
$host="localhost";
$usuario="root";
$contraseña="";
$base="dispensador";

$conexion= new mysqli($host, $usuario, $contraseña, $base);
mysqli_query($conexion,"SET NAMES 'utf8'");
if ($conexion -> connect_errno)
{
```

```

        die("Fallo la conexion:( ".$conexion ->
mysqli_connect_errno().") ".$conexion->
        mysqli_connect_error());
    }
    // conexión de Peticiones Ajax
    $pdo=new PDO("mysql:dbname=dispensador;host=localhost","root","");
?>

```

### Conexión del microcontrolador con el servidor

Dentro de la programación del microcontrolador NodeMCU, existen variables de tipo String que contienen una dirección IP en común, esta se trata de la dirección en donde está instalado el servidor, por lo tanto, independientemente si el servidor es local o alojado en la nube, se debe verificar y cambiar la IP para que el prototipo pueda funcionar. Este proceso se lo observa en la **Figura 137**.

```

11 const char *URL = "http://192.168.1.10/R_W/envio_datos.php";
12
13 const char *URL_1 = "http://192.168.1.10/R_W/lectura_pedido_id.php";
14 const char *URL_2 = "http://192.168.1.10/R_W/lectura_pedido_correo.php";
15 const char *URL_3 = "http://192.168.1.10/R_W/lectura_pedido_nombre.php";
16 const char *URL_4 = "http://192.168.1.10/R_W/lectura_pedido_grano1.php";
17 const char *URL_5 = "http://192.168.1.10/R_W/lectura_pedido_grano2.php";
18 const char *URL_6 = "http://192.168.1.10/R_W/lectura_pedido_grano3.php";
19 const char *URL_7 = "http://192.168.1.10/R_W/lectura_pedido_grano4.php";
20 const char *URL_8 = "http://192.168.1.10/R_W/lectura_pedido_retirado.php";
21 const char *URL_9 = "http://192.168.1.10/R_W/lectura_pedido_precio.php";
22 const char *URL_10 = "http://192.168.1.10/R_W/lectura_pedido_venta_total.php";
23 const char *URL_11 = "http://192.168.1.10/R_W/actualizar_pedido_retirado.php";
24

```

**Figura 137.** Conexión NodeMCU con Servidor

**Elaborado por:** el Investigador

### 3.2.18 Seguridad y Encriptación de Datos

Dentro de la programación de Back – End responsable del registro e inicio de sesión de usuarios, existen funciones de PHP que reducen el riesgo de sufrir ataques ante vulnerabilidades que pueden ser aprovechadas desde las cajas de texto de ingreso, una de estas funciones es **strip\_tags()**, la cual, retira cualquier etiqueta HTML, PHP o SQL de la variable involucrada, por lo tanto, se usa esta función en todas las variables recibidas en el servidor por medio del método POST, su sintaxis es la siguiente:

```
$name = strip_tags($_POST ['name']);
$email= strip_tags($_POST ['email']);
$edad= strip_tags($_POST ['edad']);
```

Otra función PHP usada en el presente proyecto es **sha1()**, el cuál calcula el hash de un String, es decir, esta función realiza un algoritmo matemático que transforma cualquier bloque de datos en una nueva cadena de caracteres de longitud fija, y se la emplea en el presente proyecto para encriptar las contraseñas de los usuarios, de esta manera, lo que será almacenado en la Base de Datos es el hash, mas no, la contraseña en sí; su sintaxis es la siguiente:

```
$pwd= sha1(strip_tags($_POST ['pwd'])) ;
$r_pwd= sha1(strip_tags($_POST ['r_pwd']));
```

En la **Figura 138** se observan algunos hashes almacenados en la Base de Datos en el campo de **password**.

| id | fecha                  | nombre            | correo                      | password                                 | edad | telefono  | u     |
|----|------------------------|-------------------|-----------------------------|--|------|-----------|-------|
| 4  | 2022-07-03<br>22:19:04 | Soraya<br>Naranjo | naranjosoraya@gmail.com     | 7372c359f733b6dfe68b1700e91dd1d7173bc71a | 48   | 998348029 | 2022: |
| 5  | 2022-07-03<br>22:30:41 | Danna Silva       | silva.dannalorena@gmail.com | be86557a8a4b7491d5068ae8c57fc1454f57673f | 18   | 998348029 | 2022: |

**Figura 138.** Hashes Almacenados

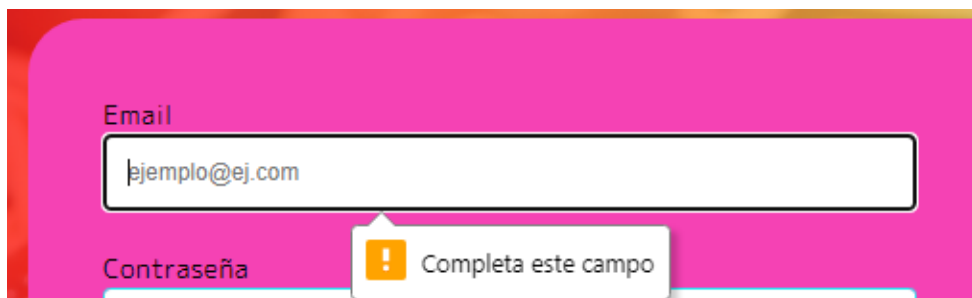
**Elaborado por:** el Investigador

Adicionalmente, todos los elementos HTML de tipo “input” empleados en el presente proyecto están validados mediante el atributo “**required**”, esto quiere decir, que las páginas HTML indicarán un mensaje de alerta al usuario en caso de que un campo no esté llenado correctamente, impidiendo la ejecución de procesos siguientes; la sintaxis se la muestra en la **Figura 139** mientras que algunos ejemplos del caso mencionado se lo observa en las **Figuras 140 y 141**.

```
no htmlspecialchars($_SERVER['PHP_SELF'])?>" method="post">
ings">
="name-label"> Nombre y Apellido: </label>
name="name" placeholder="Ejm. Patricio Silva" class="form-inputs" required> </input>
```

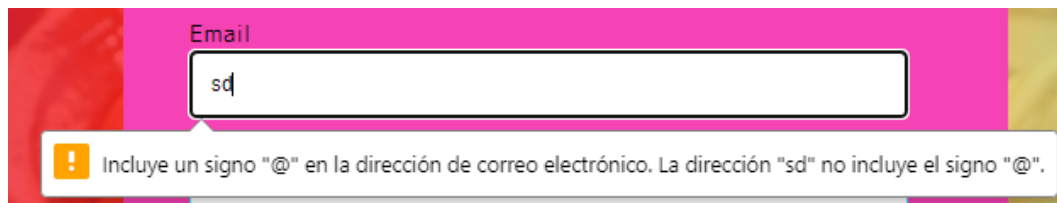
**Figura 139.** Atributo Required

**Elaborado por:** el Investigador



**Figura 140.** Caso de Campos vacíos

**Elaborado por:** el Investigador



**Figura 141.** Caso de Campos con Datos erróneos

**Elaborado por:** el Investigador

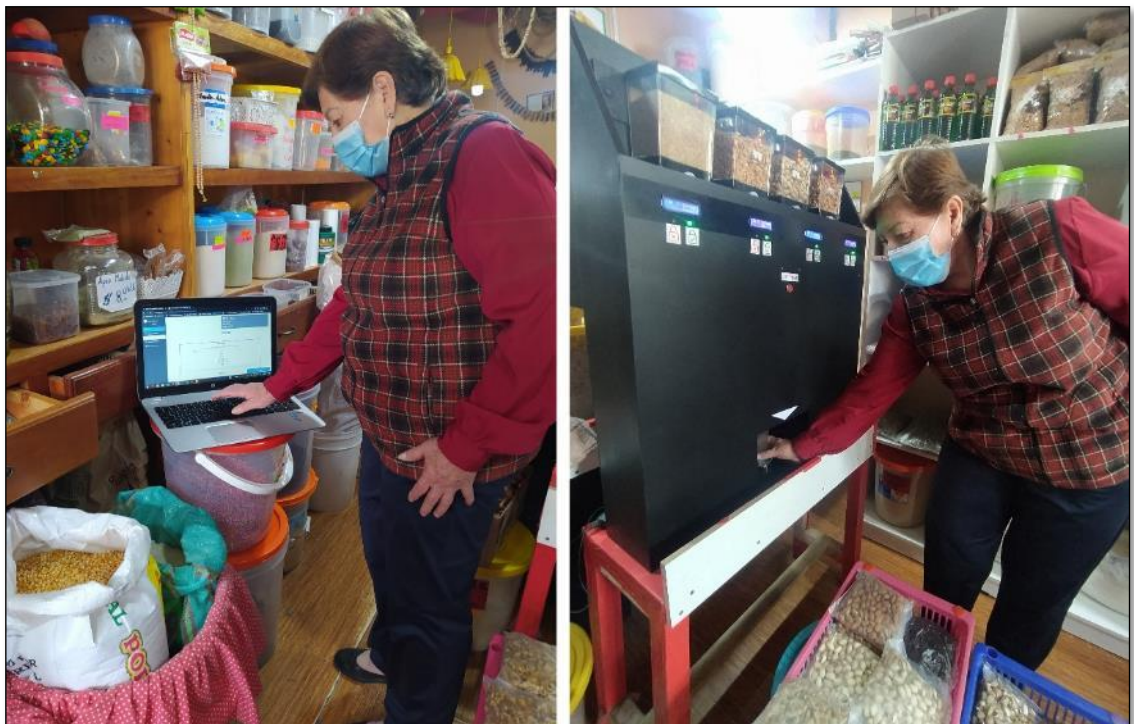
### 3.3 Pruebas de Funcionamiento y Resultados

Para la verificación del funcionamiento del prototipo, se ejecutaron pruebas con clientes de la empresa LULA ORGANIC que realizaron pedidos de distintas cantidades de frutos secos. En las **Figuras 142 hasta la 145** se presenta la implementación del prototipo con los usuarios mencionados.



**Figura 142.** Pruebas con Clientes de la Empresa

**Elaborado por:** el Investigador



**Figura 143.** Pruebas con Clientes de la Empresa

**Elaborado por:** el Investigador





**Figura 144.** Pruebas con Clientes de la Empresa

**Elaborado por:** el Investigador



**Figura 145.** Pruebas con Clientes de la Empresa

**Elaborado por:** el Investigador

El análisis de los datos obtenidos de los pedidos de prueba se detalla en la **Tabla 10**.

**Tabla 10.** Datos de los clientes de prueba

|                          |                 | <b>Ciente 1</b> | <b>Ciente 2</b> | <b>Ciente 3</b> | <b>Ciente 4</b> |
|--------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| <b>Pedido</b>            | <b>Ajonjolí</b> | 10 gr.          | 13 gr.          | 15 gr.          | 20 gr.          |
|                          | <b>Maní</b>     | 18 gr.          | 14 gr.          | 7 gr.           | 10 gr.          |
|                          | <b>Pistacho</b> | 20 gr.          | 30 gr.          | 15 gr.          | 10 gr.          |
|                          | <b>Nuez</b>     | 17 gr.          | 12 gr.          | 23 gr.          | 39 gr.          |
|                          |                 |                 |                 |                 |                 |
| <b>Dispensado</b>        | <b>Ajonjolí</b> | 10.5 gr.        | 13.6 gr.        | 15.3 gr.        | 20.4 gr.        |
|                          | <b>Maní</b>     | 19.3 gr.        | 15 gr.          | 7.6 gr.         | 10.6 gr.        |
|                          | <b>Pistacho</b> | 21.3 gr.        | 30.7 gr.        | 15.5 gr.        | 10.2 gr.        |
|                          | <b>Nuez</b>     | 18.5 gr.        | 12 gr.          | 24.5 gr.        | 41 gr.          |
| <b>Precio Total (\$)</b> |                 | <b>\$ 1.72</b>  | <b>\$ 1.91</b>  | <b>\$ 1.50</b>  | <b>\$ 1.87</b>  |

**Elaborado por:** El Investigador

En la tabla anterior se puede visualizar los valores exactos que fueron solicitados por los clientes junto con los valores reales dispensados, los mismos que varían según el tipo de grano dispensado:

- **Ajonjolí**, varía entre 0.3 a 0.6 gramos.
- **Maní**, varía entre 0.6 a 1.6 gramos.
- **Pistacho**, varía entre 0.2 a 1.3 gramos.
- **Nuez**, varía entre 0 a 2 gramos.

Valores que se encuentran dentro de un rango de variación totalmente aceptable dentro de empresas dedicadas a la venta de productos a granel de forma no automatizada, esto se debe a que, el peso de un grano individual generalmente varía con respecto a otro grano del mismo tipo, en especial en los granos de tamaño grueso como pistacho y nuez, sin embargo, el prototipo está diseñado para dispensar el valor más aproximado posible al real. Es importante destacar que, este tipo de empresas ofertan sus productos en pesos estandarizados en libras y/o kilogramos, por lo que, una variación mínima en

gramos es considerada despreciable, y no perjudica en lo absoluto a la Empresa Lula Organic, además, es una variación que es compensada con el porcentaje elevado al precio convencional de los productos, precio que el cliente, está dispuesto a pagar a cambio de disfrutar de un servicio que le brinde una mejor experiencia de personalización de su compra, según las características de la Industria 4.0 analizadas en el presente documento. En la **Tabla 11** se presenta el cálculo del porcentaje de error de las pruebas ejecutadas.

**Tabla 11.** Error Porcentual

| <b>Muestra</b>        | <b>Cantidad Solicitada</b> | <b>Cantidad Dispensada</b> | <b>Variación</b> |          |
|-----------------------|----------------------------|----------------------------|------------------|----------|
| <b>1</b>              | 10 gr.                     | 10.5 gr.                   | + 0.5 gr.        | Ajonjolí |
| <b>2</b>              | 13 gr.                     | 13.6 gr.                   | + 0.6 gr.        |          |
| <b>3</b>              | 15 gr.                     | 15.3 gr.                   | + 0.3 gr.        |          |
| <b>4</b>              | 20 gr.                     | 20.4 gr.                   | + 0.4 gr.        |          |
| <b>5</b>              | 18 gr.                     | 19.3 gr.                   | + 1.3 gr.        | Maní     |
| <b>6</b>              | 14 gr.                     | 15 gr.                     | + 1.0 gr.        |          |
| <b>7</b>              | 7 gr.                      | 7.6 gr.                    | + 0.6 gr.        |          |
| <b>8</b>              | 10 gr.                     | 10.6 gr.                   | + 0.6 gr.        |          |
| <b>9</b>              | 20 gr.                     | 21.3 gr.                   | + 1.3 gr.        | Pistacho |
| <b>10</b>             | 30 gr.                     | 30.7 gr.                   | + 0.7 gr.        |          |
| <b>11</b>             | 15 gr.                     | 15.5 gr.                   | + 0.5 gr.        |          |
| <b>12</b>             | 10 gr.                     | 10.2 gr.                   | + 0.2 gr.        |          |
| <b>13</b>             | 17 gr.                     | 18.5 gr.                   | + 1.5 gr.        | Nuez     |
| <b>14</b>             | 12 gr.                     | 12 gr.                     | + 0 gr.          |          |
| <b>15</b>             | 23 gr.                     | 24.5 gr.                   | + 1.5 gr.        |          |
| <b>16</b>             | 39 gr.                     | 41 gr.                     | + 2 gr.          |          |
| <b>Error Promedio</b> |                            |                            | + 0.8125 gr.     | 4.81 %   |

**Elaborado por:** El Investigador

### 3.4 Presupuesto

En el análisis de los costos necesarios para la implementación del prototipo se toma como referencia un sueldo promedio que percibe un Ingeniero en Telecomunicaciones



en Ecuador según la Remuneración Mensual por Puesto detallada en el Art. 7 de la Ley Orgánica de Transparencia y Acceso a la Información Pública, misma que indica que el mencionado valor es de \$ 700,00 [52].

A partir de este valor, se calcula la remuneración diaria percibida, considerando 21 días laborales por cada mes, por lo tanto:

$$Remuneración_{diaria} = \frac{Remuneración_{mensual}}{Días_{laborables}} = \frac{\$ 700}{21}$$

$$Remuneración_{diaria} = \$ 33.33$$

Se toma en consideración que el día laboral consta de 8 horas, por lo tanto, la remuneración por hora es:

$$Remuneración_{hora} = \frac{Remuneración_{diaria}}{Horas_{laborales}} = \frac{\$ 33.33}{8}$$

$$Remuneración_{hora} = \$ 4.17$$

Una vez diseñado e implementado el prototipo de manera exitosa, se plantea un tiempo de construcción del dispensador completo en 120 horas laborales, por lo que el presupuesto del diseño es:

$$Presupuesto_{diseño} = Remuneración_{hora} * horas_{laborales} = 4.17 * 120$$

$$Presupuesto_{diseño} = \$ 500.40$$

**Tabla 12.** Presupuesto de Diseño

| <b>Presupuesto de Diseño</b> |           |
|------------------------------|-----------|
| <b>Total</b>                 | \$ 500.40 |

**Elaborado por:** El Investigador

En las **Tablas 13** y **14** se detalla el costo de los materiales utilizados para la construcción del prototipo.

**Tabla 13.** Presupuesto de Construcción 1

| <b>Presupuesto de Construcción</b> |                                     |               |                 |                       |                    |
|------------------------------------|-------------------------------------|---------------|-----------------|-----------------------|--------------------|
| <b>Ítem</b>                        | <b>Descripción</b>                  | <b>Unidad</b> | <b>Cantidad</b> | <b>Costo Unitario</b> | <b>Costo Total</b> |
| 1                                  | Dispensador de plástico para granos | c/u           | 4               | \$ 6.18               | \$ 24.72           |
| 2                                  | Servo MG-995 de 360°                | c/u           | 8               | \$ 13.00              | \$ 104.00          |
| 3                                  | LCD 16x2 con I2C                    | c/u           | 4               | \$ 7.00               | \$ 28.00           |
| 4                                  | Leds de alto brillo                 | c/u           | 11              | \$ 0.20               | \$ 2.20            |
| 5                                  | Arduino Mega 2560                   | c/u           | 1               | \$ 22.00              | \$ 22.00           |
| 6                                  | NodeMCU Esp8266                     | c/u           | 1               | \$ 10.00              | \$ 10.00           |
| 7                                  | Impresión 3D                        | c/u           | 5               | \$ 15.00              | \$ 75.00           |
| 8                                  | Pulsador                            | c/u           | 1               | \$ 1.50               | \$ 1.50            |
| 9                                  | Sensor Efecto Hall M44              | c/u           | 4               | \$ 2.50               | \$ 10.00           |
| 10                                 | HC-SR04                             | c/u           | 4               | \$ 2.50               | \$ 10.00           |
| 11                                 | Resistencias varias                 | c/u           | 13              | \$ 0.10               | \$ 1.30            |
| 12                                 | Fuente DC 5V 2A                     | c/u           | 1               | \$ 5.00               | \$ 5.00            |
| 13                                 | Fuente DC 6V                        | c/u           | 1               | \$ 6.40               | \$ 6.40            |
| 14                                 | Cables Arduino Macho - Macho        | Kit           | 5               | \$3.20                | \$ 16.00           |
| 15                                 | Cables Arduino Macho-Hembra         | Kit           | 5               | \$3.20                | \$ 16.00           |
| 16                                 | Espadines Hembra                    | Kit de 40     | 4               | \$0.60                | \$ 2.40            |
| 17                                 | Espadines Macho                     | Kit de 40     | 4               | \$0.40                | \$ 1.60            |
| 18                                 | Tornillos                           | c/u           | 10              | \$ 0.15               | \$ 1.50            |
| 19                                 | Cable de Timbre                     | m.            | 8               | \$ 0.50               | \$ 4.00            |
| 20                                 | Placa PCB                           | c/u           | 1               | \$ 25.00              | \$ 25.00           |
| 21                                 | Estaño                              | c/u           | 1               | \$ 1.70               | \$ 1.70            |

**Elaborado por:** El Investigador

**Tabla 14.** Presupuesto de Construcción 2

| <b>Presupuesto de Construcción</b> |                          |               |                 |                       |                    |
|------------------------------------|--------------------------|---------------|-----------------|-----------------------|--------------------|
| <b>Ítem</b>                        | <b>Descripción</b>       | <b>Unidad</b> | <b>Cantidad</b> | <b>Costo Unitario</b> | <b>Costo Total</b> |
| 22                                 | Cautín de Madera         | c/u           | 1               | \$ 3.00               | \$ 3.00            |
| 23                                 | Pasta para Soldar        | c/u           | 1               | \$ 2.50               | \$ 2.50            |
| 25                                 | Celda de Carga con HX711 | c/u           | 1               | \$ 18.00              | \$ 18.00           |
| 26                                 | Internet                 | Meses         | 6               | \$ 22.00              | \$ 132.00          |
| 27                                 | Cortes de Madera MDF     | varios        |                 | \$ 35.00              | \$ 35.00           |
| 28                                 | Brocas para Baquelita    | c/u           | 3               | \$ 2.50               | \$ 7.50            |
| 29                                 | Pistola de Silicón       | c/u           | 1               | \$ 4.00               | \$ 4.00            |
| 30                                 | Barras de Silicón        | c/u           | 7               | \$ 0.50               | \$ 3.50            |
| 31                                 | Frutos Secos             | varios        |                 | \$ 20.00              | \$ 20.00           |
| 32                                 | Borneras de 2 patas 5mm  | c/u           | 15              | \$ 0.25               | \$ 3.75            |
| <b>Subtotal</b>                    |                          |               |                 |                       | \$ 617,57          |
| <b>Imprevistos (15%)</b>           |                          |               |                 |                       | \$ 92.64           |
| <b>TOTAL</b>                       |                          |               |                 |                       | \$ 710.21          |

**Elaborado por:** El Investigador

Por lo tanto, el presupuesto total de la implementación del prototipo es detallada en la Tabla 14.

**Tabla 15.** Presupuesto total de implementación

| <b>Presupuesto total de Implementación</b> |                       |                    |
|--|-----------------------|--------------------|
| <b>Ítem</b>                                | <b>Descripción</b>    | <b>Costo Total</b> |
| <b>1</b>                                   | Costo de Diseño       | \$ 500.40          |
| <b>2</b>                                   | Costo de Construcción | \$ 710.21          |
| <b>TOTAL</b>                               |                       | <b>\$ 1210.61</b>  |

**Elaborado por:** El Investigador

## CAPÍTULO IV

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1 Conclusiones

- La Industria 4.0 presenta múltiples mejoras con respecto a las tres revoluciones industriales previas, las mismas que, hacen posible la personalización en masa de productos por medio de la implementación de tecnologías adecuadas con el requisito fundamental que estas deben estar hiperconectadas entre sí a través de una arquitectura IoT.
- La arquitectura IoT de 3 capas utilizada, junto con el procesamiento de datos Fog computing, reducen la latencia del sistema y disminuyen la sobrecarga de datos en el servidor, debido a que, los microcontroladores implementados Arduino Mega 2560 y NodeMCU ESP8266 envían datos a la nube únicamente si existe una medición en sus sensores distinta a la última almacenada. Con respecto al software implementado, las interfaces web al estar desarrolladas con herramientas como HTML, CSS, JavaScript, PHP y SQL, permiten escalabilidad en el sistema y son aptas para ser modificadas ante cualquier requerimiento de los usuarios.
- El prototipo dispensa cantidades reales que varían entre 0 a +2 gramos por cada pedido, esto se debe a que, cada grano en particular tiene un peso único y ligeramente distinto a otro grano de la misma categoría, sin embargo, las variaciones se encuentran dentro de un rango totalmente aceptable por la empresa; la lógica de programación del prototipo funciona de manera que, la cantidad real dispensada por cada pedido jamás será menor a la cantidad solicitada por el cliente.
- Las funciones PHP strip\_tags() y sha1(), junto con el atributo HTML “required” de los elementos de tipo INPUT, evitan totalmente los intentos de ataques al medio de comunicación e interacción del sistema, que es la Base de

Datos MySQL, por medio de usuarios que busquen inyectar código SQL, HTML o PHP en los formularios de registro con fines maliciosos.

- El control de variables globales de sesiones PHP por medio de `SESSION_START()`, permite que los dos tipos de usuarios (administradores y clientes) ingresen a sus respectivas interfaces web por medio de una misma página de Log In, a la vez, evita el acceso a las páginas web a usuarios que no estén autorizados ni registrados previamente.

## 4.2 Recomendaciones

- Se sugiere verificar detalladamente que la dirección IP en donde se encuentre alojada la programación web, es decir la IP del servidor, coincida con las direcciones establecidas en el algoritmo de programación de la NodeMCU, debido a que, este microcontrolador se comunicará directamente con archivos PHP almacenados en dicho servidor web.
- Para lograr una medición de datos más estable de los contenedores, se recomienda implementar sensores ultrasónicos de mayor precisión y presupuesto, debido a que, los frutos secos no presentan superficies planas, por lo tanto, la onda ultrasónica con la que trabaja el sensor HC-SR04 puede rebotar en sentidos erróneos al receptor, provocando en algunas ocasiones, una medición incorrecta.
- Se recomienda indagar sobre distintos frameworks o entornos de programación que soporten más protocolos de mensajería compatibles con JavaScript, como NodeJS, ReactJS, Angular, etc. Para de esta forma, ampliar las posibilidades de mensajería de la arquitectura IoT propuesta a MQTT y CoAp en futuros proyectos.
- El recipiente o vaso en el cual caen los granos dispensados debe ser colocado encima de la celda de carga antes de encender el prototipo, en caso contrario,

se recomienda reiniciar el sistema para que el microcontrolador Arduino registre el peso del recipiente como referencia.

## BIBLIOGRAFÍA

- [1] D. Guevara y J. Guaca, Dispensador Mascotas Club Controlado Remotamente desde la Web. Bogotá: FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES, 2017.
- [2] C. J. Pérez, "Investigación de pre Factibilidad para la Fabricación de dispensador de comida para mascotas automatizado", Lima: Universidad San Ignacio De Loyola, 2018
- [3] J. A. Cabañeros, "Dispensador inteligente de medicamentos conectados y sistema de gestion", Tesis de Masterado, Escuela de Ingeniería Informática Madrid, UNIOVI, Oviedo, Art. 2018.
- [4] J. Escobar, "Diseño de sistemas de control industrial de robots basados en industria 4.0", Tesis de Ingeniería, FISEI, UTA, Ambato, ECU, 2019.
- [5] L. P. Davila, "Sistema electronico automatico de alimentacion para mascotas en el hogar", Tesis de Ingeniería, FISEI, UTA, Ambato, ECU, 2020
- [6] Val Román, J. "Industria 4.0: la transformación digital de la industria", Conferencia de Directores y Decanos de Ingeniería Informática, Bilbao, España. 2017, pp. 3-6.
- [7] N. Hernández. (2019, Marzo 13). "Personalización en Masa" [En línea]. Available: <https://hablemosdeempresas.com/grandes-empresas/industria-4-0-personalizacion-masiva/>.
- [8] C. Fenech y B. Perkins, "Made-to-order: The rise of mass personalisation", Deloitte, Londres, Inglaterra, 11.º ed. Deloitte Consumer Review, 2019.
- [9] A. Carrera. (2020, Marzo 11). "Industria 4.0 En El Ecuador" [En línea]. Available: <https://sanfranciscoglobal.org/industria-4-0-en-el-ecuador/>.
- [10] C. Martin, R. Samans, H. Leurent, F. Beti, M. Hanouz, T. Geiger, "Readiness for the Future of Production", The World Economic Forum, Cologny, Suiza, ISBN 978-1-944835-16-3, 2018.

- [11] Ministerio de Telecomunicaciones y de la Sociedad de la Información, "El MINTEL y el Municipio de Ambato trabajan para convertirla en Ciudad Inteligente", Gobierno de la República del Ecuador, Ambato, 2020.
- [12] S. Madakam, R. Ramaswamy y S. Tripathi, "Internet of Things (IoT): A Literature Review", *Journal of Computer and Communications*, vol. 3, n° <http://dx.doi.org/10.4236/jcc.2015.35021>, pp. 164-173, 2015.
- [13] K. Armijos, Interviewee, Automatización. [Entrevista]. 07 Enero 2022.
- [14] R. Rodríguez y J. Gómez, (2019). "Electrónica Digital", [En línea]. Available: [https://www.edu.xunta.gal/centros/iesblancoamorculledo/aulavirtual/pluginfile.php/37095/mod\\_resource/content/2/analo%CC%81gica.pdf#:~:text=Los%20sistemas%20electr%C3%B3nicos%20son%20conjuntos,teclado%2C%20temperatura%2C%20etc.](https://www.edu.xunta.gal/centros/iesblancoamorculledo/aulavirtual/pluginfile.php/37095/mod_resource/content/2/analo%CC%81gica.pdf#:~:text=Los%20sistemas%20electr%C3%B3nicos%20son%20conjuntos,teclado%2C%20temperatura%2C%20etc.).
- [15] J. Crespo, (2016, Diciembre 18) "Aprendiendo Arduino". [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2016/12/18/sensores-y-actuadores/>.
- [16] MecatrónicaLATAM, "¿Qué es la Mecatrónica?", Editorial Polar, 2021.
- [17] Keyence Group, (2021, Mayo). "¿Qué es un sensor ultrasónico?". [En línea]. Available: <https://www.keyence.com.mx/ss/products/sensor/sensorbasics/ultrasonic/info/>.
- [18] R. Carrera. (2020, Agosto 26). "Sensor Fotoeléctrico". [En línea]. Available: <https://infosensores.com/sensor-fotoelectrico/>.
- [19] Editorial Seguridad 360. (2022, Enero 20). "¿Qué es un sensor magnético?" [En línea]. Available: <https://revistaseguridad360.com/destacados/sensor-magnetico/>.
- [20] J. M. Zafrilla. (2019, Diciembre 12). "PCE". [En línea]. Available: <https://www.pce-iberica.es/instrumentos-de-medida/sistemas/controladores-digitales.htm>.
- [21] R. Marmolejo. (2021, Agosto 11). "Hetpro". [En línea]. Available: <https://hetpro-store.com/TUTORIALES/microcontrolador/>.
- [22] A. Moises, Internet de las Cosas, 2da. ed. Madrid: Reus, 2018.



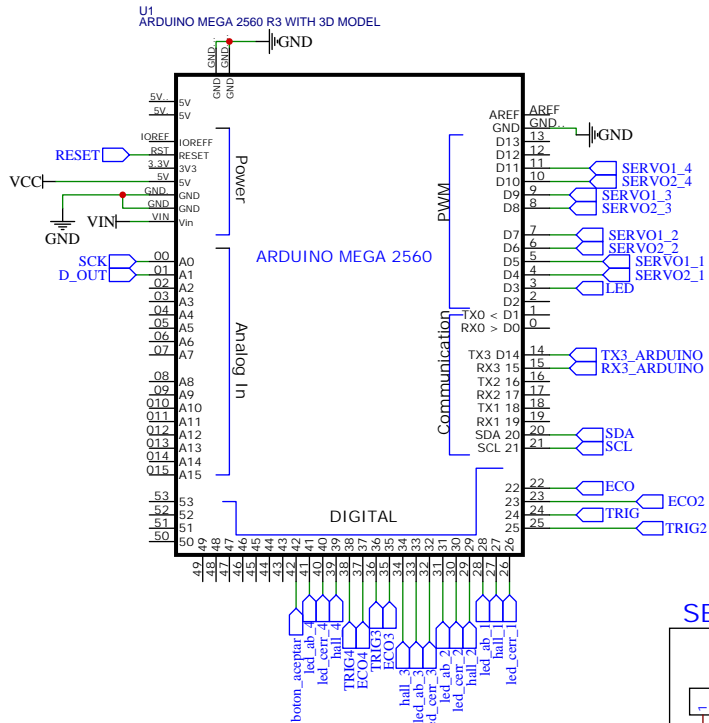
- [23] "Aprendiendo Arduino", 2018. [En línea]. Available: <https://aprendiendoarduino.wordpress.com/category/conectividad-iot/>.
- [24] L. J. Aguilar. *Industria 4.0 La Cuarta Revolución Industrial*. 1ra Ed. México. 2017
- [25] El Universo, (2018, Marzo 19). " La cuarta revolución industrial es digital". [En línea]. Available: <https://www.eluniverso.com/tendencias/2018/03/19/nota/6675070/cuarta-revolucion-industrial-es-digital/>.
- [26] ISOTools, (2018, Julio 12). "Industria 4.0". [En línea]. Available: <https://www.isotools.org/2018/07/12/industria-4-0-que-debemos-saber/>.
- [27] M. Juez. (2017), "La Personalización en Masa" [En línea]. Available: <https://www.beetrack.com/es/blog/produccion-del-futuro-personalizada-y-masiva>.
- [28] F. Herranz. (2021. Diciembre 20). "La industria 4.0 sitúa a la persona en el centro". [En línea]. Available: <https://industrytalks.es/la-industria-4-0-situa-a-la-persona-en-el-centro/>.
- [29] F. Panceri, (2021, Octubre 10). "IoT aplicado al marketing". [En línea]. Available: <https://www.mundopmmi.com/automatizacion/internet-industrial-de-las-cosas-iiot/article/21771544/el-internet-industrial-de-las-cosas-y-el-marketing>.
- [30] F. Moya, (2022). "Protocolos de comunicación inalámbricos más comunes en IoT". [En línea]. Available: <https://www.electrodaddy.com/internet-de-las-cosas-protocolos-de-comunicacion-inalambricos-mas-comunes-iiot/>.
- [31] A. Vélez, "Arquitecturas De Referencia Para Iot Con Transferencia Segura De Información", trabajo fin de máster, ECBTI, UNAD, Tuluá, COL, 2019
- [32] "Aprendiendo arduino" 2019. [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2018/11/17/protocolos-iiot-capacidad-aplicacion/>.
- [33] F. Chicaiza, W. Rivera, " Diseño Y Desarrollo De Una Plataforma Prototipo Para Visualización, Control Y Comunicación Con Raspberry Pi En Ambientes Iot", Tesis de Ingeniería, Sistemas, UPS, Quito, ECU, 2021.

- [34] O. Rius, (2021, Junio 14). " Cloud, Edge, and Fog Computing". [En línea]. Available: <https://www.e-zigurat.com/innovation-school/blog/cloud-edge-fog-computing-practical-applications/>.
- [35] E. Rodal, (2021, Mayo 24). " Fog computing, edge computing y cloud computing: diferencias". [En línea]. Available: <https://www.podcastindustria40.com/fog-computing/>
- [36] R. Vayas, (2021, Agosto 04). "Principales modelos de servicio cloud: IaaS, PaaS y SaaS". [En línea]. Available: <https://www.stackscale.com/es/blog/modelos-de-servicio-cloud/>.
- [37] H. Mendoza, (2019, Mayo 16). "5 requerimientos de una arquitectura IoT". [En línea]. Available: <https://www.chakray.com/es/5-requisitos-de-una-arquitectura-iot/>.
- [38] J. Green, B. McCarson, M. Devine, "Building the Internet of Things", Cisco, Illinois, Chicago, IoT World Forum, 2019.
- [39] A. Lima, (2022). "Arquitectura De 5 Capas De Internet De Las Cosas". [En línea]. Available: <https://es.acervolima.com/arquitectura-de-5-capas-de-internet-de-las-cosas/>.
- [40] IBM Cloud Education, (2020, Octubre 28). "Arquitectura de tres niveles". [En línea]. Available: <https://www.ibm.com/mx-es/cloud/learn/three-tier-architecture>.
- [41] C. Juárez, (2020, Abril 09) "Codificación industrial" [En línea]. Available: <https://thefoodtech.com/maquinaria-para-ensado-y-procesamiento/codificacion-de-productos-desde-la-basica-hasta-la-serializacion-en-masa/>.
- [42] Marketing Vending, (2020, Marzo 24) "¿Qué es el Vending?" [En línea]. Available: [https://www.marketingvending.com/que\\_es\\_vending.htm](https://www.marketingvending.com/que_es_vending.htm).
- [43] Personal de Mayo Clinic, (2019, Noviembre 14). "Los frutos secos y el corazón". [En línea]. Available: <https://www.mayoclinic.org/es-es/diseases-conditions/heart-disease/in-depth/nuts/art-20046635>.

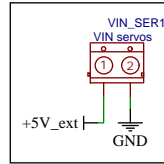
- [44] European Knowledge Center for Information Technology (Ed.). (2019, Julio 8). “Servidores”. [En línea]. Available: <https://www.ticportal.es/glosario-tic/servidores>.
- [45] R. Marín, (2019, Abril 16). “Los gestores de bases de datos más usados en la actualidad”. [En línea]. Available: <https://www.inesem.es/revistadigital/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>.
- [46] I. de Souza, (2020, Marzo 2). “Entiende las diferencias entre Front-End y Back-end en el ambiente de los sitios web”. [En línea]. Available: <https://rockcontent.com/es/blog/front-end-y-back-end/>.
- [47] G. Munte, (2020, Enero 8). “Guía completa del Framework: qué es, cuáles tipos existen y por qué es importante en Internet”. [En línea]. Available: <https://rockcontent.com/es/blog/framework/>.
- [48] I. Palma, (2018, Abril 24). “Sésamo O Ajonjolí: La Semilla Rica En Calcio”. [En línea]. Available: <https://www.conasi.eu/blog/consejos-de-salud/consejos-de-salud-consejos-de-salud/ajonjoli-sesamo-semilla-calcio/>.
- [49] American Peanut Council, (2019, Octubre 14). “7 Poderosas razones para comer mani”. [En línea]. Available: <https://www.saborusa.com/blog/siete-poderosas-razones-para-comer-mani/>.
- [50] J. Fernández, (2022, Enero 14). “¿Qué variedades de pistachos existen?”. [En línea]. Available: <https://www.salysol.com/que-variedades-de-pistachos-existen>.
- [51] G. Meca, (2020, Octubre 29). “Recetas con nueces”. [En línea]. Available: <https://okdiario.com/recetas/recetas-nueces-aprovecharlas-dia-todos-santos-6356957>
- [52] LOTAIP, “Remuneración mensual por puesto”, CNT, Quito, Ecuador, Art. 7, 2022.

# **ANEXOS**

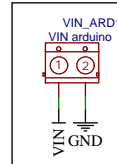
# ANEXO 1



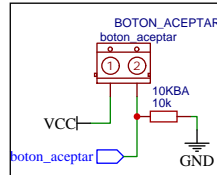
## Fuente Servos



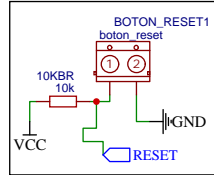
## Fuente Arduino



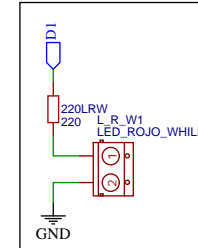
## BOTON ACEPT.



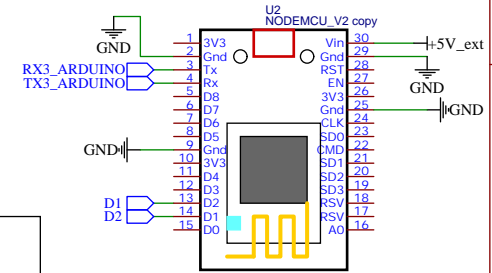
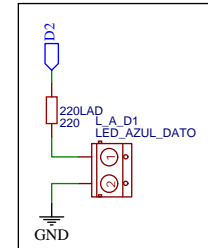
## RESET ARD.



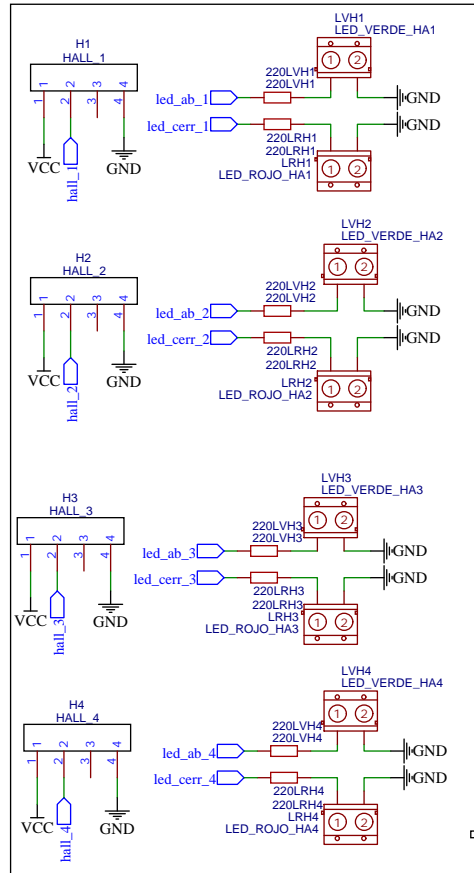
## LED WHILE



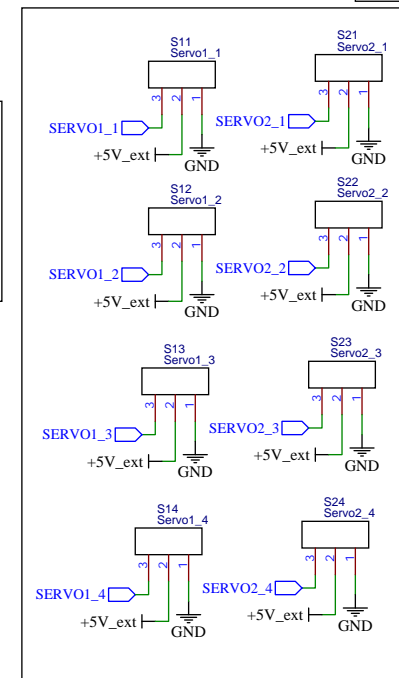
## LED DATO



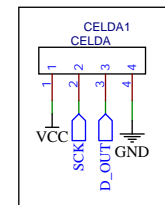
## SENSORES HALL Y LEDS DE ESTADO



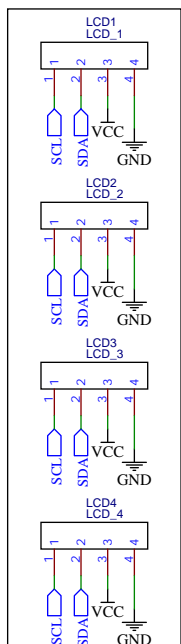
## SERVOMOTORES



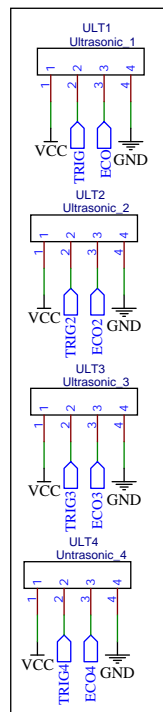
## CELDA



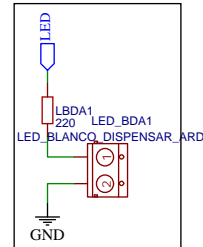
## PANTALLAS



## ULTRASONICOS



## LED DISPENSAR



## Anexo 2: Programación del Microcontrolador Arduino MEGA

```
#include <SoftwareSerial.h>

// Librerias pantalla LCD I2C =====
#include <Wire.h>      // libreria de comunicacion por I2C
#include <LCD.h>       // libreria para funciones de LCD
#include <LiquidCrystal_I2C.h>  // libreria para LCD por I2C
//=====

// Librerias celda de carga =====
#include <HX711_ADC.h>
#if defined(ESP8266) || defined(ESP32) || defined(AVR)
#include <EEPROM.h>
#endif
//=====

// Librerias servo =====
#include <Servo.h>
//=====

// Pines y objeto pantalla LCD I2C =====
LiquidCrystal_I2C lcd_grano1 (0x27, 2, 1, 0, 4, 5, 6, 7); // DIR, E, RW, RS, D4, D5,
D6, D7
LiquidCrystal_I2C lcd_grano2 (0x23, 2, 1, 0, 4, 5, 6, 7);
LiquidCrystal_I2C lcd_grano3 (0x25, 2, 1, 0, 4, 5, 6, 7);
LiquidCrystal_I2C lcd_grano4 (0x26, 2, 1, 0, 4, 5, 6, 7);
//=====

//                                     Pines                                     Ultrasonico1
Ajonjoli=====
int TRIG=24;
int ECO=22;
int LED=3;
float DURACION;
float DISTANCIA;

float PORCENTAJE1 = 0;
float PORCENTAJE1_anterior = 0;
//=====

// Pines Ultrasonico2 Mani=====
int TRIG2=25;
int ECO2=23;
float DURACION2;
float DISTANCIA2;

float PORCENTAJE2 = 0;
float PORCENTAJE2_anterior = 0;
```

```

//=====
// Pines Ultrasonico3 PISTACHO=====
int TRIG3=36;
int ECO3=35;
float DURACION3;
float DISTANCIA3;

float PORCENTAJE3 = 0;
float PORCENTAJE3_anterior = 0;
//=====

// Pines Ultrasonico4 NUEZ=====
int TRIG4=38;
int ECO4=37;
float DURACION4;
float DISTANCIA4;

float PORCENTAJE4 = 0;
float PORCENTAJE4_anterior = 0;
//=====

// Pines HALL_1 AJONJOLI =====
#define ledAbierto1 28
#define sensor_Hall_1 27
#define ledCerrado1 26

String estado1="";
String estado1_anterior="";
//=====

// Pines HALL_2 MANI =====
#define ledAbierto2 31
#define sensor_Hall_2 29
#define ledCerrado2 30

String estado2="";
String estado2_anterior="";
//=====

// Pines HALL_3 PISTACHO =====
#define ledAbierto3 33
#define sensor_Hall_3 34
#define ledCerrado3 32

String estado3="";
String estado3_anterior="";
//=====

// Pines HALL_4 NUEZ =====

```

```

#define ledAbierto4 41
#define sensor_Hall_4 39
#define ledCerrado4 40

String estado4="";
String estado4_anterior="";
//=====

//=====VARIABLES DE ENVIO DE DATOS PARA 3 CIFRAS =====
String P_1="";
String P_2="";
String P_3="";
String P_4="";

//=====VARIABLES recepcion de DATOS PEDIDO=====
int PED_G1_num=0;
int PED_G2_num=0;
int PED_G3_num=0;
int PED_G4_num=0;
int PED_PR_num=0;
int PED_VT_num=0;
//=====

//Pines y objeto celda de carga =====
const int HX711_dout = A1; //mcu > HX711 dout pin
const int HX711_sck = A0; //mcu > HX711 sck pin

//HX711 constructor:
HX711_ADC LoadCell(HX711_dout, HX711_sck);

const int calVal_eeepromAdress = 0;
unsigned long t = 0;
float peso1=0;
float peso2=0;
float peso3=0;
float peso4=0;
float peso_retirar;

float peso_salida;
float aux1=0;

//=====

//objetos de todos los servos =====
Servo servo1_1; // ajonjolí
Servo servo2_1;

Servo servo1_2; // maní

```



```

Servo servo2_2;

Servo servo1_3; // pistacho
Servo servo2_3;

Servo servo1_4; // nuez
Servo servo2_4;
//=====

#define boton_aceptar 42

void setup(){
//Setup pantalla lcd_grano1 =====

  lcd_grano1.setBacklightPin(3,POSITIVE); // puerto P3 de PCF8574 como positivo
  lcd_grano1.setBacklight(HIGH); // habilita iluminacion posterior de lcd_grano1
  lcd_grano1.begin(16, 2); // 16 columnas por 2 lineas para lcd_grano1 1602A
  lcd_grano1.clear(); // limpia pantalla
//=====

//Setup pantalla lcd_grano2 =====

  lcd_grano2.setBacklightPin(3,POSITIVE); // puerto P3 de PCF8574 como positivo
  lcd_grano2.setBacklight(HIGH); // habilita iluminacion posterior de lcd_grano2
  lcd_grano2.begin(16, 2); // 16 columnas por 2 lineas para lcd_grano1 1602A
  lcd_grano2.clear(); // limpia pantalla
//=====

//Setup pantalla lcd_grano3 =====

  lcd_grano3.setBacklightPin(3,POSITIVE); // puerto P3 de PCF8574 como positivo
  lcd_grano3.setBacklight(HIGH); // habilita iluminacion posterior de lcd_grano2
  lcd_grano3.begin(16, 2); // 16 columnas por 2 lineas para lcd_grano1 1602A
  lcd_grano3.clear(); // limpia pantalla
//=====

//Setup pantalla lcd_grano4 =====

  lcd_grano4.setBacklightPin(3,POSITIVE); // puerto P3 de PCF8574 como positivo
  lcd_grano4.setBacklight(HIGH); // habilita iluminacion posterior de lcd_grano2
  lcd_grano4.begin(16, 2); // 16 columnas por 2 lineas para lcd_grano1 1602A
  lcd_grano4.clear(); // limpia pantalla
//=====

//setup pines TODOS ultrasonicos =====
  pinMode(TRIG, OUTPUT);
  pinMode(ECO, INPUT);
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
  Serial3.begin(9600);

```

```

//Serial3.print("RETIRADO");

pinMode(TRIG2, OUTPUT);
pinMode(ECO2, INPUT);

pinMode(TRIG3, OUTPUT);
pinMode(ECO3, INPUT);

pinMode(TRIG4, OUTPUT);
pinMode(ECO4, INPUT);
//=====

pinMode(boton_aceptar, INPUT);

//setup          TODOS          LOS          HALL
=====

pinMode(ledAbierto1, OUTPUT);
pinMode(ledCerrado1, OUTPUT);
pinMode(sensor_Hall_1, INPUT);

pinMode(ledAbierto2, OUTPUT);
pinMode(ledCerrado2, OUTPUT);
pinMode(sensor_Hall_2, INPUT);

pinMode(ledAbierto3, OUTPUT);
pinMode(ledCerrado3, OUTPUT);
pinMode(sensor_Hall_3, INPUT);

pinMode(ledAbierto4, OUTPUT);
pinMode(ledCerrado4, OUTPUT);
pinMode(sensor_Hall_4, INPUT);
//=====

// Setup celda de carga =====
    LoadCell.begin();
    //LoadCell.setReverseOutput(); //uncomment to turn a negative output value to
positive
    float calibrationValue; // calibration value (see example file
"Calibration.ino")
    calibrationValue = 462.18; // uncomment this if you want to set the calibration
value in the sketch
    #if defined(ESP8266) || defined(ESP32)
    //EEPROM.begin(512); // uncomment this if you use ESP8266/ESP32 and want to
fetch the calibration value from eeprom
    #endif
    //EEPROM.get(calVal_eepromAdress, calibrationValue); // uncomment this if you
want to fetch the calibration value from eeprom

    unsigned long stabilizingtime = 2000; // preciscion right after power-up can
be improved by adding a few seconds of stabilizing time

```

```

        boolean _tare = true; //set this to false if you don't want tare to be performed
in the next step
        LoadCell.start(stabilizingtime, _tare);
        if (LoadCell.getTareTimeoutFlag()) {
            //Serial.println("Timeout, check MCU>HX711 wiring and pin designations");
            while (1);
        }
        else {
            LoadCell.setCalFactor(calibrationValue); // set calibration value (float)
            //Serial.println("Startup is complete");
        }
    }
//=====

// setup servos TODOS =====
    servo1_1.attach(5,100,2500);// pin del servo, pulso minimo y pulso
maximo //normalmente un pulso de 1000 microsegundos equivale a 0 grados y un pulso
de 2000 equivale a 180 grados
    servo2_1.attach(4,100,2500); //ajonjolo

    servo1_2.attach(7,100,2500); // mani
    servo2_2.attach(6,100,2500);

    servo1_3.attach(9,100,2500); // pistacho
    servo2_3.attach(8,100,2500);

    servo1_4.attach(11,100,2500); //nuez
    servo2_4.attach(10,100,2500);
//=====
}

void ultrasonic1() {
//=====A
JONJOLI

    digitalWrite(TRIG, HIGH);
    delay(1);
    digitalWrite(TRIG, LOW);
    DURACION= pulseIn(ECO,HIGH); //devuelve un valor de tiempo en microsegundos
//espera por un pulso alto

    DISTANCIA=DURACION/58.2; //constante especificada por el fabricante para
transformar tiempo a distancia

    if ((DISTANCIA>=10) && (DISTANCIA<=14)){ // 0 %
        PORCENTAJE1=0;
    }
    else if ((DISTANCIA<10) && (DISTANCIA>=8.5)) { //MAP
        PORCENTAJE1 = 20.00;
    }
}

```

```

else if ((DISTANCIA<8.5) && (DISTANCIA>=7)) { //MAP
    PORCENTAJE1 = 40.00;
}

else if ((DISTANCIA<7) && (DISTANCIA>=5.5)) { //MAP
    PORCENTAJE1 = 60.00;
}

else if ((DISTANCIA<5.5) && (DISTANCIA>=4)) { //MAP
    PORCENTAJE1 = 80.00;
}

else if ((DISTANCIA<4) && (DISTANCIA>=2)) { //MAP
    PORCENTAJE1 = 100;
}
else { //100
    PORCENTAJE1 = 100;
}

// Impresión en pantalla
lcd_grano1.setCursor(0,0);
lcd_grano1.print(int(round(PORCENTAJE1)));
lcd_grano1.print(" % ");
}

//=====

void ultrasonico2() {
//=====

    digitalWrite(TRIG2, HIGH);
    delay(1);
    digitalWrite(TRIG2, LOW);
    DURACION2= pulseIn(ECO2,HIGH); //devuelve un valor de tiempo en microsegundos
    //espera por un pulso alto

    DISTANCIA2=DURACION2/58.2; //constante especificada por el fabricante para
    transformar tiempo a distancia

    if ((DISTANCIA2>=10) && (DISTANCIA2<=14)){ // 0 %
        PORCENTAJE2=0;
    }
    else if ((DISTANCIA2<10) && (DISTANCIA2>=8.5)) { //MAP
        PORCENTAJE2 = 20;
    }

    else if ((DISTANCIA2<8.5) && (DISTANCIA2>=7)) { //MAP
        PORCENTAJE2 = 40;
    }
}

```

```

else if ((DISTANCIA2<7) && (DISTANCIA2>=5.5)) { //MAP
    PORCENTAJE2 = 60;
}

else if ((DISTANCIA2<5.5) && (DISTANCIA2>=4)) { //MAP
    PORCENTAJE2 = 80;
}

else if ((DISTANCIA2<4) && (DISTANCIA2>=2)) { //MAP
    PORCENTAJE2 = 100;
}
else { //100
    PORCENTAJE2 = 100;
}

// Impresión en pantalla
lcd_grano2.setCursor(0,0);
lcd_grano2.print(int(round(PORCENTAJE2)));
lcd_grano2.print(" % ");
}

//=====

void ultrasonico3() {
//=====

    digitalWrite(TRIG3, HIGH);
    delay(1);
    digitalWrite(TRIG3, LOW);
    DURACION3= pulseIn(ECO3,HIGH); //devuelve un valor de tiempo en microsegundos
        //espera por un pulso alto

    DISTANCIA3=DURACION3/58.2; //constante especificada por el fabricante para
transformar tiempo a distancia

    if ((DISTANCIA3>=9) && (DISTANCIA3<=14)){ // 0 %
        PORCENTAJE3=0;
    }
    else if ((DISTANCIA3<9) && (DISTANCIA3>=7.3)) { //MAP
        PORCENTAJE3=25;
    }
    else if ((DISTANCIA3<7.3) && (DISTANCIA3>=5.6)) { //MAP
        PORCENTAJE3=50;
    }
    else if ((DISTANCIA3<5.6) && (DISTANCIA3>=3.9)) { //MAP
        PORCENTAJE3=75;
    }
    else if ((DISTANCIA3<3.9) && (DISTANCIA3>=2.0)) { //MAP

```

```

    PORCENTAJE3=100;
}
else { //100
    PORCENTAJE3 = 100;
}

// IMPresión en pantalla
    lcd_grano3.setCursor(0,0);
    lcd_grano3.print(int(round(PORCENTAJE3)));
    lcd_grano3.print(" % ");
}

//=====

void ultrasonico4() {
//=====

    digitalWrite(TRIG4, HIGH);
    delay(1);
    digitalWrite(TRIG4, LOW);
    DURACION4= pulseIn(ECO4,HIGH); //devuelve un valor de tiempo en microsegundos
        //espera por un pulso alto

    DISTANCIA4=DURACION4/58.2; //constante especificada por el fabricante para
transformar tiempo a distancia
    //Serial.println(DISTANCIA);
    //delay(200);
    /* if (DISTANCIA4<=10 && DISTANCIA4>=0){// S DISTANCIA ENTRE 0 Y 10
        digitalWrite(LED,HIGH);
        delay(DISTANCIA4*10);
        digitalWrite(LED,LOW);
    }*/

    if ((DISTANCIA4>=9) && (DISTANCIA4<=14)){ // 0 %
        PORCENTAJE4=0;
    }
    else if ((DISTANCIA4<9) && (DISTANCIA4>=7.3)) { //MAP
        PORCENTAJE4=25;
    }
    else if ((DISTANCIA4<7.3) && (DISTANCIA4>=5.6)) { //MAP
        PORCENTAJE4=50;
    }
    else if ((DISTANCIA4<5.6) && (DISTANCIA4>=3.9)) { //MAP
        PORCENTAJE4=75;
    }
    else if ((DISTANCIA4<3.9) && (DISTANCIA4>=2.0)) { //MAP
        PORCENTAJE4=100;
    }
    else { //100

```

```

PORCENTAJE4 = 100;
}

// Impresión en pantalla
lcd_grano4.setCursor(0,0);
lcd_grano4.print(int(round(PORCENTAJE4)));
lcd_grano4.print(" % ");
}

//=====

void sensor_hall_1() {
  if (digitalRead(sensor_Hall_1) == LOW){// NO DETECTA NADA ALGUIEN ABRIÓ
    //Serial.println("Detecto algo Negro");
    digitalWrite(ledCerrado1, HIGH);
    digitalWrite(ledAbierto1, LOW);
    estado1 = "ABIERTO ";
  }
  else{
    //Serial.println("Decectó algo Blanco"); // DETECTA IMAN, CERRADO
    digitalWrite(ledAbierto1, HIGH);
    digitalWrite(ledCerrado1, LOW);
    estado1 = "Cerrado ";
  }

  // Impresión en pantalla
  lcd_grano1.setCursor(8,0);
  lcd_grano1.print(estado1);
  //delay(10);
}

//=====

void sensor_hall_2() {
  if (digitalRead(sensor_Hall_2) == HIGH){// Detecto algo ESTA CERRADA LA TAPA OK
    //Serial.println("Detecto algo Negro");
    digitalWrite(ledCerrado2, HIGH);
    digitalWrite(ledAbierto2, LOW);
    estado2 = "ABIERTO ";
  }
  else{
    //Serial.println("Decectó algo Blanco"); // No detecta nada ALGUIEN ABRIÓ
    digitalWrite(ledAbierto2, HIGH);
    digitalWrite(ledCerrado2, LOW);
    estado2 = "Cerrado ";
  }
}

```

```

    // Impresión en pantalla
    lcd_grano2.setCursor(8,0);
    lcd_grano2.print(estado2);
    //delay(10);
}

//=====

void sensor_hall_3() {
    if (digitalRead(sensor_Hall_3) == HIGH){// Detecto algo ESTA CERRADA LA TAPA OK
        //Serial.println("Detecto algo Negro");
        digitalWrite(ledCerrado3, HIGH);
        digitalWrite(ledAbierto3, LOW);
        estado3 = "ABIERTO ";
    }
    else{
        //Serial.println("Decectó algo Blanco"); // No detecta nada ALGUIEN ABRIO
        digitalWrite(ledAbierto3, HIGH);
        digitalWrite(ledCerrado3, LOW);
        estado3 = "Cerrado ";
    }
    // Impresión en pantalla
    lcd_grano3.setCursor(8,0);
    lcd_grano3.print(estado3);
    //delay(10);
}

//=====

//=====

void sensor_hall_4() {
    if (digitalRead(sensor_Hall_4) == HIGH){// Detecto algo ESTA CERRADA LA TAPA OK
        //Serial.println("Detecto algo Negro");
        digitalWrite(ledCerrado4, HIGH);
        digitalWrite(ledAbierto4, LOW);
        estado4 = "ABIERTO ";
    }
    else{
        //Serial.println("Decectó algo Blanco"); // No detecta nada ALGUIEN ABRIO
        digitalWrite(ledAbierto4, HIGH);
        digitalWrite(ledCerrado4, LOW);
        estado4 = "Cerrado ";
    }
    // Impresión en pantalla
    lcd_grano4.setCursor(8,0);
    lcd_grano4.print(estado4);
    //delay(10);
}

```



```

//=====

void pesar_grano1() {
    static boolean newDataReady = 0;
    const int serialPrintInterval = 0; //increase value to slow down serial print
activity

    // check for new data/start next conversion:
    if (LoadCell.update()) newDataReady = true;

    // get smoothed value from the dataset:
    if (newDataReady) {
        if (millis() > t + serialPrintInterval) {
            peso1 = LoadCell.getData();
            //Serial.print("Load_cell output val: ");
            //Serial.println(peso);

            // impresion en pantalla
            lcd_grano1.setCursor(0,1);
            if ((peso1<=0)||((peso1>0) && (peso1<0.4))){
                lcd_grano1.print("0.00  ");
            }
            else {
                lcd_grano1.print(peso1,1); // se imprime con un decimal
                lcd_grano1.print("  ");
            }
            delay(5);

            newDataReady = 0;
            t = millis();
        }
    }

    /* // receive command from serial terminal, send 't' to initiate tare operation:
    if (Serial.available() > 0) {
        char inByte = Serial.read();
        if (inByte == 't') LoadCell.tareNoDelay();
    }*/

    // check if last tare operation is complete:
    /* if (LoadCell.getTareStatus() == true) {
        Serial.println("Tare complete");
    }*/
}

//=====

void pesar_grano2() {
    static boolean newDataReady = 0;

```

```

    const int serialPrintInterval = 0; //increase value to slow down serial print
activity

// check for new data/start next conversion:
if (LoadCell.update()) newDataReady = true;

// get smoothed value from the dataset:
if (newDataReady) {
    if (millis() > t + serialPrintInterval) {
        peso2 = LoadCell.getData() - peso1;
        if (peso2<=0){
            peso2=0;
        }

        // impresion en pantalla
        lcd_grano2.setCursor(0,1);
        if ((peso2<=0)||((peso2>0) && (peso2<0.4))){
            lcd_grano2.print("0.00  ");
        }
        else {
            lcd_grano2.print(peso2,1); // se imprime con un decimal
            lcd_grano2.print("  ");
        }
        delay(5);

        newDataReady = 0;
        t = millis();
    }
}

//=====

void pesar_grano3() {
    static boolean newDataReady = 0;
    const int serialPrintInterval = 0; //increase value to slow down serial print
activity

// check for new data/start next conversion:
if (LoadCell.update()) newDataReady = true;

// get smoothed value from the dataset:
if (newDataReady) {
    if (millis() > t + serialPrintInterval) {
        peso3 = LoadCell.getData() - (peso2+peso1);
        if (peso3<=0){
            peso3=0;
        }

        // impresion en pantalla

```

```

    lcd_grano3.setCursor(0,1);
    if ((peso3<=0)||((peso3>0) && (peso3<0.4))){
        lcd_grano3.print("0.00  ");
    }
    else {
        lcd_grano3.print(peso3,1); // se imprime con un decimal
        lcd_grano3.print("  ");
    }
    delay(5);

    newDataReady = 0;
    t = millis();
}
}

//=====================================================

void pesar_grano4() {
    static boolean newDataReady = 0;
    const int serialPrintInterval = 0; //increase value to slow down serial print
    activity

    // check for new data/start next conversion:
    if (LoadCell.update()) newDataReady = true;

    // get smoothed value from the dataset:
    if (newDataReady) {
        if (millis() > t + serialPrintInterval) {
            peso4 = LoadCell.getData() - (peso3+peso2+peso1);
            if (peso4<=0){
                peso4=0;
            }

            // impresion en pantalla
            lcd_grano4.setCursor(0,1);
            if ((peso4<=0)||((peso4>0) && (peso4<0.4))){
                lcd_grano4.print("0.00  ");
            }
            else {
                lcd_grano4.print(peso4,1); // se imprime con un decimal
                lcd_grano4.print("  ");
            }
            delay(5);

            newDataReady = 0;
            t = millis();
        }
    }
}

```

```

}

//=====
//180 antihorario
//0 detenido
//90 horario

void dispensar_grano1() {
    // Ajonjoli

    servo1_1.write(180);
    servo2_1.write(0);
    delay(500);
    servo1_1.write(0);
    servo2_1.write(180);
    delay(720); // giro de 2 paletas exacto 360*2 grados 1gr aprox
    servo1_1.write(0);
    servo2_1.write(0);
    //delay(2000);
}

//=====

void dispensar_grano2() {
    // MANI

    servo2_2.write(180);
    servo1_2.write(180);
    delay(6400); // giro completo de 360 cuatro paletas 1600*4
    servo2_2.write(90);
    servo1_2.write(90);
    delay(5850); //1950*3
    servo1_2.write(0);
    servo2_2.write(0);
}

//=====

void dispensar_grano3() {
    servo1_3.write(180);
    servo2_3.write(0);
    delay(500);
    servo1_3.write(0);

    for (int i = 0; i <= 0; i++) { // PISTACHO
        servo2_3.write(180);
        servo1_3.write(0);
        delay(4000);
        servo2_3.write(90);
        delay(4000);
        servo2_3.write(0);
    }
}

```

```

}

//=====

//=====

void dispensar_grano4() {
    servo1_4.write(180);
    servo2_4.write(0);
    delay(900);
    servo1_4.write(90);
    delay(400);
    servo1_4.write(0);

    for (int i = 0; i <= 0; i++) {          // NUEZ
        servo2_4.write(180);
        servo1_4.write(0);
        delay(4000);
        servo2_4.write(90);
        delay(4000);
        servo2_4.write(0);
    }
}

//=====
//=====

void envio_datos() {

    // envio datos solo si hay algun valor nuevo
    if ( (int(round(PORCENTAJE1)) != int(round(PORCENTAJE1_anterior))) ||
(int(round(PORCENTAJE2)) != int(round(PORCENTAJE2_anterior))) ||
(int(round(PORCENTAJE3)) != int(round(PORCENTAJE3_anterior))) ||
(int(round(PORCENTAJE4)) != int(round(PORCENTAJE4_anterior))) || (estado1 !=
estado1_anterior) || (estado2 != estado2_anterior) || (estado3 !=
estado3_anterior) || (estado4 != estado4_anterior) ) {

        if (((int(round(PORCENTAJE1))/1)>=1) && ((int(round(PORCENTAJE1))/10)<1))
{ // numero de 1 cifra
            P_1 = "00" + String(int(round(PORCENTAJE1)));
        }
        else if (((int(round(PORCENTAJE1))/10)>=1) &&
((int(round(PORCENTAJE1))/100)<1)) { // numero de 2 cifras
            P_1 = "0" + String(int(round(PORCENTAJE1)));
        }

        else if ((int(round(PORCENTAJE1))/100)>=1) { //
numero de 3 cifras
            P_1 = String(int(round(PORCENTAJE1)));

```

```

    }

    else {
        P_1 ="000";
    }

    if (((int(round(PORCENTAJE2))/1)>=1) && ((int(round(PORCENTAJE2))/10)<1))
{ // numero de 1 cifra
    P_2 = "00" + String(int(round(PORCENTAJE2)));
}
    else if (((int(round(PORCENTAJE2))/10)>=1) &&
((int(round(PORCENTAJE2))/100)<1)) { // numero de 2 cifras
    P_2 = "0" + String(int(round(PORCENTAJE2)));
}

    else if ((int(round(PORCENTAJE2))/100)>=1) { //
numero de 3 cifras
    P_2 = String(int(round(PORCENTAJE2)));
}

    else {
        P_2 ="000";
    }

    if (((int(round(PORCENTAJE3))/1)>=1) && ((int(round(PORCENTAJE3))/10)<1))
{ // numero de 1 cifra
    P_3 = "00" + String(int(round(PORCENTAJE3)));
}
    else if (((int(round(PORCENTAJE3))/10)>=1) &&
((int(round(PORCENTAJE3))/100)<1)) { // numero de 2 cifras
    P_3 = "0" + String(int(round(PORCENTAJE3)));
}

    else if ((int(round(PORCENTAJE3))/100)>=1) { //
numero de 3 cifras
    P_3 = String(int(round(PORCENTAJE3)));
}
    else {
        P_3 ="000";
    }
}

```

```

        if (((int(round(PORCENTAJE4))/1)>=1) && ((int(round(PORCENTAJE4))/10)<1))
{ // numero de 1 cifra
    P_4 = "00" + String(int(round(PORCENTAJE4)));
}
else if (((int(round(PORCENTAJE4))/10)>=1) &&
((int(round(PORCENTAJE4))/100)<1)) { // numero de 2 cifras
    P_4 = "0" + String(int(round(PORCENTAJE4)));
}

else if ((int(round(PORCENTAJE4))/100)>=1) { //
numero de 3 cifras
    P_4 = String(int(round(PORCENTAJE4)));
}
else {
    P_4 ="000";
}

// PERO ANTES ENVIO TODOS CON TRES CIFRAS

String niveles = P_1 + "==" + P_2 + "==" + P_3 + "==" + P_4;
String estados = String(estado1) + "==" + String(estado2) + "==" +
String(estado3) + "==" + String(estado4);

String datos_de_control = "CONTROL==" + niveles + "==" + estados;

Serial3.println(datos_de_control);
delay(15);

PORCENTAJE1_anterior = PORCENTAJE1;
PORCENTAJE2_anterior = PORCENTAJE2;
PORCENTAJE3_anterior = PORCENTAJE3;
PORCENTAJE4_anterior = PORCENTAJE4;

estado1_anterior = estado1;
estado2_anterior = estado2;
estado3_anterior = estado3;
estado4_anterior = estado4;
}
else{
    PORCENTAJE1_anterior = PORCENTAJE1;
    PORCENTAJE2_anterior = PORCENTAJE2;
    PORCENTAJE3_anterior = PORCENTAJE3;
    PORCENTAJE4_anterior = PORCENTAJE4;

    estado1_anterior = estado1;
    estado2_anterior = estado2;
    estado3_anterior = estado3;
    estado4_anterior = estado4;
}
}

```

```

}

void lectura_datos() { // LEO SI HAY ALGUN PEDIDO

if (Serial3.available() > 0){
String bufferString = "";
while(Serial3.available() > 0){ bufferString += (char)Serial3.read(); }

String ECABEZADO = "";
String PED_G1 = "";
String PED_G2 = "";
String PED_G3 = "";
String PED_G4 = "";
String PED_PR = "";
String PED_VT = "";

for (int i = 0; i <= 10; i++) {
    ECABEZADO += bufferString[i]; //ENCABEZADO EXTRAIDO
}

for (int i = 13; i <= 15; i++) {
    PED_G1 += bufferString[i]; //PED GRANO 1 EXTRAIDO
    PED_G1_num = PED_G1.toInt();
}

for (int i = 18; i <= 20; i++) {
    PED_G2 += bufferString[i]; //PED GRANO 2 EXTRAIDO
    PED_G2_num = PED_G2.toInt();
}

for (int i = 23; i <= 25; i++) {
    PED_G3 += bufferString[i]; //PED GRANO 3 EXTRAIDO
    PED_G3_num = PED_G3.toInt();
}

for (int i = 28; i <= 30; i++) {
    PED_G4 += bufferString[i]; //PED GRANO 4 EXTRAIDO
    PED_G4_num = PED_G4.toInt();
}

for (int i = 36; i <= 40; i++) {
    PED_PR_num += bufferString[i]; //PED PRECIO EXTRAIDO
}

for (int i = 43; i <= 48; i++) {
    PED_VT_num += bufferString[i]; //PED VENTA EXTRAIDO
}

Serial3.flush();
delay(5);
/*Serial.println(ECABEZADO);
Serial.println(PED_G1_num);

```



```

Serial.println(PED_G2_num);
Serial.println(PED_G3_num);
Serial.println(PED_G4_num);*/

if (ECABEZADO=="PEDIDONUEVO"){           // confirmo si los datos son de algun
pedido nuevo
    digitalWrite(LED, HIGH);
    delay(10);
    //entro en modo dispensacion
    //peso todo
    //envio un caracter para que la nodeMCU me cierre el pedido
    lcd_grano1.clear();
    lcd_grano2.clear();
    lcd_grano3.clear();
    lcd_grano4.clear();

    lcd_grano1.print("DISPENSANDO");
    lcd_grano2.print("DISPENSANDO");
    lcd_grano3.print("DISPENSANDO");
    lcd_grano4.print("DISPENSANDO");

    while (digitalRead(boton_aceptar)==LOW){ // SOLO SI APLASTO EL BOTON ACEPTAN
CONTINUARÁ
        ;
        if (digitalRead(boton_aceptar)==HIGH){
            break;
        }
    }

    while (peso1 < PED_G1_num) { // DISPENSANDO GRANO 1 ====
        dispensar_grano1();
        for (long i = 0; i <= 600000; i++) { // se queda pesando correctamente
            pesar_grano1();
        }
    }

    while (peso2 < PED_G2_num) { // DISPENSANDO GRANO 2 ====
        dispensar_grano2();
        for (long i = 0; i <= 700000; i++) { // se queda pesando correctamente
            pesar_grano2();
        }
    }

    while (peso3 < PED_G3_num) { // DISPENSANDO GRANO 3 ====
        dispensar_grano3();
        for (long i = 0; i <= 700000; i++) { // se queda pesando correctamente
            pesar_grano3();
        }
    }
}

```

```

while (peso4 < PED_G4_num) { // DISPENSANDO GRANO 4 ====
    dispensar_grano4();
    for (long i = 0; i <= 700000; i++) { // se queda pesando correctamente
        pesar_grano4();
    }
}

// VERIFICANDO SI SE ESTA RETIRANDO EL VASO =====
peso_salida=peso1+peso2+peso3+peso4;
lcd_grano1.clear();
lcd_grano2.clear();
lcd_grano3.clear();
lcd_grano4.clear();
lcd_grano1.print("RETIRAR! ");
lcd_grano2.print("RETIRAR! ");
lcd_grano3.print("RETIRAR! ");
lcd_grano4.print("RETIRAR! ");

if (PED_G1_num==0){
    peso1=peso2+peso3+peso4;
}

while (peso_salida > 0) {
    pesar_grano1();
    peso_salida=peso1;

}
}

digitalWrite(LED, LOW);
lcd_grano1.clear();
lcd_grano2.clear();
lcd_grano3.clear();
lcd_grano4.clear();
peso1=0;
peso2=0;
peso3=0;
peso4=0;
;
//Serial3.flush();
Serial3.println("RETIRADO");
//Serial.println("RETIRADO");
delay(10);
}
}
//=====
void loop(){

    ultrasonico1();
    ultrasonico2();

```

```
ultrasonico3();
ultrasonico4();

sensor_hall_1();
sensor_hall_2();
sensor_hall_3();
sensor_hall_4();

lcd_grano1.setCursor(8,1);
lcd_grano1.print("Ajonjolí ");

lcd_grano2.setCursor(8,1);
lcd_grano2.print("Maní ");

lcd_grano3.setCursor(8,1);
lcd_grano3.print("Pistacho ");

lcd_grano4.setCursor(8,1);
lcd_grano4.print("Nuez ");

lectura_datos(); // LECTURA Y DISPENSACION
envio_datos();

}
```

### Anexo 3: Programación del Microcontrolador NodeMCU

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <SoftwareSerial.h>

#define LED2 4 //D2
#define LED 5 //D1

const char *WIFI_SSID = "RED_SILVA NARANJO";
const char *WIFI_PASSWORD = "1802729341";
const char *URL = "http://192.168.1.10/R_W/envio_datos.php";

const char *URL_1 = "http://192.168.1.10/R_W/lectura_pedido_id.php";
const char *URL_2 = "http://192.168.1.10/R_W/lectura_pedido_correo.php";
const char *URL_3 = "http://192.168.1.10/R_W/lectura_pedido_nombre.php";
const char *URL_4 = "http://192.168.1.10/R_W/lectura_pedido_grano1.php";
const char *URL_5 = "http://192.168.1.10/R_W/lectura_pedido_grano2.php";
const char *URL_6 = "http://192.168.1.10/R_W/lectura_pedido_grano3.php";
const char *URL_7 = "http://192.168.1.10/R_W/lectura_pedido_grano4.php";
const char *URL_8 = "http://192.168.1.10/R_W/lectura_pedido_retirado.php";
const char *URL_9 = "http://192.168.1.10/R_W/lectura_pedido_precio.php";
const char *URL_10 = "http://192.168.1.10/R_W/lectura_pedido_venta_total.php";
const char *URL_11 = "http://192.168.1.10/R_W/actualizar_pedido_retirado.php";

//`grano1`, `grano2`, `grano3`, `grano4`, `retirado`, `precio`, `venta_total`
WiFiClient client;
HTTPClient httpClient;

String G1="";
String G2="";
String G3="";
String G4="";
String PRE="";
String VT="";

void setup()
{
  pinMode(LED, OUTPUT);
  pinMode(LED2, OUTPUT);
  Serial.begin(9600);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  //WiFi.begin(WIFI_SSID); //cuando es abierta
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
}
```

```

Serial.println("Connected");

}

void consultas_pedidos() {
    httpClient.begin(client, URL_1); // ID
    httpClient.addHeader("Content-Type", "application/x-www-form-urlencoded");
    httpClient.POST("");
    String content1 = httpClient.getString();
    int ID = content1.toInt();
    httpClient.end();

    httpClient.begin(client, URL_2); //CORREO
    httpClient.addHeader("Content-Type", "application/x-www-form-urlencoded");
    httpClient.POST("");
    String content2 = httpClient.getString();
    String CORREO = content2;
    httpClient.end();

    httpClient.begin(client, URL_3); //NOMBRE
    httpClient.addHeader("Content-Type", "application/x-www-form-urlencoded");
    httpClient.POST("");
    String content3 = httpClient.getString();
    String NOMBRE = content3;
    httpClient.end();

    httpClient.begin(client, URL_4); //GRANO1
    httpClient.addHeader("Content-Type", "application/x-www-form-urlencoded");
    httpClient.POST("");
    String content4 = httpClient.getString();
    int GRANO1 = content4.toInt();
    httpClient.end();

    httpClient.begin(client, URL_5); //GRANO2
    httpClient.addHeader("Content-Type", "application/x-www-form-urlencoded");
    httpClient.POST("");
    String content5 = httpClient.getString();
    int GRANO2 = content5.toInt();
    httpClient.end();

    httpClient.begin(client, URL_6); //GRANO3
    httpClient.addHeader("Content-Type", "application/x-www-form-urlencoded");
    httpClient.POST("");
    String content6 = httpClient.getString();
    int GRANO3 = content6.toInt();
    httpClient.end();

    httpClient.begin(client, URL_7); //GRANO4

```

```

httpClient.addHeader("Content-Type", "application/x-www-form-urlencoded");
httpClient.POST("");
String content7 = httpClient.getString();
int GRANO4 = content7.toInt();
httpClient.end();

httpClient.begin(client, URL_8); //RETIRADO
httpClient.addHeader("Content-Type", "application/x-www-form-urlencoded");
httpClient.POST("");
String content8 = httpClient.getString();
int RETIRADO = content8.toInt();
httpClient.end();

httpClient.begin(client, URL_9); //PRECIO
httpClient.addHeader("Content-Type", "application/x-www-form-urlencoded");
httpClient.POST("");
String content9 = httpClient.getString();
double PRECIO = content9.toDouble();
httpClient.end();

httpClient.begin(client, URL_10); //VENTA
httpClient.addHeader("Content-Type", "application/x-www-form-urlencoded");
httpClient.POST("");
String content10 = httpClient.getString();
double VENTA_TOTAL = content10.toDouble();
httpClient.end();

//====TODOS LOS GRANOS CON 3 CIFRAS=====

if (((GRANO1/1)>=1) && ((GRANO1/10)<1)) { // numero de 1
cifra
    G1 = "00" + String(GRANO1);
}
else if (((GRANO1/10)>=1) && ((GRANO1/100)<1)) { // numero
de 2 cifras
    G1 = "0" + String(GRANO1);
}
else if ((GRANO1/100)>=1) { // numero de 3 cifras
    G1 = String(GRANO1);
}
else {
    G1 ="000";
}

if (((GRANO2/1)>=1) && ((GRANO2/10)<1)) { // numero de 1
cifra
    G2 = "00" + String(GRANO2);
}
}

```

```

else if (((GRANO2/10)>=1) && ((GRANO2/100)<1)) { // numero
de 2 cifras
    G2 = "0" + String(GRANO2);
}
else if ((GRANO2/100)>=1) { // numero de 3 cifras
    G2 = String(GRANO2);
}
else {
    G2 ="000";
}

if (((GRANO3/1)>=1) && ((GRANO3/10)<1)) { // numero de 1
cifra
    G3 = "00" + String(GRANO3);
}
else if (((GRANO3/10)>=1) && ((GRANO3/100)<1)) { // numero
de 2 cifras
    G3 = "0" + String(GRANO3);
}
else if ((GRANO3/100)>=1) { // numero de 3 cifras
    G3 = String(GRANO3);
}
else {
    G3 ="000";
}

if (((GRANO4/1)>=1) && ((GRANO4/10)<1)) { // numero de 1
cifra
    G4 = "00" + String(GRANO4);
}
else if (((GRANO4/10)>=1) && ((GRANO4/100)<1)) { // numero
de 2 cifras
    G4 = "0" + String(GRANO4);
}
else if ((GRANO4/100)>=1) { // numero de 3 cifras
    G4 = String(GRANO4);
}
else {
    G4 ="000";
}

//=====

//Precio con 2 ENTEROS =====
if (((PRECIO/1)>=1) && ((PRECIO/10)<1) ) { // numero de
1 cifra

```

```

    PRE = "0" + String(PRECIO);
  }
  else if (((PRECIO/10)>=1) && ((PRECIO/100)<1) ) { //
numero de 2 cifras
    PRE = String(PRECIO);
  }

  else if ((PRECIO/1)<1) { // menons a una cifra
    PRE = "0" + String(PRECIO);
  }

  //=====

  //VENTA con 3 ENTEROS =====
  if (((VENTA_TOTAL/1)>=1) && ((VENTA_TOTAL/10)<1) ) { //
numero de 1 cifra
    VT = "00" + String(VENTA_TOTAL);
  }
  else if (((VENTA_TOTAL/10)>=1) && ((VENTA_TOTAL/100)<1) )
{ // numero de 2 cifras
    VT = "0" +String(VENTA_TOTAL);
  }
  else if ((VENTA_TOTAL/100)>=1) { // numero de 3 cifras
    VT = String(VENTA_TOTAL);
  }

  else if ((VENTA_TOTAL/1)<1) { // menons a una cifra
    VT = "00" + String(VENTA_TOTAL);
  }

  //=====

  if (RETIRADO == 0) { // PEDIDO NUEVO O PEDIDO PENDIENTE

    String info_cliente_pedido = String(CORREO)+ "==" + String(NOMBRE);
    String granos_pedido = G1 + "==" + G2 + "==" + G3 + "==" + G4;
    String ventas_pedido = String(RETIRADO) + "==" + PRE + "==" + VT;

    String datos_pedido = "PEDIDONUEVO==" + granos_pedido + "==" + ventas_pedido;

    Serial.println(datos_pedido); // envio los datos al ARDUINO para dispensar
    delay(5);

    //QUEDARSE EN UN BUCLE INFINITO HASTA QUE EL ARDUINO ENVIE UNA ALERTA QUE YA
DE DISPENSO
    while (true){
      digitalWrite(LED2,HIGH);
      //Serial.flush();
      if (Serial.available() > 0){

```



```

        /*String RESPUESTA = Serial.readString();
        Serial.print (RESPUESTA);
        if (RESPUESTA=="RETIRADO"){
            digitalWrite(LED2,LOW);
            break;
        }*/
        digitalWrite(LED2,LOW);
        break;
        delay(10);
    }
}
// actualizar base en retirado AQUI
httpClient.begin(client, URL_11); //ACTUALIZAR RETIRADO
httpClient.addHeader ("Content-Type",          "application/x-www-form-
urlencoded");
httpClient.POST("");
httpClient.end();
}
else { // NO HAY PEDIDO NUEVO
    ;
}
delay(1000);
}

//=====
void lectura_datos() {
    if (Serial.available() > 0){
        //delay(10);
        String CADENA = Serial.readString();
        //Serial.println(CADENA);

        String ENCABEZADO = CADENA.substring(0,7);
        String nivel_cont_1 = CADENA.substring(9,12);
        String nivel_cont_2 = CADENA.substring(14,17);
        String nivel_cont_3 = CADENA.substring(19,22);
        String nivel_cont_4 = CADENA.substring(24,27);

        String estado_cont_1 = CADENA.substring(29,36);
        String estado_cont_2 = CADENA.substring(39,46);
        String estado_cont_3 = CADENA.substring(49,56);
        String estado_cont_4 = CADENA.substring(59,66);

        /*Serial.println(ENCABEZADO);
        Serial.println(nivel_cont_1);
        Serial.println(nivel_cont_2);
        Serial.println(nivel_cont_3);
        Serial.println(nivel_cont_4);
        Serial.println(estado_cont_1);
        Serial.println(estado_cont_2);
        Serial.println(estado_cont_3);

```

```

Serial.println(estado_cont_4);*/

if (ENCABEZADO=="CONTROL"){
    digitalWrite(LED,HIGH);
    //String data = "ped1=999&ped2=999";
    String niveles = "niv_1=" + nivel_cont_1 + "&niv_2=" + nivel_cont_2 + "&niv_3="
+ nivel_cont_3 + "&niv_4=" + nivel_cont_4;
    String estados = "est_1=" + estado_cont_1 + "&est_2=" + estado_cont_2 +
"&est_3=" + estado_cont_3 + "&est_4=" + estado_cont_4;

    String data = niveles + "&" +estados;
    httpClient.begin(client, URL);
    httpClient.addHeader("Content-Type", "application/x-www-form-urlencoded");
    httpClient.POST(data);
    String content = httpClient.getString();
    httpClient.end();
    //Serial.println(content);
    //Serial.println(data);
    delay(20);
    digitalWrite(LED,LOW);

}
else {
    digitalWrite(LED,LOW);
}
Serial.flush();

}
}
//=====
void loop()
{
    consultas_pedidos();
    lectura_datos();
}
}

```

## Anexo 4: Programación Web del Sitio de Log-In

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Login - LULA Organic</title>

  <link rel="stylesheet" href="estilos/registro.css">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">
</head>

<header>
  <h1 id="title"> Lula Organic</h1>
</header>

<body>

  <p id="description">
    Gracias por preferirnos, inicia sesión para poder hacer un pedido.
  </p>

  <form action="<?php echo htmlspecialchars($_SERVER['PHP_SELF'])?>"
method="post">
<?php
//compruebo primero si se han enviado los datos del formulario
if(isset($_POST['boton_iniciar'])) {
  //echo "<p style='color:white; font-size: 20px;'>* JAJAJAJJA</p>";
  date_default_timezone_set("America/Bogota");
  session_set_cookie_params(60*60*60*14);
  session_start(); //con este guardo las variables de registro

  //me conecto a La Base de Datos
  include('conectar.php');

  $email= strip_tags($_POST ['email']);
  $pwd_orig= (strip_tags($_POST ['pwd'])) ;
```

```

$pwd= sha1(strip_tags($_POST ['pwd'])) ; //encripta Las claves

// si hay algun espacio vacío
if ($email==NULL || $pwd==NULL) {
    echo "<p style='color:white; font-size: 20px;'>* No pueden quedar
campos vacios</p>";
    die();
}

//Primero busco si es que existe ese usuario en La Base y evito que me
inyecten codigo SQL
$sql_busq='SELECT * FROM usuarios_clientes WHERE correo = "'
.mysql_real_escape_string($conexion,$email).'" AND password = "'
.mysql_real_escape_string($conexion,$pwd)."'";
//SELECT * FROM usuarios_clientes WHERE correo =
'bpsilva1968@uta.edu.ec' AND password =
'8c31b65bdecdc9f18b695d7318186fd1feed690d'
$res_busq = mysqli_query ($conexion,$sql_busq);

//Si es que si existe:
if ($existe = @mysqli_fetch_object($res_busq)){
    $hoy = date("Y-m-d H:i:s");

    $actualizar = mysqli_query ($conexion,"UPDATE `usuarios_clientes`
SET `ultimo acceso` = '$hoy' WHERE `usuarios_clientes`.`correo` =
'$email'");

    $_SESSION ['logged']='yes'; //el usuario se confirma y paso por el
proceso de loggeo
    //con eso se evita que el ususario inicie sesion en cada pagina
nueva

    //luego extraigo todos los datos de esa persona
    $sql_extraer="SELECT * FROM usuarios_clientes WHERE correo =
'$email'";
    $res_extraer = mysqli_query ($conexion,$sql_extraer);
    $row = mysqli_fetch_array($res_extraer);

    $id = $row [0];
    $fecha = $row [1];
    $nombre = $row [2];
    $email;
    $edad = $row [5];
    $telf = $row [6];
    $tipo_user = $row [8];

    $_SESSION ['user_id']=$id;
    $_SESSION ['nombre']=$nombre;

```

```

$_SESSION ['email']=$email;
$_SESSION ['tipo']=$tipo_user;

mysqli_close($conexion);

//lo direcciono a otra pagina
echo "<p style='color:white; font-size: 20px;'>USUARIO IDENTIFICADO
EXITOSAMENTE!</p>";
    if ($_SESSION ['logged']=='yes' && $_SESSION ['tipo']=='admin')
{ // si es admin entonces abro panel
    header("Location:../UI_admins/index.php");
    }elseif ($_SESSION ['logged']=='yes' && $_SESSION
['tipo']=='cliente') { // si es cliente entonces abro pagina cliente
    header("Location:../UI_clientes/cliente.php");

    }

} else {
    $_SESSION ['logged']='no';
    echo "<p style='color:white; font-size: 20px;'>* Correo o
contraseña incorrectos</p>";
}
}
?>

<div class="input-headings">

    <label for="email" id="email-label"> Email </label>
    <input type="email" name="email" placeholder="ejemplo@ej.com"
class="form-inputs" value="<?php if(isset($email)) echo $email ?>"
required> </input>
    <br>

    <label for="pwd" id="pwd-label"> Contraseña </label>
    <input id="pwd" type="password" name="pwd" placeholder="*****"
class="form-inputs" value="<?php if(isset($pwd_orig)) echo $pwd_orig ?>"
required> </input>
    <br>

    <button id="submit" type="submit" class="submit-button"
name="boton_iniciar">Iniciar Sesión</button>

</div>

<p>
    <a href="Registro.php">REGISTRAR NUEVO USUARIO</a>
</p>

```

```
</form>

<section class="buttons">
  <a href="https://www.facebook.com/lula.organic" class="fa fa-
facebook"></a>
  <a href="#" class="fa fa-google-plus"></a>
  <a href="#" class="fa fa-youtube"></a>
  <a href="#" class="fa fa-linkedin"></a>
</section>

</body>
</html>
```

## Anexo 5: Programación Web del Sitio “Registrar usuarios”

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Registro de Usuarios</title>

  <link rel="stylesheet" href="estilos/registro.css">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">
</head>

<header>
  <h1 id="title"> Lula Organic - Registro</h1>
</header>

<body>

  <p id="description">
    Gracias por preferirnos, registra tus datos para poder hacer un
pedido.
  </p>

  <form action="<?php echo htmlspecialchars($_SERVER['PHP_SELF'])?>"
method="post">

    <div class="input-headings">

      <label for="name" id="name-label"> Nombre y Apellido: </label>
      <input type="text" name="name" placeholder="Ejm. Patricio Silva"
class="form-inputs" required> </input>
      <br>
      <label for="email" id="email-label"> Email </label>
      <input type="email" name="email" placeholder="ejemplo@ej.com"
class="form-inputs" required> </input>
      <br>
      <label for="edad" id="edad-label"> Edad </label>
```

```

        <input type="number" name="edad" min="15" max="100"
placeholder="Ejemplo 18" class="form-inputs" required> </input>
    <br>

    <label for="pwd" id="pwd-label"> Contraseña </label>
    <input id="pwd" type="password" name="pwd" placeholder="*****"
class="form-inputs" required> </input>
    <br>

    <label for="r_pwd" id="r_pwd-label"> Repita Contraseña </label>
    <input id="r_pwd" type="password" name="r_pwd"
placeholder="*****" class="form-inputs" required> </input>
    <br>

    <label for="telf" id="telf-label"> Número de Contacto </label>
    <input id="telf" type="text" name="telf" placeholder="Ej:
0998765432" class="form-inputs" required> </input>
    <br>

    <button id="submit" type="submit" class="submit-button"
name='boton_registro'>Registrarse</button>

</div>

<p>
    <a href="Login.php">log In</a>
</p>

<?php
//compruebo primero si se han enviado los datos del formulario
if(isset($_POST['boton_registro'])) {
    date_default_timezone_set("America/Bogota");
    include('conectar.php');

    $name = strip_tags($_POST ['name']); //este elimina las etiquetas html
y php que pueden ser enviadas por usuarios con segundas intenciones
    $email= strip_tags($_POST ['email']);
    $edad= strip_tags($_POST ['edad']);
    $pwd= sha1(strip_tags($_POST ['pwd'])); //encripta las claves
    $r_pwd= sha1(strip_tags($_POST ['r_pwd']));
    $telf= strip_tags($_POST ['telf']);

    $pwd_orig =strip_tags($_POST ['pwd']);
    $tamano = strlen($pwd_orig);
    $hoy = date("Y-m-d H:i:s");

    if ($tamano<8) {

```



```

    echo "<p style='color:white; font-weight: bold; font-size: 20px;'>La
contraseña debe tener al menos 8 caracteres!</p>";
    die();
}

// si hay algun espacio vacío
if ($name==NULL || $email==NULL || $edad==NULL || $pwd==NULL ||
$r_pwd==NULL ||$telf==NULL) {
    echo "<p style='color:white; font-weight: bold; font-size:
20px;'>No pueden quedar campos vacios</p>";
    die();
}

$sql_email_rep="SELECT correo FROM usuarios_clientes WHERE correo =
'$email'";
$res_email_rep = mysqli_query ($conexion,$sql_email_rep);

$row_email_rep=mysqli_fetch_array($res_email_rep);

if(mysqli_num_rows($res_email_rep)>0){ //si hay un usuario repetido

    if ($row_email_rep[0] == $email ) { //si hay un usuario repetido
        echo "<p style='color:white; font-weight: bold; font-size:
20px;'>Ya existe un usuario registrado con este correo</p>";
        die();
    }
}

elseif ($pwd != $r_pwd){
    echo "<p style='color:white; font-weight: bold; font-size:
20px;'>Las contraseñas no coinciden</p>";
    die();
}
else {
    $sql_insertar="INSERT INTO `usuarios_clientes` (`id`, `fecha`,
`nombre`, `correo`, `password`, `edad`, `telefono`,`ultimo
acceso`,`tipo`) VALUES (NULL, current_timestamp(), '$name', '$email',
'$pwd', '$edad', '$telf',current_timestamp(), 'cliente');";
    $res_insertar = mysqli_query ($conexion,$sql_insertar);

    echo "<p style='color:white; font-weight: bold; font-size:
20px;'>Registrado con éxito</p>";

}
mysqli_close($conexion);
}

```

```
?>

</form>

<section class="buttons">
  <a href="https://www.facebook.com/lula.organic" class="fa fa-
facebook"></a>
  <a href="#" class="fa fa-google-plus"></a>
  <a href="#" class="fa fa-youtube"></a>
  <a href="#" class="fa fa-linkedin"></a>
</section>

</body>
</html>
```

## Anexo 6: Programación Web de script “Log Out”

```
<?php
SESSION_START();
SESSION_DESTROY();
header("Location:Login.php");
?>
```

## **Anexo 7: Programación Web de Interfaces de Administrador y de Cliente**

**Link:**

<https://drive.google.com/drive/folders/1HyRsfL2WSZsTPBzVuTXCiKZs8TM7UXb?usp=sharing>

## Anexo 8: Especificaciones del sensor HC-SR04



Tech Support: [services@elecfreaks.com](mailto:services@elecfreaks.com)

### Ultrasonic Ranging Module HC - SR04

#### Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time x velocity of sound (340M/S) / 2,

#### Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

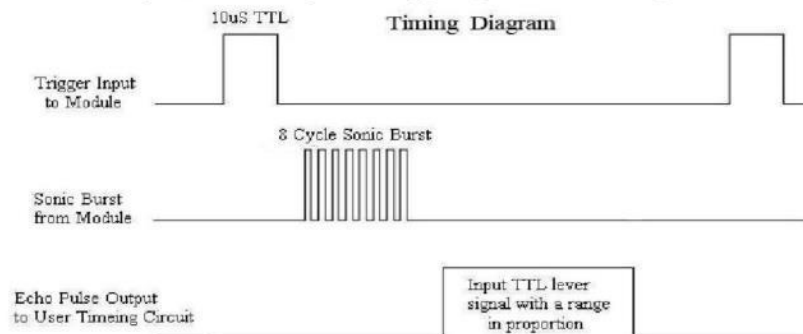
#### Electric Parameter

|                      |  |
|----------------------|--|
| Working Voltage      | DC 5 V   |
| Working Current      | 15mA   |
| Working Frequency    | 40Hz   |
| Max Range            | 4m   |
| Min Range            | 2cm  |
| MeasuringAngle       | 15 degree  |
| Trigger Input Signal | 10uS TTL pulse                                     |
| Echo Output Signal   | Input TTL lever signal and the range in proportion |
| Dimension            | 45*20*15mm   |



### Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula:  $\mu\text{S} / 58 = \text{centimeters}$  or  $\mu\text{S} / 148 = \text{inch}$ ; or: the range = high level time \* velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



---

**Attention:**

- The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.
- When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise ,it will affect the results of measuring.

[www.Elecfreaks.com](http://www.Elecfreaks.com)

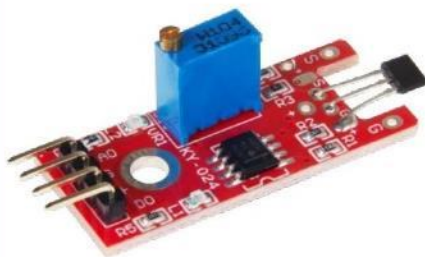


## Anexo 9: Especificaciones del sensor KY-024

JOY-IT

# LINEAR MAGNETIC HALL SENSOR

Hall sensor with analog and digital output



This magnetic Hall sensor, based on a 49E Hall sensor, outputs both the analogue sensor signal and a digital signal when a certain threshold value is reached. This threshold value can be adjusted with the potentiometer. In addition, an LED on the module lights up when the threshold value is reached.

### MAIN FEATURES

|                   |  |
|-------------------|--|
| Operating voltage | 3 V - 5 V  |
| Logic level       | 3,3 V / 5 V  |
| Chip set          | 49E, LM393   |
| Special features  | Analog and digital signal output, potentiometer to adjust the digital output |

### FURTHER SPECIFICATIONS

|                 |  |
|-----------------|--|
| Compatible with | Arduino, Raspberry Pi (with ADC), etc. |
| Colour          | Red                                    |
| Dimensions      | 45 x 15 x 14 mm                        |
| Items delivered | Hall sensor module                     |

### FURTHER DETAILS

|                    |               |
|--------------------|---------------|
| Article No.        | SEN-KY024LM   |
| EAN                | 4250236816869 |
| Customs Tariff No. | 8473302000    |

Published: 23.02.2021

www.joy-it.net  
SIMAC Electronics GmbH,  
Pascalstr. 8, 47506 Neukirchen-Vluyn



## Anexo 10: Especificaciones del servo MG-995

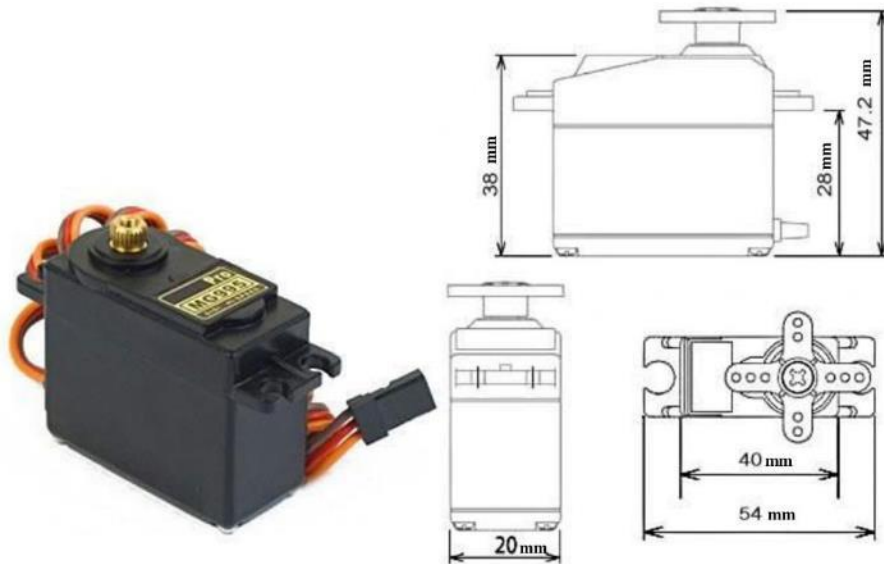
# 31150-MP MG995 High Speed Servo Actuator

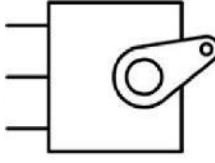
The unit comes complete with color coded 30cm wire leads with a 3 X 1 pin 0.1" Pitch type female header connector that matches most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

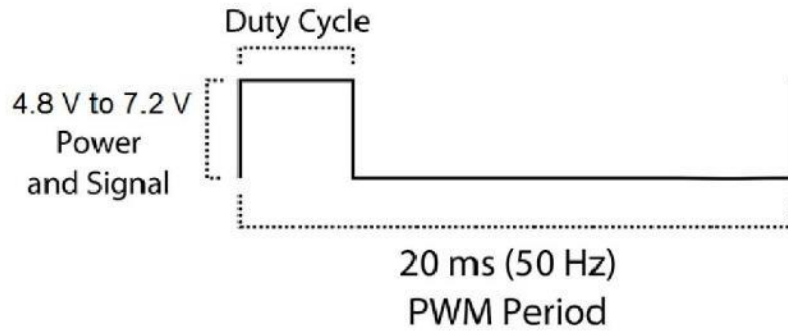
This high-speed servo actuator is not code dependant; You can use any servo code, hardware or library to control them. The MG995 Actuator includes arms and hardware to get started.

### Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 8.5 kgf·cm (4.8 V), 10 kgf·cm (6 V)
- Rotation Angle: 120deg. (+- 60 from center)
- Operating speed: 0.2 s/60° (4.8 V), 0.16 s/60° (6 V)
- Operating voltage: 4.8 V to 7.2 V
- Dead band width: 5  $\mu$ s
- Stable and shock proof double ball bearing design
- Metal Gears for longer life
- Temperature range: 0 °C – 55 °C



PWM=Orange (⏏)   
 Vcc = Red (+)  
 Ground=Brown (-)



Information obtained from or supplied by mpja.com or Marlin P. Jones and Associates inc. is supplied as a service to our customers and accuracy is not guaranteed nor is it definitive of any particular part or manufacturer. Use of information and suitability for any application is at users own discretion and userassumes all risk.



**MARLIN P. JONES & ASSOC., INC.**

P.O. Box 530400 Lake Park, Fl 33403800-652-  
 6733 FAX 561-844-8764

**WWW.MPJA.COM**

## Anexo 11: Especificaciones de la celda de Carga de 5 KG

TAL220B  
TAL220B

### MINIATURE LOAD CELL

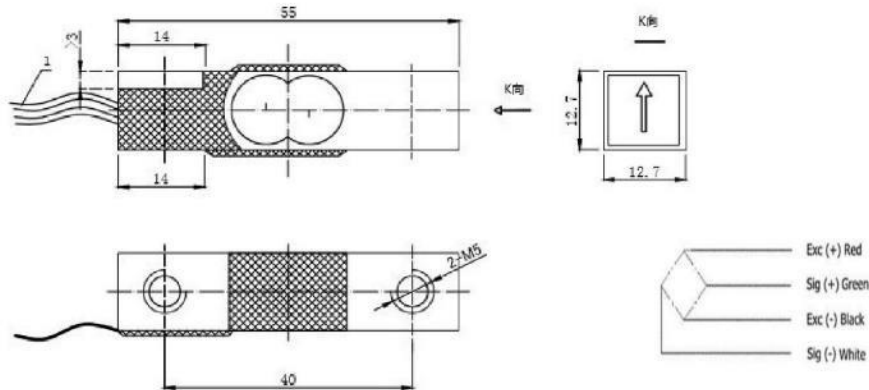


#### Features:

- ◆ Capacity : 2-50kg
- ◆ Material: aluminum-alloy
- ◆ Type: Parallel beam type
- ◆ Defend grade: IP65
- ◆ Application : hand scales, kitchen scales, mailing scales, fishing scales, baby scales and body scales and other electronic weighing devices.

WWW.HTC-SENSOR.COM

Electrical connection and Dimensions: (dimension unit: mm)



| Specifications:                 |          |   |
|---------------------------------|----------|---|
| capacity                        | kg       | 2,3,5,10,20,50kg  |
| safe overload                   | %FS      | 120   |
| ultimate overload               | %FS      | 150   |
| rated output                    | mV/V     | 1.0 ± 0.1   |
| excitation voltage              | Vdc      | 3-10  |
| combined error                  | %FS      | ± 0.05  |
| zero balance                    | %FS      | ± 0.1   |
| non-linearity                   | %FS      | ± 0.05  |
| hysteresis                      | %FS      | ± 0.05  |
| repeatability                   | %FS      | ± 0.05  |
| creep                           | %FS/3min | ± 0.1   |
| input resistance                | Ω        | 1000 ± 10   |
| output resistance               | Ω        | 1090 ± 10   |
| insulation resistance           | M Ω      | ≥ 2000  |
| operating temperature range     | °C       | -10 - +55   |
| compensated temperature range   | °C       | -10 - +40   |
| temperature coefficient of SPAN | %FS/10°C | ± 0.05  |
| temperature coefficient of ZERO | %FS/10°C | ± 0.05  |
| Electrical connection           | cable    | 4 color wire, Ø0.8 × 200 mm   |
|                                 |          | excitation(+):Red signal(+):Green excitation(-):Black signal(-):White |

※Ordering code: model-capacity- rated output-accuracy-defend grade- the length of cable

## Anexo 12: Especificaciones del controlador HX711



HX711

### 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales

#### DESCRIPTION

Based on Avia Semiconductor's patented technology, HX711 is a precision 24-bit analog-to-digital converter (ADC) designed for weigh scales and industrial control applications to interface directly with a bridge sensor.

The input multiplexer selects either Channel A or B differential input to the low-noise programmable gain amplifier (PGA). Channel A can be programmed with a gain of 128 or 64, corresponding to a full-scale differential input voltage of  $\pm 20\text{mV}$  or  $\pm 40\text{mV}$  respectively, when a 5V supply is connected to AVDD analog power supply pin. Channel B has a fixed gain of 32. On-chip power supply regulator eliminates the need for an external supply regulator to provide analog power for the ADC and the sensor. Clock input is flexible. It can be from an external clock source, a crystal, or the on-chip oscillator that does not require any external component. On-chip power-on-reset circuitry simplifies digital interface initialization.

There is no programming needed for the internal registers. All controls to the HX711 are through the pins.

#### FEATURES

- Two selectable differential input channels
- On-chip active low noise PGA with selectable gain of 32, 64 and 128
- On-chip power supply regulator for load-cell and ADC analog power supply
- On-chip oscillator requiring no external component with optional external crystal
- On-chip power-on-reset
- Simple digital control and serial interface: pin-driven controls, no programming needed
- Selectable 10SPS or 80SPS output data rate
- Simultaneous 50 and 60Hz supply rejection
- Current consumption including on-chip analog power supply regulator:
  - normal operation  $< 1.5\text{mA}$ , power down  $< 1\mu\text{A}$
- Operation supply voltage range: 2.6 ~ 5.5V
- Operation temperature range:  $-40 \sim +85^\circ\text{C}$
- 16 pin SOP-16 package

#### APPLICATIONS

- Weigh Scales
- Industrial Process Control

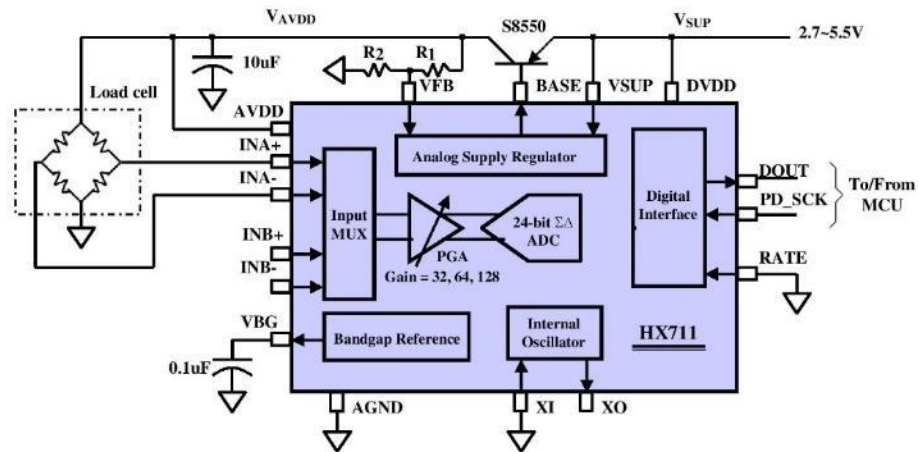
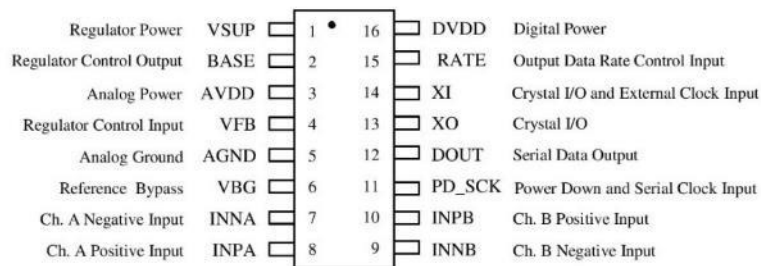


Fig. 1 Typical weigh scale application block diagram

**Pin Description**


SOP-16L Package

| Pin # | Name   | Function       | Description  |
|-------|--------|----------------|--|
| 1     | VSUP   | Power          | Regulator supply: 2.7 ~ 5.5V                                   |
| 2     | BASE   | Analog Output  | Regulator control output (NC when not used)                    |
| 3     | AVDD   | Power          | Analog supply: 2.6 ~ 5.5V                                      |
| 4     | VFB    | Analog Input   | Regulator control input (connect to AGND when not used)        |
| 5     | AGND   | Ground         | Analog Ground  |
| 6     | VBG    | Analog Output  | Reference bypass output  |
| 7     | INA-   | Analog Input   | Channel A negative input                                       |
| 8     | INA+   | Analog Input   | Channel A positive input                                       |
| 9     | INB-   | Analog Input   | Channel B negative input                                       |
| 10    | INB+   | Analog Input   | Channel B positive input                                       |
| 11    | PD_SCK | Digital Input  | Power down control (high active) and serial clock input        |
| 12    | DOUT   | Digital Output | Serial data output   |
| 13    | XO     | Digital I/O    | Crystal I/O (NC when not used)                                 |
| 14    | XI     | Digital Input  | Crystal I/O or external clock input, 0: use on-chip oscillator |
| 15    | RATE   | Digital Input  | Output data rate control, 0: 10Hz; 1: 80Hz                     |
| 16    | DVDD   | Power          | Digital supply: 2.6 ~ 5.5V                                     |

**Table 1 Pin Description**



**KEY ELECTRICAL CHARACTERISTICS**

| Parameter                                   | Notes                                  | MIN                  | TYP     | MAX      | UNIT    |
|---|--|----------------------|---------|----------|---------|
| Full scale differential input range         | V(inp)-V(inn)                          | $\pm 0.5(AVDD/GAIN)$ |         |          | V       |
| Common mode input                           |  | AGND+1.2             |         | AVDD-1.3 | V       |
| Output data rate                            | Internal Oscillator, RATE = 0          | 10                   |         |          | Hz      |
|   | Internal Oscillator, RATE = DVDD       | 80                   |         |          |         |
|   | Crystal or external clock, RATE = 0    | $f_{clk}/1,105,920$  |         |          |         |
|   | Crystal or external clock, RATE = DVDD | $f_{clk}/138,240$    |         |          |         |
| Output data coding                          | 2's complement                         | 800000               |         | 7FFFFFF  | HEX     |
| Output settling time <sup>(1)</sup>         | RATE = 0                               | 400                  |         |          | ms      |
|   | RATE = DVDD                            | 50                   |         |          |         |
| Input offset drift                          | Gain = 128                             | 0.2                  |         |          | mV      |
|   | Gain = 64                              | 0.4                  |         |          |         |
| Input noise                                 | Gain = 128, RATE = 0                   | 50                   |         |          | nV(rms) |
|   | Gain = 128, RATE = DVDD                | 90                   |         |          |         |
| Temperature drift                           | Input offset (Gain = 128)              | $\pm 6$              |         |          | nV/°C   |
|   | Gain (Gain = 128)                      | $\pm 5$              |         |          | ppm/°C  |
| Input common mode rejection                 | Gain = 128, RATE = 0                   | 100                  |         |          | dB      |
| Power supply rejection                      | Gain = 128, RATE = 0                   | 100                  |         |          | dB      |
| Reference bypass (VBG)                      |  | 1.25                 |         |          | V       |
| Crystal or external clock frequency         |  | 1                    | 11.0592 | 20       | MHz     |
| Power supply voltage                        | DVDD                                   | 2.6                  |         | 5.5      | V       |
|   | AVDD, VSUP                             | 2.6                  |         | 5.5      |         |
| Analog supply current (including regulator) | Normal                                 | 1400                 |         |          | $\mu$ A |
|   | Power down                             | 0.3                  |         |          |         |
| Digital supply current                      | Normal                                 | 100                  |         |          | $\mu$ A |
|   | Power down                             | 0.2                  |         |          |         |

(1) Settling time refers to the time from power up, reset, input channel change and gain change to valid stable output data.

**Table 2 Key Electrical Characteristics**

### Analog Inputs

Channel A differential input is designed to interface directly with a bridge sensor's differential output. It can be programmed with a gain of 128 or 64. The large gains are needed to accommodate the small output signal from the sensor. When 5V supply is used at the AVDD pin, these gains correspond to a full-scale differential input voltage of  $\pm 20\text{mV}$  or  $\pm 40\text{mV}$  respectively.

Channel B differential input has a fixed gain of 32. The full-scale input voltage range is  $\pm 80\text{mV}$ , when 5V supply is used at the AVDD pin.

### Power Supply Options

Digital power supply (DVDD) should be the same power supply as the MCU power supply.

When using internal analog supply regulator, the dropout voltage of the regulator depends on the external transistor used. The output voltage is equal to  $V_{AVDD} = V_{BG} * (R1+R2)/R2$  (Fig.1). This voltage should be designed with a minimum of 100mV below VSUP voltage.

If the on-chip analog supply regulator is not used, the VSUP pin should be connected to either AVDD or DVDD, depending on which voltage is higher. Pin VFB should be connected to Ground and pin BASE becomes NC. The external 0.1uF bypass capacitor shown on Fig. 1 at the VBG output pin is then not needed.

### Clock Source Options

By connecting pin XI to Ground, the on-chip oscillator is activated. The nominal output data rate when using the internal oscillator is 10 (RATE=0) or 80SPS (RATE=1).

If accurate output data rate is needed, crystal or external reference clock can be used. A crystal can be directly connected across XI and XO pins. An external clock can be connected to XI pin, through a 20pF ac coupled capacitor. This external clock is not required to be a square wave. It can come directly from the crystal output pin of the MCU chip, with amplitude as low as 150 mV.

When using a crystal or an external clock, the internal oscillator is automatically powered down.

### Output Data Rate and Format

When using the on-chip oscillator, output data rate is typically 10 (RATE=0) or 80SPS (RATE=1).

When using external clock or crystal, output data rate is directly proportional to the clock or crystal frequency. Using 11.0592MHz clock or crystal results in an accurate 10 (RTE=0) or 80SPS (RATE=1) output data rate.

The output 24 bits of data is in 2's complement format. When input differential signal goes out of the 24-bit range, the output data will be saturated at 800000h (MIN) or 7FFFFFFh (MAX), until the input signal comes back to the input range.

### Serial Interface

Pin PD\_SCK and DOUT are used for data retrieval, input selection, gain selection and power down controls.

When output data is not ready for retrieval, digital output pin DOUT is high. Serial clock input PD\_SCK should be low. When DOUT goes to low, it indicates data is ready for retrieval. By applying 25~27 positive clock pulses at the PD\_SCK pin, data is shifted out from the DOUT output pin. Each PD\_SCK pulse shifts out one bit, starting with the MSB bit first, until all 24 bits are shifted out. The 25<sup>th</sup> pulse at PD\_SCK input will pull DOUT pin back to high (Fig.2).

Input and gain selection are controlled by the number of the input PD\_SCK pulses (Table 3). PD\_SCK clock pulses should not be less than 25 or more than 27 within one conversion period, to avoid causing serial communication error.

| PD_SCK Pulses | Input channel | Gain |
|---------------|---------------|------|
| 25            | A             | 128  |
| 26            | B             | 32   |
| 27            | A             | 64   |

**Table 3 Input Channel and Gain Selection**

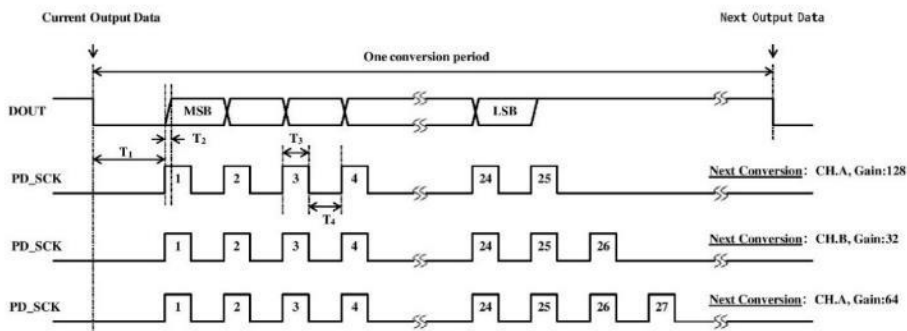


Fig.2 Data output, input and gain selection timing and control

| Symbol         | Note                                    | MIN | TYP | MAX | Unit |
|----------------|---|-----|-----|-----|------|
| T <sub>1</sub> | DOUT falling edge to PD_SCK rising edge | 0.1 |     |     | μs   |
| T <sub>2</sub> | PD_SCK rising edge to DOUT data ready   |     |     | 0.1 | μs   |
| T <sub>3</sub> | PD_SCK high time                        | 0.2 | 1   | 50  | μs   |
| T <sub>4</sub> | PD_SCK low time                         | 0.2 | 1   |     | μs   |

### Reset and Power-Down

When chip is powered up, on-chip power on rest circuitry will reset the chip.

Pin PD\_SCK input is used to power down the HX711. When PD\_SCK Input is low, chip is in normal working mode.

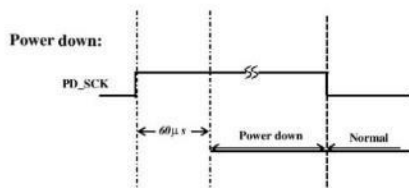


Fig.3 Power down control

When PD\_SCK pin changes from low to high and stays at high for longer than 60μs, HX711 enters power down mode (Fig.3). When internal regulator is used for HX711 and the external transducer, both HX711 and the transducer will be powered down. When PD\_SCK returns to low,

chip will return back to the setup conditions before power down and enter normal operation mode.

If PD\_SCK pulse number is changed during the current conversion period, power down should be executed after current conversion period is completed. This is to ensure that the change is saved. When chip returns back to normal operation from power down, it will return to the set up conditions of the last change.

### Application Example

Fig.1 is a typical weigh scale application using HX711. It uses on-chip oscillator (XI=0), 10Hz output data rate (RATE=0). A Single power supply (2.7~5.5V) comes directly from MCU power supply. Channel B can be used for battery level detection. The related circuitry is not shown on Fig. 1.



Reference PCB Board (Single Layer)

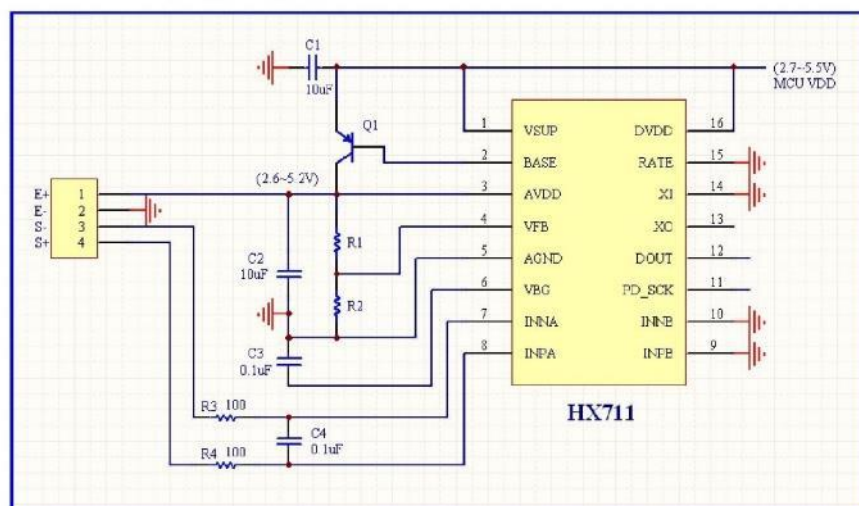


Fig.4 Reference PCB board schematic

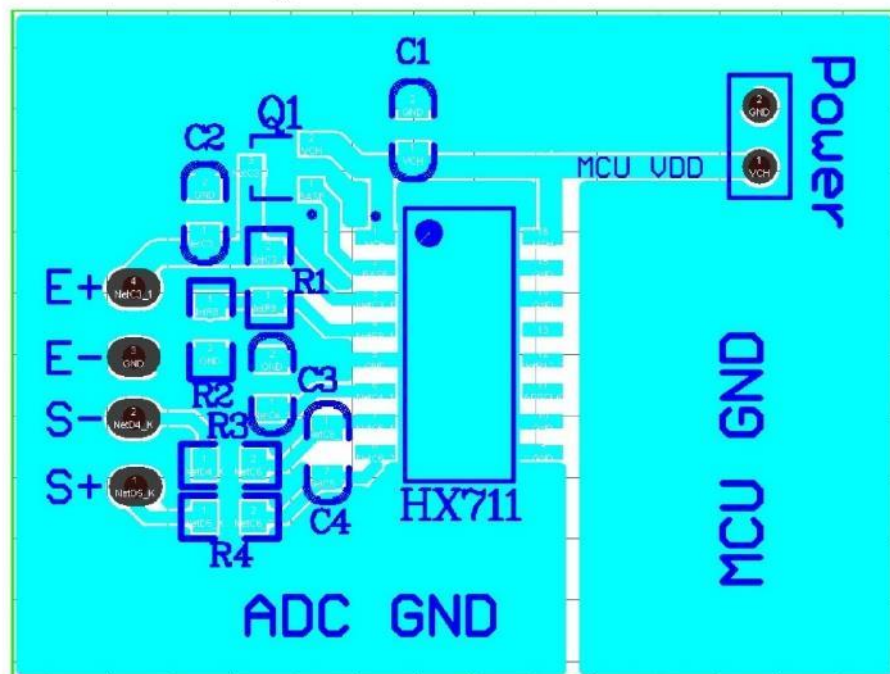


Fig.5 Reference PCB board layout

## Anexo 13: Especificaciones de la placa NodeMcu esp8266



ON THE

# INTRODUCTION TO NodeMCU ESP8266

DEVKIT v1.0

JULY 2017



[www.einstronic.com](http://www.einstronic.com)

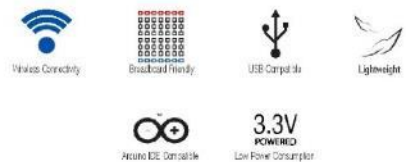
## NodeMCU ESP8266 ESP-12E WiFi Development Board

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the DevKit. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson, and spiffs.



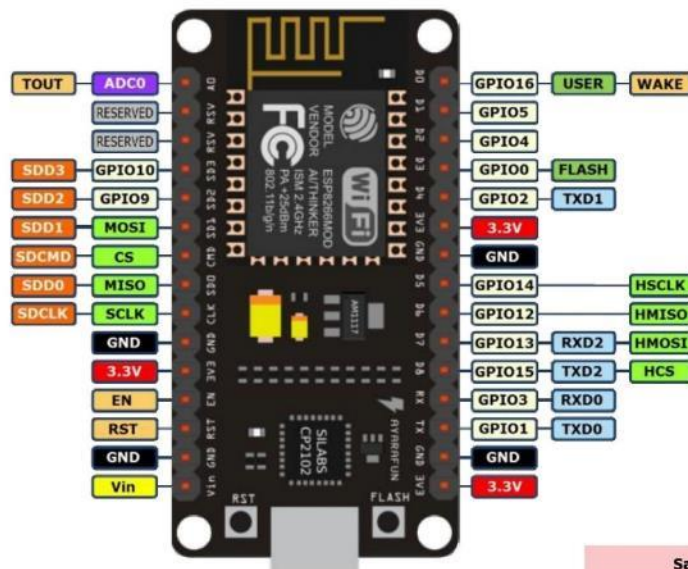
### Features

- ▶ Version : DevKit v1.0
- ▶ Breadboard Friendly
- ▶ Light Weight and small size.
- ▶ 3.3V operated, can be USB powered.
- ▶ Uses wireless protocol 802.11b/g/n.
- ▶ Built-in wireless connectivity capabilities.
- ▶ Built-in PCB antenna on the ESP-12E chip.
- ▶ Capable of PWM, I2C, SPI, UART, 1-wire, 1 analog pin.
- ▶ Uses CP2102 USB Serial Communication interface module.
- ▶ Arduino IDE compatible (extension board manager required).
- ▶ Supports Lua (alike node.js) and Arduino C programming language.



### PINOUT DIAGRAM

#### NodeMCU ESP8266 v1.0



Source  
<https://iotbytes.wordpress.com/nodemcu-pinout/>

**Safety Precaution**  
 All GPIO runs at 3.3V !!

1

## NodeMCU ESP8266



Front View



Front View

### Specifications of ESP-12E WiFi Module

|                           |  |
|---------------------------|--|
| Wireless Standard         | IEEE 802.11 b/g/n  |
| Frequency Range           | 2.412 - 2.484 GHz  |
| Power Transmission        | 802.11b : +16 ± 2 dBm (at 11 Mbps)<br>802.11g : +14 ± 2 dBm (at 54 Mbps)<br>802.11n : +13 ± 2 dBm (at HT20, MCS7)      |
| Receiving Sensitivity     | 802.11b : -93 dBm (at 11 Mbps, CCK)<br>802.11g : -85 dBm (at 54 Mbps, OFDM)<br>802.11n : -82 dBm (at HT20, MCS7)       |
| Wireless Form             | On-board PCB Antenna   |
| IO Capability             | UART, I2C, PWM, GPIO, 1 ADC  |
| Electrical Characteristic | 3.3 V Operated<br>15 mA output current per GPIO pin<br>12 - 200 mA working current<br>Less than 200 uA standby current |
| Operating Temperature     | -40 to +125 °C   |
| Serial Transmission       | 110 - 921600 bps, TCP Client 5   |
| Wireless Network Type     | STA / AP / STA + AP  |
| Security Type             | WEP / WPA-PSK / WPA2-PSK   |
| Encryption Type           | WEP64 / WEP128 / TKIP / AES  |
| Firmware Upgrade          | Local Serial Port, OTA Remote Upgrade  |
| Network Protocol          | IPv4, TCP / UDP / FTP / HTTP   |
| User Configuration        | AT + Order Set, Web Android / iOS, Smart Link APP  |

#### Disclaimer

Information provided in this document are compilation from various online resources. Elstronic Enterprise does not ensure the completeness, accuracy and reliability of the information and do not own any rights on any registered trademarks involved. Information provided should be intended for references only.



## Anexo 14: Especificaciones de la placa Arduino Mega 2560



Arduino® MEGA 2560 Rev3

Product Reference Manual

SKU: A000067



### Description

Arduino® Mega 2560 is an exemplary development board dedicated for building extensive applications as compared to other maker boards by Arduino. The board accommodates the ATmega2560 microcontroller, which operates at a frequency of 16 MHz. The board contains 54 digital input/output pins, 16 analog inputs, 4 UARTs (hardware serial ports), a USB connection, a power jack, an ICSP header, and a reset button.

### Target Areas

3D Printing, Robotics, Maker



## Features

- **ATmega2560 Processor**
  - Up to 16 MIPS Throughput at 16MHz
  - 256k bytes (of which 8k is used for the bootloader)
  - 4k bytes EEPROM
  - 8k bytes Internal SRAM
  - 32 × 8 General Purpose Working Registers
  - Real Time Counter with Separate Oscillator
  - Four 8-bit PWM Channels
  - Four Programmable Serial USART
  - Controller/Peripheral SPI Serial Interface
- **ATmega16U2**
  - Up to 16 MIPS Throughput at 16 MHz
  - 16k bytes ISP Flash Memory
  - 512 bytes EEPROM
  - 512 bytes SRAM
  - USART with SPI master only mode and hardware flow control (RTS/CTS)
  - Master/Slave SPI Serial Interface
- **Sleep Modes**
  - Idle
  - ADC Noise Reduction
  - Power-save
  - Power-down
  - Standby
  - Extended Standby
- **Power**
  - USB Connection
  - External AC/DC Adapter
- **I/O**
  - 54 Digital
  - 16 Analog
  - 15 PWM Output



## 2 Ratings

### 2.1 Recommended Operating Conditions

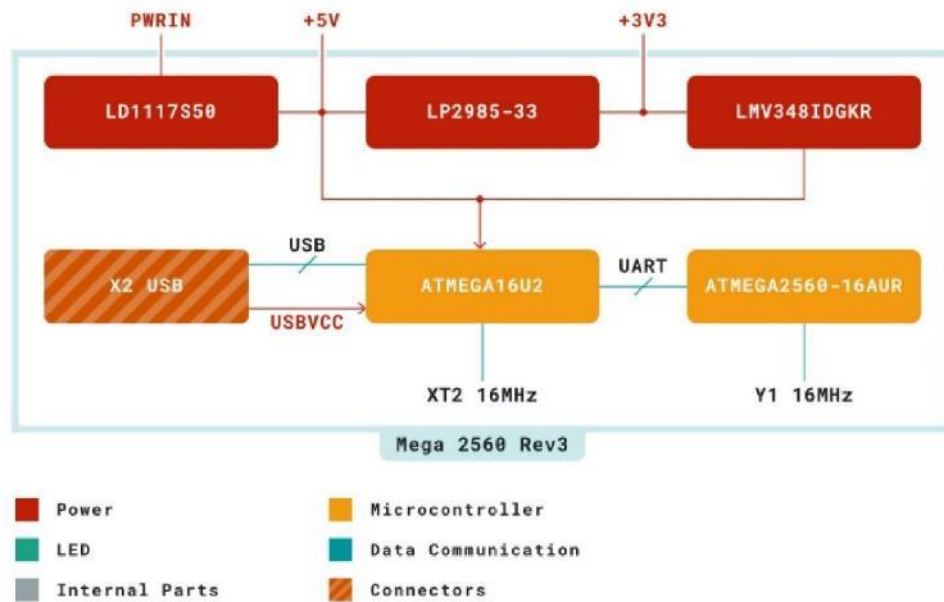
| Symbol | Description                                      | Min    | Max   |
|--------|--|--------|-------|
|        | Conservative thermal limits for the whole board: | -40 °C | 85 °C |

### 2.2 Power Consumption

| Symbol  | Description                  | Min | Typ | Max | Unit |
|---------|------------------------------|-----|-----|-----|------|
| PWRIN   | Input supply from power jack |     | TBC |     | mW   |
| USB VCC | Input supply from USB        |     | TBC |     | mW   |
| VIN     | Input from VIN pad           |     | TBC |     | mW   |

## 3 Functional Overview

### 3.1 Block Diagram

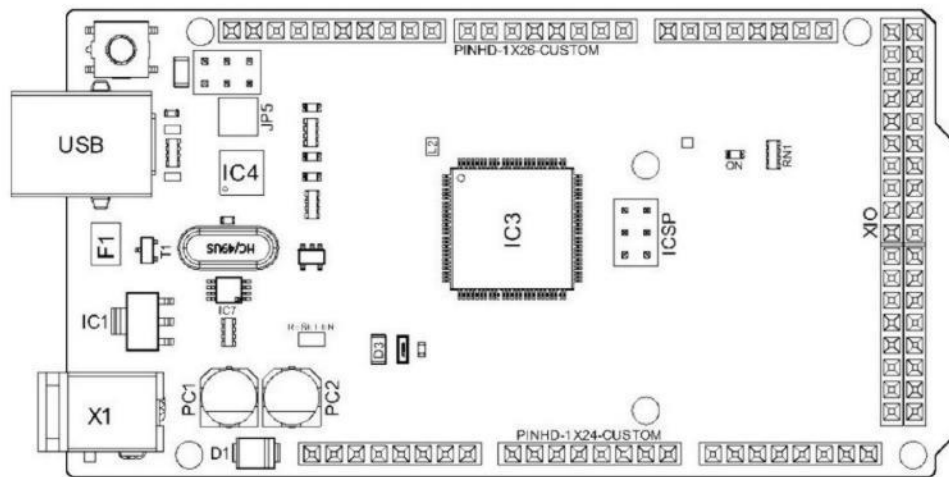


Arduino MEGA Block Diagram



3.2 Board Topology

Front View



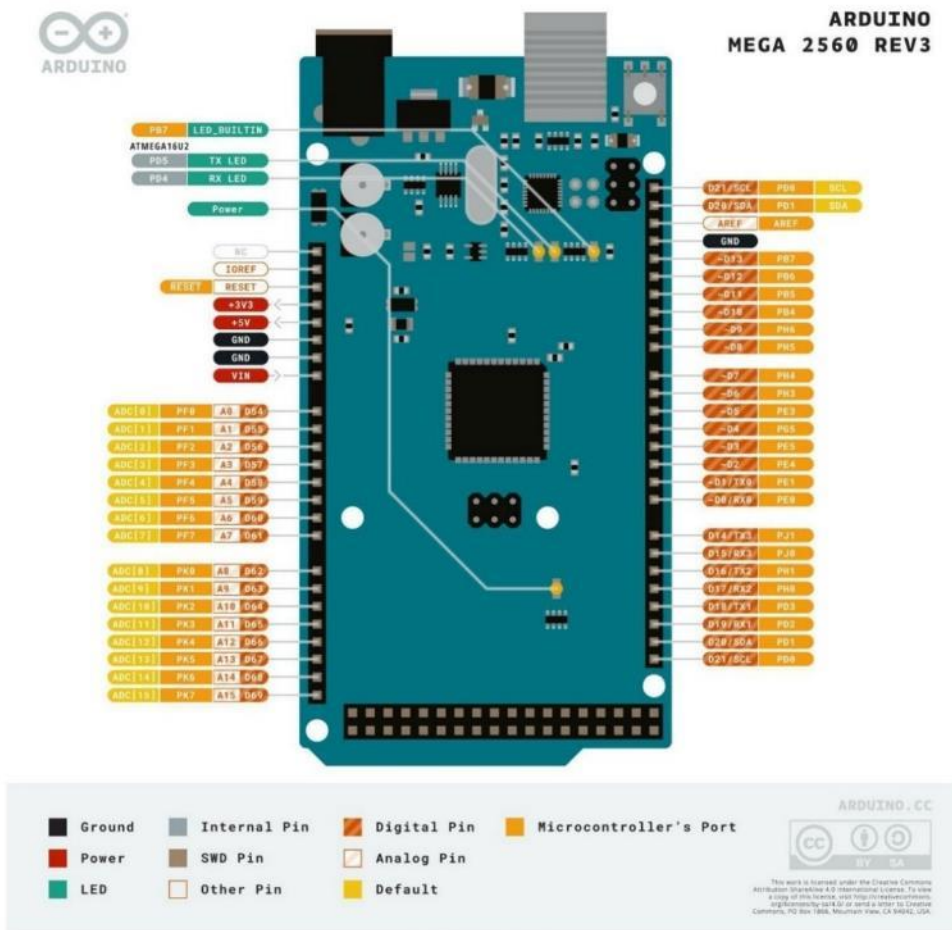
Arduino MEGA Top View

| Ref. | Description                     | Ref. | Description                     |
|------|---------------------------------|------|---------------------------------|
| USB  | USB B Connector                 | F1   | Chip Capacitor                  |
| IC1  | 5V Linear Regulator             | X1   | Power Jack Connector            |
| JP5  | Plated Holes                    | IC4  | ATmega16U2 chip                 |
| PC1  | Electrolytic Aluminum Capacitor | PC2  | Electrolytic Aluminum Capacitor |
| D1   | General Purpose Rectifier       | D3   | General Purpose Diode           |
| L2   | Fixed Inductor                  | IC3  | ATmega2560 chip                 |
| ICSP | Connector Header                | ON   | Green LED                       |
| RN1  | Resistor Array                  | XIO  | Connector                       |





### 5 Connector Pinouts





## 5.1 Analog

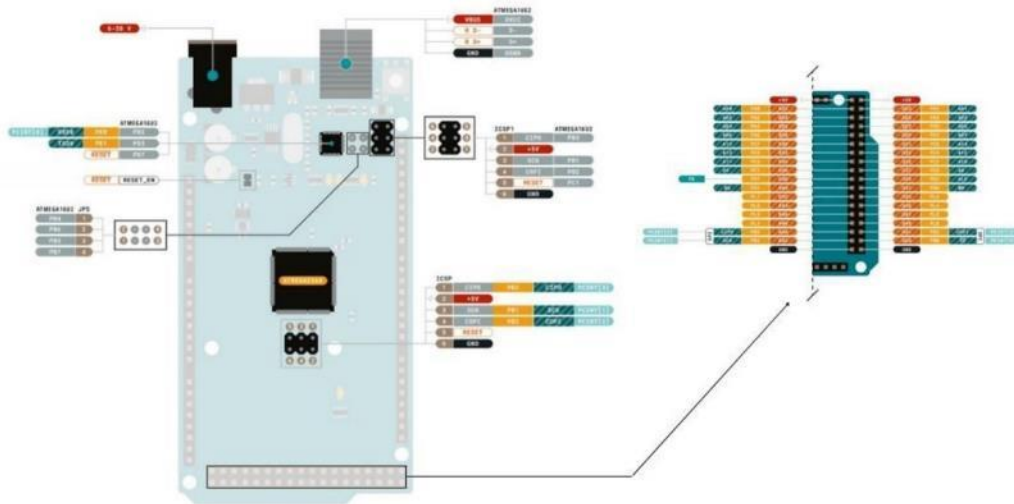
| Pin | Function | Type   | Description                                     |
|-----|----------|--------|---|
| 1   | NC       | NC     | Not Connected                                   |
| 2   | IOREF    | IOREF  | Reference for digital logic V - connected to 5V |
| 3   | Reset    | Reset  | Reset   |
| 4   | +3V3     | Power  | +3V3 Power Rail                                 |
| 5   | +5V      | Power  | +5V Power Rail                                  |
| 6   | GND      | Power  | Ground  |
| 7   | GND      | Power  | Ground  |
| 8   | VIN      | Power  | Voltage Input                                   |
| 9   | A0       | Analog | Analog input 0 /GPIO                            |
| 10  | A1       | Analog | Analog input 1 /GPIO                            |
| 11  | A2       | Analog | Analog input 2 /GPIO                            |
| 12  | A3       | Analog | Analog input 3 /GPIO                            |
| 13  | A4       | Analog | Analog input 4 /GPIO                            |
| 14  | A5       | Analog | Analog input 5 /GPIO                            |
| 15  | A6       | Analog | Analog input 6 /GPIO                            |
| 16  | A7       | Analog | Analog input 7 /GPIO                            |
| 17  | A8       | Analog | Analog input 8 /GPIO                            |
| 18  | A9       | Analog | Analog input 9 /GPIO                            |
| 19  | A10      | Analog | Analog input 10 /GPIO                           |
| 20  | A11      | Analog | Analog input 11 /GPIO                           |
| 21  | A12      | Analog | Analog input 12 /GPIO                           |
| 22  | A13      | Analog | Analog input 13 /GPIO                           |
| 23  | A14      | Analog | Analog input 14 /GPIO                           |
| 24  | A15      | Analog | Analog input 15 /GPIO                           |

## 5.2 Digital

| Pin | Function | Type              | Description                   |
|-----|----------|-------------------|-------------------------------|
| 1   | D21/SCL  | Digital Input/I2C | Digital input 21/I2C Dataline |
| 2   | D20/SDA  | Digital Input/I2C | Digital input 20/I2C Dataline |
| 3   | AREF     | Digital           | Analog Reference Voltage      |
| 4   | GND      | Power             | Ground                        |
| 5   | D13      | Digital/GPIO      | Digital input 13/GPIO         |
| 6   | D12      | Digital/GPIO      | Digital input 12/GPIO         |
| 7   | D11      | Digital/GPIO      | Digital input 11/GPIO         |
| 8   | D10      | Digital/GPIO      | Digital input 10/GPIO         |
| 9   | D9       | Digital/GPIO      | Digital input 9/GPIO          |
| 10  | D8       | Digital/GPIO      | Digital input 8/GPIO          |
| 11  | D7       | Digital/GPIO      | Digital input 7/GPIO          |
| 12  | D6       | Digital/GPIO      | Digital input 6/GPIO          |
| 13  | D5       | Digital/GPIO      | Digital input 5/GPIO          |
| 14  | D4       | Digital/GPIO      | Digital input 4/GPIO          |



| Pin | Function | Type         | Description            |
|-----|----------|--------------|------------------------|
| 15  | D3       | Digital/GPIO | Digital input 3/GPIO   |
| 16  | D2       | Digital/GPIO | Digital input 2/GPIO   |
| 17  | D1/TX0   | Digital/GPIO | Digital input 1 /GPIO  |
| 18  | D0/Tx1   | Digital/GPIO | Digital input 0 /GPIO  |
| 19  | D14      | Digital/GPIO | Digital input 14 /GPIO |
| 20  | D15      | Digital/GPIO | Digital input 15 /GPIO |
| 21  | D16      | Digital/GPIO | Digital input 16 /GPIO |
| 22  | D17      | Digital/GPIO | Digital input 17 /GPIO |
| 23  | D18      | Digital/GPIO | Digital input 18 /GPIO |
| 24  | D19      | Digital/GPIO | Digital input 19 /GPIO |
| 25  | D20      | Digital/GPIO | Digital input 20 /GPIO |
| 26  | D21      | Digital/GPIO | Digital input 21 /GPIO |



Arduino Mega Pinout



## 5.3 ATMEGA16U2 JP5

| Pin | Function | Type     | Description       |
|-----|----------|----------|-------------------|
| 1   | PB4      | Internal | Serial Wire Debug |
| 2   | PB6      | Internal | Serial Wire Debug |
| 3   | PB5      | Internal | Serial Wire Debug |
| 4   | PB7      | Internal | Serial Wire Debug |

## 5.4 ATMEGA16U2 ICSP1

| Pin | Function | Type     | Description                  |
|-----|----------|----------|------------------------------|
| 1   | CIPO     | Internal | Controller In Peripheral Out |
| 2   | +5V      | Internal | Power Supply of 5V           |
| 3   | SCK      | Internal | Serial Clock                 |
| 4   | COPI     | Internal | Controller Out Peripheral In |
| 5   | RESET    | Internal | Reset                        |
| 6   | GND      | Internal | Ground                       |

## 5.5 Digital Pins D22 - D53 LHS

| Pin | Function | Type    | Description           |
|-----|----------|---------|-----------------------|
| 1   | +5V      | Power   | Power Supply of 5V    |
| 2   | D22      | Digital | Digital input 22/GPIO |
| 3   | D24      | Digital | Digital input 24/GPIO |
| 4   | D26      | Digital | Digital input 26/GPIO |
| 5   | D28      | Digital | Digital input 28/GPIO |
| 6   | D30      | Digital | Digital input 30/GPIO |
| 7   | D32      | Digital | Digital input 32/GPIO |
| 8   | D34      | Digital | Digital input 34/GPIO |
| 9   | D36      | Digital | Digital input 36/GPIO |
| 10  | D38      | Digital | Digital input 38/GPIO |
| 11  | D40      | Digital | Digital input 40/GPIO |
| 12  | D42      | Digital | Digital input 42/GPIO |
| 13  | D44      | Digital | Digital input 44/GPIO |
| 14  | D46      | Digital | Digital input 46/GPIO |
| 15  | D48      | Digital | Digital input 48/GPIO |
| 16  | D50      | Digital | Digital input 50/GPIO |
| 17  | D52      | Digital | Digital input 52/GPIO |
| 18  | GND      | Power   | Ground                |

## Anexo 15: Especificaciones del LCD 16x2 I2C

HT

# Handson Technology

**User Guide**

## I2C Serial Interface 1602 LCD Module

This is I2C interface 16x2 LCD display module, a high-quality 2 line 16 character LCD module with on-board contrast control adjustment, backlight and I2C communication interface. For Arduino beginners, no more cumbersome and complex LCD driver circuit connection. The real significance advantages of this I2C Serial LCD module will simplify the circuit connection, save some I/O pins on Arduino board, simplified firmware development with widely available Arduino library.



**SKU:** [DSP-1182](#)

**Brief Data:**

- Compatible with Arduino Board or other controller board with I2C bus.
- Display Type: Negative white on Blue backlight.
- I2C Address: 0x38-0x3F (0x3F default)
- Supply voltage: 5V
- Interface: I2C to 4bits LCD data and control lines.
- Contrast Adjustment: built-in Potentiometer.
- Backlight Control: Firmware or jumper wire.
- Board Size: 80x36 mm.

---

1[www.handsontec.com](http://www.handsontec.com)



### Setting Up:

Hitachi's HD44780 based character LCD are very cheap and widely available, and is an essential part for any project that displays information. Using the LCD piggy-back board, desired data can be displayed on the LCD through the I2C bus. In principle, such backpacks are built around PCF8574 (from NXP) which is a general purpose bidirectional 8 bit I/O port expander that uses the I2C protocol. The PCF8574 is a silicon CMOS circuit provides general purpose remote I/O expansion (an 8-bit quasi-bidirectional) for most microcontroller families via the two-line bidirectional bus (I2C-bus). Note that most piggy-back modules are centered around PCF8574T (SO16 package of PCF8574 in DIP16 package) with a default slave address of 0x27. If your piggy-back board holds a PCF8574AT chip, then the default slave address will change to 0x3F. In short, if the piggy-back board is based on PCF8574T and the address connections (A0-A1-A2) are not bridged with solder it will have the slave address 0x27.



Address selection pads in the I2C-to-LCD piggy-back board.

Table 5. PCF8574A address map

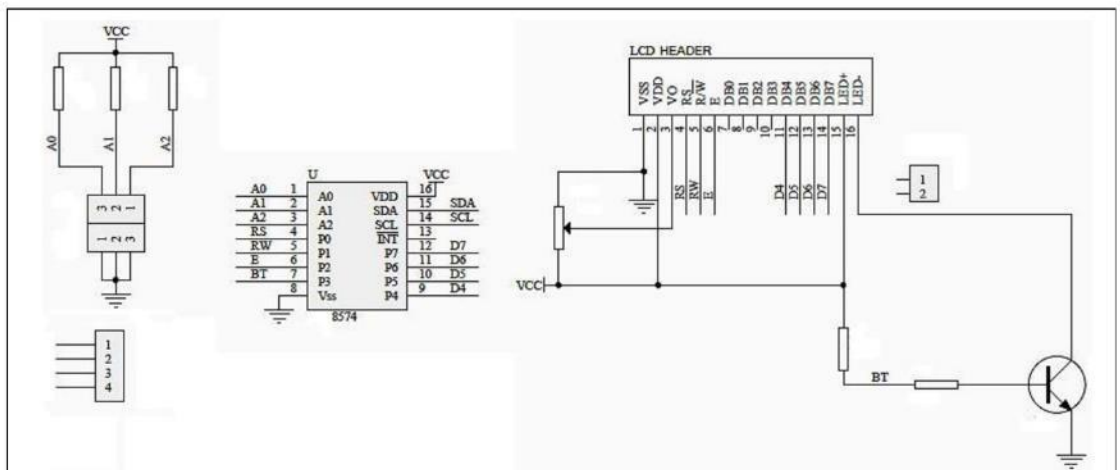
| Pin connectivity |                 |                 | Address of PCF8574A |    |    |    |    |    |    |     | Address byte value |      | 7-bit hexadecimal address without R/W |
|------------------|-----------------|-----------------|---------------------|----|----|----|----|----|----|-----|--------------------|------|---------------------------------------|
| A2               | A1              | A0              | A6                  | A5 | A4 | A3 | A2 | A1 | A0 | R/W | Write              | Read |                                       |
| V <sub>SS</sub>  | V <sub>SS</sub> | V <sub>SS</sub> | 0                   | 1  | 1  | 1  | 0  | 0  | 0  | -   | 70h                | 71h  | 38h                                   |
| V <sub>SS</sub>  | V <sub>SS</sub> | V <sub>DD</sub> | 0                   | 1  | 1  | 1  | 0  | 0  | 1  | -   | 72h                | 73h  | 39h                                   |
| V <sub>SS</sub>  | V <sub>DD</sub> | V <sub>SS</sub> | 0                   | 1  | 1  | 1  | 0  | 1  | 0  | -   | 74h                | 75h  | 3Ah                                   |
| V <sub>SS</sub>  | V <sub>DD</sub> | V <sub>DD</sub> | 0                   | 1  | 1  | 1  | 0  | 1  | 1  | -   | 76h                | 77h  | 3Bh                                   |
| V <sub>DD</sub>  | V <sub>SS</sub> | V <sub>SS</sub> | 0                   | 1  | 1  | 1  | 1  | 0  | 0  | -   | 78h                | 79h  | 3Ch                                   |
| V <sub>DD</sub>  | V <sub>SS</sub> | V <sub>DD</sub> | 0                   | 1  | 1  | 1  | 1  | 0  | 1  | -   | 7Ah                | 7Bh  | 3Dh                                   |
| V <sub>DD</sub>  | V <sub>DD</sub> | V <sub>SS</sub> | 0                   | 1  | 1  | 1  | 1  | 1  | 0  | -   | 7Ch                | 7Dh  | 3Eh                                   |
| V <sub>DD</sub>  | V <sub>DD</sub> | V <sub>DD</sub> | 0                   | 1  | 1  | 1  | 1  | 1  | 1  | -   | 7Eh                | 7Fh  | 3Fh                                   |

Address Setting of PCF8574A (extract from PCF8574A data specs).

**Note:** When the pad A0~A2 is open, the pin is pull up to VDD. When the pin is solder shorted, it is pull down to VSS.

**The default setting of this module is A0~A2 all open, so is pull up to VDD. The address is 3Fh in this case.**

Reference circuit diagram of an Arduino-compatible LCD backpack is shown below. What follows next is information on how to use one of these inexpensive backpacks to interface with a microcontroller in ways it was exactly intended.



Reference circuit diagram of the I2C-to-LCD piggy-back board.

### I2C LCD Display.

At first you need to solder the I2C-to-LCD piggy-back board to the 16-pins LCD module. Ensure that the I2C-to-LCD piggy-back board pins are straight and fit in the LCD module, then solder in the first pin while keeping the I2C-to-LCD piggy-back board in the same plane with the LCD module. Once you have finished the soldering work, get four jumper wires and connect the LCD module to your Arduino as per the instruction given below.



LCD display to Arduino wiring.

