

**UNIVERSIDAD POLITÉCNICA SALESIANA**  
**SEDE QUITO**

**CARRERA:**  
**INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:**  
**INGENIERO ELECTRÓNICO**

**TEMA:**  
**DOSIFICADOR INTELIGENTE DE BALANCEADO CON CONTROL FUZZY  
Y HERRAMIENTAS IOT PARA EL GALPÓN DE GALLINAS PONEDORAS  
ROSITA**

**AUTOR:**  
**CARLOS ALEJANDRO CLAVIJO SALAZAR**

**TUTOR:**  
**CARLOS GERMÁN PILLAJO ANGOS**

**Quito, septiembre del 2021**

## CESIÓN DE DERECHOS DE AUTOR

Yo Carlos Alejandro Clavijo Salazar con documentos de identificación No 1725508517, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación intitulado: **DOSIFICADOR INTELIGENTE DE BALANCEADO CON CONTROL FUZZY Y HERRAMIENTAS IOT PARA EL GALPÓN DE GALLINAS PONEDORAS ROSITA**, mismo que ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



Carlos Alejandro Clavijo Salazar


Cédula: 172550851-7

Quito, septiembre del 2021

## **DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR**

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, DOSIFICADOR INTELIGENTE DE BALANCEADO CON CONTROL FUZZY Y HERRAMIENTAS IOT PARA EL GALPÓN DE GALLINAS PONEDORAS ROSITA realizado por Carlos Alejandro Clavijo Salazar obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, septiembre del 2021



---

Carlos Germán Pillajo Angos MSc.

C.I: 1709255119

## **DEDICATORIA**

Este trabajo fue llevado a cabo gracias a un padre y una madre que han dado su tiempo esfuerzo y dedicación para que sus hijos lleguen más lejos. Dedico mi trabajo de titulación a mi padre amado Juan Carlos Clavijo quien toda su vida trabajo por el bienestar familiar y brindarnos una mejor calidad de vida, gracias por apoyarme en todos mis planes, a mi madre Jenny Salazar quien nos crio y estuvo siempre dándome ánimos para seguir adelante, gracias por haberme inculcado valores para ser un buen ser humano, sin su ayuda no podría haber culminado mi meta. Gracias por todo papi y mami los amo.

## **AGRADECIMIENTO**

A María Auxiliadora que de algún modo mi camino estaba escrito para ser un salesiano desde el colegio Central Técnico hasta la Universidad Politécnica Salesiana donde tuve grandes experiencias y extraordinarios docentes como mi flamante tutor Ing. Carlos Pillajo siempre estaré agradecido por su paciencia y tiempo prestado. Agradezco a todos quienes conforman la Avícola Rosita quienes me permitieron el desarrollo de este proyecto. A mis dos hermanitas que me inspiran a seguir adelante, gracias Dios por la vida que me has dado.

## ÍNDICE GENERAL

CESIÓN DE DERECHOS DE AUTOR .....	ii
DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR.....	iii
DEDICATORIA .....	iv
AGRADECIMIENTO .....	v
ÍNDICE DE TABLAS.....	xi
RESUMEN .....	vii
ABSTRACT .....	viii
INTRODUCCIÓN .....	ix
CAPITULO 1 .....	1
1. ANTECEDENTES .....	1
1.1. Planteamiento del problema .....	1
1.2. Justificación .....	2
1.3. Objetivos.....	2
1.3.1. Objetivo General.....	2
1.3.2. Objetivos Específicos .....	3
1.4. Marco Conceptual .....	3
1.4.1. Internet de las Cosas (IoT) .....	4
1.4.2. Lógica Difusa .....	5
1.4.3. Controlador difuso.....	5
1.4.4. Protocolo MQTT.....	7
1.4.5. Node-RED .....	8
1.4.6. IBM Cloud .....	9
1.4.7. Cloud Foundry .....	10
1.4.8. IBM Watson Platform IoT .....	10
1.5. Hardware utilizado.....	10
1.5.1. NodeMCU ESP8266.....	10
1.5.2. BeagleBone Black (BBB) .....	11
1.5.3. Sensor SHARP GP2Y0A51SK0F .....	11

1.5.4.	Sensor BME280.....	12
1.5.5.	Sensor AM2301A .....	13
1.5.6.	Sensor MQ135.....	13
1.5.7.	Servomotor MG995 .....	14
<b>CAPÍTULO 2 .....</b>		<b>15</b>
<b>2.</b>	<b>DISEÑO DE HARDWARE .....</b>	<b>15</b>
2.1	Arquitectura del Hardware.....	15
2.2	Descripción del galpón .....	15
2.3	Diseño estructural.....	16
2.4	Diseño de control para alimentación .....	17
2.5	Recepción de datos desde sensores ambientales .....	19
<b>CAPITULO 3 .....</b>		<b>21</b>
<b>3.</b>	<b>DISEÑO DE SOFTWARE.....</b>	<b>21</b>
3.1	Control Difuso .....	21
3.1.1.	Entradas y Salida Difusa .....	21
3.1.2.	FAM y Reglas lógicas.....	23
3.1.3.	Programa Fuzzy.....	23
3.2.	Calibración de Sensor SHARP.....	25
3.3.	Servidor local de MQTT .....	26
3.4.	MQTT en NodeMCU.....	27
3.4.1.	Algoritmo implementado en NodeMCU. ....	27
3.5.	Conexión entre Node-RED local e IBM Cloud.....	28
3.5.1.	Envío de datos desde IBM Cloud .....	29
3.5.2.	Algoritmo para envío de datos desde IBM Cloud.....	29
3.5.3.	Recepción de datos desde IBM Cloud .....	30
3.5.4.	Algoritmo para recepción de datos desde IBM Cloud .....	30
3.6.	Programación en Node-RED local.....	31
3.6.1.	Temperatura y humedad con BME280.....	31
3.6.2.	Temperatura, humedad e índice de calor con AM2301A.....	32

3.6.3.	Gases en el ambiente.....	33
3.6.4.	Envío de datos de forma manual.....	33
3.6.5.	Recepción de datos desde IBM IoT .....	34
3.6.6.	Envío de datos a IBM IoT.....	34
3.7.	Programación en Node-RED de IBM cloud .....	35
3.7.1.	Recepción de información de Temperatura, humedad y gases. ..	35
3.7.2.	Recepción de datos desde placas NodeMCU.....	36
3.7.3.	Envío de edad en semanas a placas NodeMCU .....	37
3.7.4.	Algoritmo de Node-RED en la nube.....	37
<b>CAPÍTULO 4 .....</b>		<b>39</b>
<b>4.</b>	<b>IMPLEMENTACIÓN Y PRUEBAS.....</b>	<b>39</b>
4.1	Implementación de la estructura. ....	39
4.2	Interfaz de usuario .....	39
4.3	Visualización de sensores .....	40
4.4	Implementación de controladores difusos .....	42
4.5	Implementación de red WLAN en el galpón .....	43
4.6	Prueba de control difuso. ....	43
4.7	Prueba de dosificación.....	44
<b>CONCLUSIONES .....</b>		<b>45</b>
<b>RECOMENDACIONES.....</b>		<b>46</b>
<b>BIBLIOGRAFÍA .....</b>		<b>47</b>
<b>ANEXOS .....</b>		<b>51</b>



## ÍNDICE FIGURAS

<b>Figura. 1.1</b> Proceso de lógica difusa. ....	<b>5</b>
<b>Figura. 1.2.</b> Modelo de Publicación - Suscripción MQTT.....	<b>7</b>
<b>Figura. 1.3.</b> Interfaz de desarrollo de Node-RED. ....	<b>8</b>
<b>Figura. 1.4.</b> Interfaz de IBM Cloud.....	<b>9</b>
<b>Figura. 1.5</b> Placa NodeMCU ESP8266 .....	<b>11</b>
<b>Figura. 1.6</b> Placa BeagleBone Black .....	<b>11</b>
<b>Figura. 1.7</b> Sensor SHARP .....	<b>12</b>
<b>Figura. 1.8</b> Sensor BME280.....	<b>12</b>
<b>Figura. 1.9</b> Sensor AM2301A .....	<b>13</b>
<b>Figura. 1.10</b> Sensor de gas MQ135.....	<b>14</b>
<b>Figura. 1.11</b> Servo MG995.....	<b>14</b>
<b>Figura. 2.1</b> Arquitectura del Hardware .....	<b>15</b>
<b>Figura. 2.2</b> Medidas del galpón .....	<b>16</b>
<b>Figura. 2.3</b> Tolva de dosificación.....	<b>16</b>
<b>Figura. 2.4</b> Estructura vista frontal y lateral .....	<b>17</b>
<b>Figura. 2.5.</b> Esquema de control de alimentación.....	<b>18</b>
<b>Figura. 2.6</b> Controlador difuso.....	<b>18</b>
<b>Figura. 2.7</b> Ubicación de sensores.....	<b>19</b>
<b>Figura. 2.8</b> Recepción de sensores .....	<b>20</b>
<b>Figura. 3.1.</b> Entrada Difusa 1 Distancia.....	<b>21</b>
<b>Figura. 3.2.</b> Entrada Difusa 2 Semanas.....	<b>22</b>
<b>Figura. 3.3.</b> Salida Difusa Servo.....	<b>22</b>
<b>Figura. 3.4.</b> Reglas lógicas.....	<b>23</b>
<b>Figura. 3.5.</b> Respuesta fuzzy en Arduino.ide.....	<b>24</b>
<b>Figura. 3.6.</b> Respuesta en toolbox de Matlab.....	<b>24</b>
<b>Figura. 3.7.</b> Curva de sensor Sharp.....	<b>25</b>
<b>Figura. 3.8.</b> Conexión a BeagleBone Black.....	<b>26</b>
<b>Figura. 3.9.</b> Comprobación de mosquito.....	<b>27</b>
<b>Figura. 3.10.</b> Algoritmo implementado en NodeMCU.....	<b>28</b>
<b>Figura. 3.11.</b> Dispositivo agregado en IBM Watson.....	<b>28</b>
<b>Figura. 3.12.</b> Envío de dato en formato JSON.....	<b>29</b>

<b>Figura. 3.13. Datos recibidos por IBM Watson. ....</b>	<b>29</b>
<b>Figura. 3.14. Algoritmo de envío de dato a la nube. ....</b>	<b>30</b>
<b>Figura. 3.15. Conversión de JSON a mensaje. ....</b>	<b>30</b>
<b>Figura. 3.16. Algoritmo para recepción de datos desde la nube. ....</b>	<b>31</b>
<b>Figura. 3.17 Dirección I2C del sensor BME280 en BBB. ....</b>	<b>32</b>
<b>Figura. 3.18 flujo de sensores. ....</b>	<b>32</b>
<b>Figura. 3.19 Flujo para envío de datos manual. ....</b>	<b>33</b>
<b>Figura. 3.20 Flujo de recepción de datos desde IBM Watson. ....</b>	<b>34</b>
<b>Figura. 3.21 Envío de datos a IBM IoT. ....</b>	<b>35</b>
<b>Figura. 3.22 Información subida del sensor en la nube. ....</b>	<b>35</b>
<b>Figura. 3.23 flujo de recepción de temperatura y humedad. ....</b>	<b>36</b>
<b>Figura. 3.24 Flujo de estado de placas de control. ....</b>	<b>36</b>
<b>Figura. 3.25. Envío de dato a nube. ....</b>	<b>37</b>
<b>Figura. 3.26. Algoritmo de Node-RED en la nube. ....</b>	<b>38</b>
<b>Figura. 4.1 Estructura con tolvas. ....</b>	<b>39</b>
<b>Figura. 4.2 Interfaz gráfica para el usuario. ....</b>	<b>40</b>
<b>Figura. 4.3 caja de bróker y sensores. ....</b>	<b>42</b>
<b>Figura. 4.4 Caja de controladores difusos. ....</b>	<b>42</b>
<b>Figura. 4.5 esquema de conexión WLAN. ....</b>	<b>43</b>

## ÍNDICE DE TABLAS

<b>Tabla 1.1. Forma de alimentación en granjas Avícolas. ....</b>	<b>1</b>
<b>Tabla 2.1 Índices de calor .....</b>	<b>19</b>
<b>Tabla 3.1. FAM. ....</b>	<b>23</b>
<b>Tabla 4.1 Condiciones ambientales optimas .....</b>	<b>41</b>
<b>Tabla 4.2 Prueba de dosificación.....</b>	<b>44</b>

## **RESUMEN**

La incorporación de tecnologías a una industria, sin duda mejora su productividad y promueve la actualización a nuevas tecnologías, tal es, que el IoT (Internet de las cosas) se está usando en varias industrias, pero también a nivel doméstico otorgando facilidades de operación y recopilación de datos en actuadores y sensores. El presente proyecto fue desarrollado en la Avícola Rosita en la cual se realizó una estructura móvil que consta por dos tolvas usadas como sistema dosificador de balanceado para seis hileras de jaulas, el sistema es controlado por un control fuzzy ejecutándose en la placa NodeMCU, donde ingresa información mediante protocolo MQTT a través de la placa BeagleBone Black que trabaja conjuntamente con los servicios de IBM Watson IoT y Node-RED. Para el desarrollo del control FUZZY (MISO) se utilizó un sensor SHARP con la finalidad de medir distancia al comedero siendo esta la primera entrada, como segunda entrada se ingresa la edad que tienen las gallinas, mediante un dispositivo conectado a internet y como salida se tiene un servomotor de alto torque que realiza la función de apertura y cierre a la caída de balanceado. Para la comunicación entre placas y la nube se creó un bróker local de MQTT en la placa BeagleBone Black, y por medio de la aplicación Node-RED se realizó la comunicación entre el bróker local y los servicios de IBM Watson siendo IBM cloud nuestro servidor de Node-RED en la nube.

## **ABSTRACT**

The incorporation of technologies into an industry, undoubtedly improves its productivity and promotes the update to new technologies, such is that the IoT (Internet of things) is being used in various industries, but also at the domestic level, granting operation facilities and data collection on actuators and sensors. This project was developed at Avícola Rosita in which a mobile structure was made consisting of two hoppers used as a balancing dosing system for six rows of cages, the system is controlled by a fuzzy control running on the NodeMCU board, where it enters information through the MQTT protocol through the BeagleBone Black board that works simultaneously with the IBM Watson IoT and Node-RED services. For the development of the FUZZY (MISO) control, a SHARP sensor was used in order to measure the distance to the trough, this being the first input, as the second input, the age of the hens is entered, through a device connected to the internet and as an output It has a high torque servomotor that performs the opening and closing function when the balance falls. For communication between boards and the cloud, a local MQTT broker was created on the BeagleBone Black board, and through the Node-RED application, communication was carried out between the local broker and the IBM Watson services, with IBM cloud being our server. Node-RED in the cloud.

## INTRODUCCIÓN

La ejecución del presente proyecto técnico está dividido en cuatro capítulos, en los cuales se documenta el diseño del hardware, diseño de software, conexión MQTT y nube en plataforma IBM Watson, y pruebas finales de funcionamiento y conectividad en la granja avícola Rosita. En el primer capítulo, se especifica los antecedentes del proyecto y la recopilación de información que fue necesaria para la ejecución de este proyecto además del hardware que será usado.

En el segundo capítulo, se explica los requerimientos estructurales del galpón al igual que el análisis para el desarrollo de la estructura metálica que servirá como dosificador del balanceado para una nave de seis hileras de jaulas, además del diseño de las placas PCB. (placa de circuito impreso) para el control difuso y recepción de sensores que fueron usadas en mencionado proyecto

En el tercer capítulo, se desarrolla el programa para el control fuzzy en las placas NodeMCU con sus respectivos sensores SHARP y servomotores, además se realiza la comunicación local por medio de protocolo MQTT entre las placas NodeMCU y BeagleBone Black donde esta última tiene comunicación con la plataforma IBM en la que fue incorporado Node-RED en nube y de manera local para facilitar la interacción entre dispositivos y nube. Dispositivos que enviaran a la nube información sobre los sensores y estado del sistema.

Para finalizar, en el cuarto capítulo, se implementa y pone a prueba toda la operación del sistema dosificador con herramientas IoT siendo operado mediante un dispositivo inteligente, así quedando demostrada la funcionalidad de la propuesta.

# CAPITULO 1

## 1. ANTECEDENTES

### 1.1. Planteamiento del problema

En Ecuador existían hasta el año 2015 alrededor de 310 granjas avícolas de gallinas ponedoras según MAGAP (MAGAP, 2015), entre algunas de ellas tienen una capacidad promedio de 2.000 hasta 20.000 aves, las grandes granjas productoras de huevos disponen de sistemas automatizados que facilitan y aumentan la productividad de sus empresas, los pequeños y algunos medianos avicultores trabajan de forma manual realizando un desgaste físico, condicionando su producción y limitando sus tareas diarias. En el cantón Salcedo se realizó la visita a algunas granjas avícolas que se dedican a la producción de huevos de mesa:

Tabla 1.1. Forma de alimentación en granjas Avícolas.

Nombre	Cantidad de Gallinas	Forma de alimentación.
Avícola Gavilánez	2500	Manual
Avícola Gavilánez	2600	Manual
Granja Val-S.I.	3500	Manual
Granja Val-S.I.	10000	Manual-Mecanizado
Granja Val-S.I.	6500	Manual
Avícola Marlito	6500	Manual
Avícola Marlito	6500	Manual
Avícola Marlito	8400	Manual-Mecanizado

Tabla de método de alimentación en granjas avícolas del cantón Salcedo.

La granja “Rosita” como ya se ha mencionado realizan labores de manera manual, que en el país es algo normal al igual que muchas granjas que se dedican a esta actividad. Estas trabajan con la misma metodología como se corroboró en las visitas técnicas realizadas. A inicios de la granja “Rosita” obtuvieron pérdidas por más de 35 quintales de balanceado que se traduce en dinero a 735 dólares, esto por colocar más de lo necesario teniendo en cuenta que lo hacen de forma manual, las personas que colaboran con esta actividad no viven cerca de la granja, con lo cual conociendo la problemática que se ha venido dando en dicho establecimiento surge la necesidad de

incorporar un sistema que controle la dosificación de alimento y esté al alcance desde la nube.

## **1.2. Justificación**

En la granja de gallinas ponedoras “Rosita” disponen de 5000 aves las cuales necesitan atención diaria por el resto de su estancia en dicha granja, las personas encargadas del mantenimiento y producción de la granja tienen una edad promedio de 45 a 60 años, la inclusión de la tecnología a pequeñas y medianas granjas mejoraría en gran manera su estilo de trabajo, causando un impacto económico favorable al colocar cantidades exactas de comida que las aves necesitan diariamente, tomando en cuenta que actualmente la alimentación de las aves es manual y por error humano puede haber un desperdicio de comida al colocar más de lo necesario.

Tales granjas quedan cerca o alejadas de su domicilio con lo cual la implementación IoT facilitaría en las labores diarias de sus trabajadores, permitiendo que estos se dediquen a realizar alguna otra actividad que demanda la granja, al igual que una alimentación automática y controlada permite un ahorro en su materia prima, por lo tanto, disminuye sus egresos económicos. Un sistema de alimentación automática va a reducir el esfuerzo físico efectuado por los trabajadores, también tendrá la intención de causar un ahorro en compra de sus recursos. Un control y monitoreo constante permite al propietario disponer de información ligada a la buena calidad de vida de las aves, tener un monitoreo sobre la calidad del aire, temperatura y humedad factores de gran importancia en la crianza de aves, pues estos datos ayudarán al propietario a identificar si puede haber algún tipo de problema en la crianza de las aves y posteriormente la producción de los huevos.

## **1.3. Objetivos**

### **1.3.1. Objetivo General**

Implementar un sistema de dosificación inteligente de balanceado mediante herramientas IoT y control fuzzy para una reducción del esfuerzo físico además de una alimentación más precisa.



### **1.3.2. Objetivos Específicos**

- Analizar los métodos de dosificación existentes mediante visitas técnicas y bibliografía científica pertinente para que se determinen las ventajas que tiene un sistema inteligente de dosificación.
- Construir una estructura metálica acorde con las dimensiones del lugar donde se desarrollará la parte mecánica del sistema, de tal manera que funcione para la distribución del balanceado a través de toda la granja.
- Realizar un sistema de control basado en lógica difusa para una dosificación inteligente y así causar un ahorro en el balanceado.
- Emplear la programación para IoT utilizando NODE RED para una aplicación de monitoreo y control remoto por parte de los encargados de la granja.
- Valorar el uso y aplicación de tecnología IoT y control fuzzy mediante pruebas de monitoreo y control para granjas avicultoras.

### **1.4. Marco Conceptual**

Los Pequeños avicultores representan un gran número en la industria avícola en el Ecuador, la producción de huevos de mesa en el país en el año 2019 fue de 3944 millones y hubieron 14.43 millones de gallinas ponedoras de huevos de mesa según CONAVE (CONAVE 2019). Los pollos y huevos de mesa son uno de los alimentos más consumidos en el país y esto se da por su precio accesible. Y así, criar a estas aves para la demanda masiva se ha convertido en una de las iniciativas comerciales más florecientes y tradicionales. En Ecuador siendo Pichincha y Tungurahua las provincias con mayor participación en la producción de huevos de mesa (Rosales Tapia and Segovia 2017).

Para que todo salga bien, es muy importante que la granja se encuentre en excelentes condiciones, ya que se requiere una intensidad de luz óptima y una buena higiene. La higiene y el monitoreo continuo disminuyen las enfermedades en las aves como trastornos digestivos, respiratorio y de comportamiento (Islam et al. 2019). Incorporar tecnologías en cualquier industria mejora la calidad de su producción, a los avicultores ayuda disminuir sus procesos físicos y sistematizar su gestión de producción y esto equivale a un mejor bienestar de las aves, ya que estas dependen de gran manera de factores ambientales. Tanto así que una mala condición ambiental en el galpón puede

afectar al crecimiento de las aves y la producción de huevos (Ghazal, Al-Khatib, and Chahine 2017).

El ambiente de crianza de las gallinas es fundamental y es por ello que debe existir un monitoreo constante de su ambiente para poder predecir cualquier anomalía en las aves, los factores que más influyen son: la temperatura, humedad y luminosidad. El gran avance que ha tenido el internet de las cosas a facilitado a las industrias para obtener ingeniosas soluciones económicas en varios campos, lo que solo empezó como una tecnología en la optimización de aplicaciones ahora puede ser usada para automatizar lo que antes era trabajo manual y tedioso (Nicolas et al. 2019). Tan ingeniosas como lo han implementado varias industrias con Edge Computing para el monitoreo de temperatura y humedad del gallinero con el control de intensidad de luz que se van cargando a la nube (Debauche et al. 2020), con IoT se puede monitorear en tiempo real el estado de la granja y poder controlar y modificar algunos factores que influyen en la salud de las aves. Las conexiones inalámbricas facilitan la extensión del área de aplicación de los sistemas de control automático hacia objetos móviles, pues las medidas de las variables y el control de las variables pueden ahora ser transmitidos al controlador desde casi todos los lugares de una planta.(Pillajo and Hincapie 2018).

El enfoque de una granja inteligente recalca en el monitoreo y suministro de alimentos usando apps o la web, mediante microcontroladores como Arduino o Raspberry en el caso de este proyecto una BBB, lo más difícil es el acceso a internet en las granjas, aunque estos siguen monitoreando y funcionando sin conexión a internet (Batuto et al. 2020). La automatización de una granja se logra incluyendo varios campos como la mecánica, hidráulica, eléctrica y electrónica. El sistema de agricultura automatizado es uno de los servicios más buscados en países de primer mundo dando excelentes resultados desde mayor producción y seguridad a sus trabajadores.

#### **1.4.1. Internet de las Cosas (IoT)**

La palabra internet de las cosas se ha tornado muy común dentro de los avances tecnológicos debido a que la mayoría de personas tienen acceso a internet y el siguiente paso era que nuestros dispositivos tengan accesos al mismo. IoT facilita la creación de nuevas soluciones en la optimización de procesos de producción, las nuevas enfermedades se prestan para que empresas actualicen sus servicios y producción a

algo con menos presencia humana (Pereira et al. 2020), Las bases de IoT constan de varios elementos, como son el hardware que es usado para recopilar información de su medio, software para almacenar, analizar y procesar datos, y de una interfaz donde los usuarios puedan interactuar con la plataforma IoT.(Astill et al. 2020)

### 1.4.2. Lógica Difusa

La lógica difusa trata de resolver problemas ambiguos donde la lógica binaria o tradicional no puede solucionarlo, problemas de la vida real como toma de decisiones en la que el canal de balanceado esta poco lleno o poco vacío, es inevitable usar lógica difusa cuando la información es tan ambigua o imprecisa. Desde el punto de vista humano, es necesario contar con herramientas para tratar la información ambigua y desde la perspectiva informática. Los sistemas informáticos requieren definiciones precisas para actuar, aunque los humanos disponen de una capacidad increíble para trabajar con definiciones vagas, imprecisas y lingüísticas como se ve en el conocimiento del mundo real.(Omomule, Ajayi, and Orogun 2020)

### 1.4.3. Controlador difuso

El controlador difuso es similar a otros controladores que toman la señal de uno o varios sensores de entrada, las procesan y a la salida responden los actuadores con el fin de regular algún proceso de la planta (GONZALEZ CRUZ 2012), la figura 1.1 demuestra el proceso de lógica difusa.

Figura. 1.1 Proceso de lógica difusa.

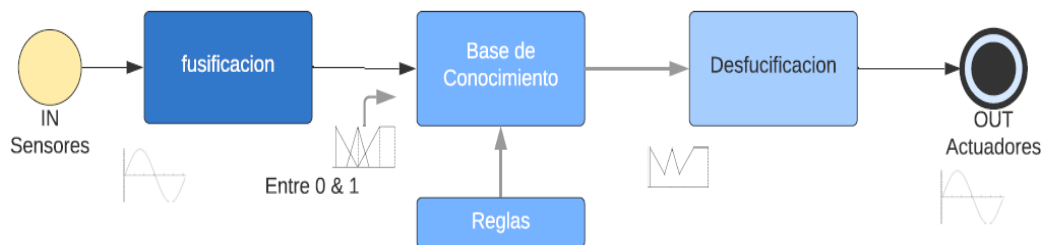


Diagrama representativo de un control difuso. Elaborado por Clavijo Carlos.

#### **1.4.3.1. Fusificación**

Consiste en medir los valores de las entradas difusas de los sensores para realizar un mapeo a escala que transforma los datos de entradas en variables lingüísticas que son las etiquetas de las funciones de pertenencia o conjuntos difusos. En cualquier aplicación práctica en industrias donde se realiza medición de voltaje, corriente, temperatura, etc., puede haber un error insignificante. Esto provoca imprecisión en los datos. Esta imprecisión puede estar representada por las funciones de pertenencia. Así se realiza la fusificación. Por lo tanto, el proceso de fusificación puede implicar la asignación de valores de pertenencia para las cantidades ambiguas (Hooda and Raich 2016). La función de pertenencia puede ser triangular, trapezoidal, campana de Gauss, o sigmoidea.

#### **1.4.3.2. Base de Conocimiento**

La lógica difusa usa reglas lógicas como **IF antecedente - THEN consecuente** para ejecutar operaciones semánticas que permiten realizar la inferencia para encontrar una situación y tener la capacidad de resolverla, la base de conocimiento contiene toda la información del sistema que se va a controlar, esta proporciona las definiciones para el establecimiento de reglas y la manipulación de datos difusos. La base de reglas caracteriza las claves en el control y la finalidad con que usan los expertos para llevar a cabo el control, manejando proposiciones (MEZA TTITO 2019).

#### **1.4.3.3. Defusificación**

La Defusificación es un proceso matemático el cual se usa para convertir un conjunto difuso en un número real, su operación es simple donde corta la función de membresía al grado de la membresía respectiva, en otras palabras, segmenta las funciones de membresía generando en cada función dos áreas. El área inferior es la que se toma para realizar el cálculo, se superponen todas estas áreas y se saca el centroide de las áreas superpuestas, la cual indica la salida real del sistema (Ponce 2010). Es decir, la defusificación consiste en transformar las señales analógicas de la señal difusa en señales que el actuador pueda entender.

#### 1.4.4. Protocolo MQTT

El modelo de publicación-suscripción que utiliza MQTT separa los puntos finales e incluye el componente central que es utilizado como nodo intermedio. Hay dos tipos de clientes: publicadores y suscriptores. En principio, es posible que no estén enlazados entre sí. Los publicadores envían mensajes al bróker que reenvía el mensaje a los suscriptores. Tenga en cuenta que el cliente MQTT puede ser cualquier dispositivo que esté conectado al broker.(Mileva et al. 2021), la figura 1.2 hace referencia al protocolo MQTT usado en este proyecto.

Figura. 1.2. Modelo de Publicación - Suscripción MQTT.

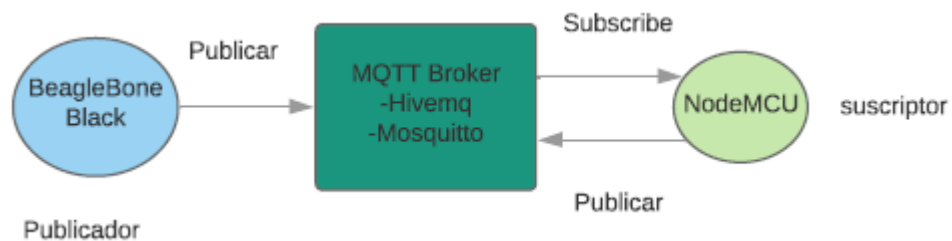


Diagrama representativo de protocolo MQTT. Elaborado por Clavijo Carlos.

MQTT es el punto central o simplemente el servidor de IoT que controla la comunicación entre el publicador y suscriptor(Balaji et al. 2020), un ejemplo sencillo de como viajan los datos es cuando un avicultor recibe en su teléfono celular información sobre la temperatura actual de su galpón, siendo el sensor de temperatura el publicador debido a que envía información al bróker mientras que el teléfono viene siendo el suscriptor donde llega la información enviada por el sensor de temperatura.

El protocolo MQTT se basa en el modelo de transmisión que facilita la conexión para una transferencia de datos eficiente y también una comunicación fiable. Este protocolo se utiliza para la recopilación de datos sin ningún retraso en la infraestructura. La nube es el almacenamiento bajo demanda para almacenar, procesar y acceder a datos en Internet, MQTT tiene 3 niveles de calidad (QoS).

- 0 = como máximo una vez: El publicador envía un mensaje al bróker una única vez en su comunicación de transferencia de datos. El nivel no es el óptimo para

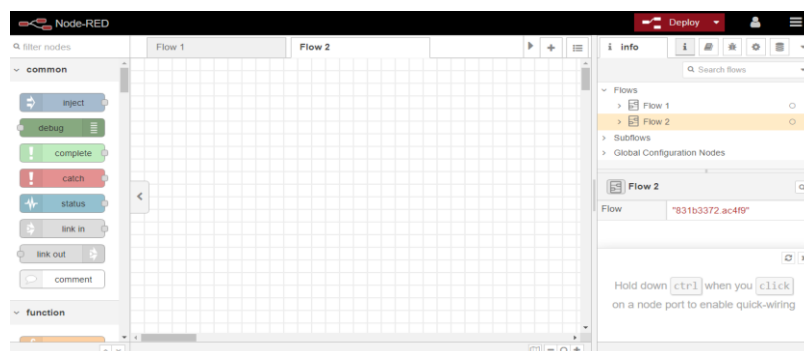
la información de misión y el suscriptor no asegurará su mensaje en la comunicación.

- 1= Al menos una vez: El publicador envía el mensaje al bróker hasta que se garantiza la entrega. El mensaje de confirmación se recibe solo una vez en la comunicación, por lo que es mejor para las métricas de rendimiento.
- 2= Exactamente una vez: El bróker ejecutará el mensaje exactamente una vez
- como el publicador trabaja en conjunto en la arquitectura. Dará la mejor métrica de rendimiento en términos de ancho de banda y latencia, por lo que es un método muy lento, cuando realmente lo vea necesario utilice este protocolo de comunicación (Arivubrakan and Prema 2020).

#### 1.4.5. Node-RED

Node-RED es una herramienta de programación basada en flujo, desarrollada originalmente por el equipo de Servicios de tecnología emergente de IBM y ahora forma parte de JS Foundation. (Node-RED 2021), el entorno de trabajo de Node-RED se basa en el flujo como se indica en la figura 1.3, donde los usuarios escriben el programa a través de combinación de nodos. Estos nodos pueden ser equipos de hardware, aplicaciones web o servicios en la nube, lo interesante de Node-RED es su tablero grafico que se puede usar para controlar dispositivos y monitorear el estado de varios dispositivos dentro del mismo dominio (Lu et al. 2020).

Figura. 1.3. Interfaz de desarrollo de Node-RED.



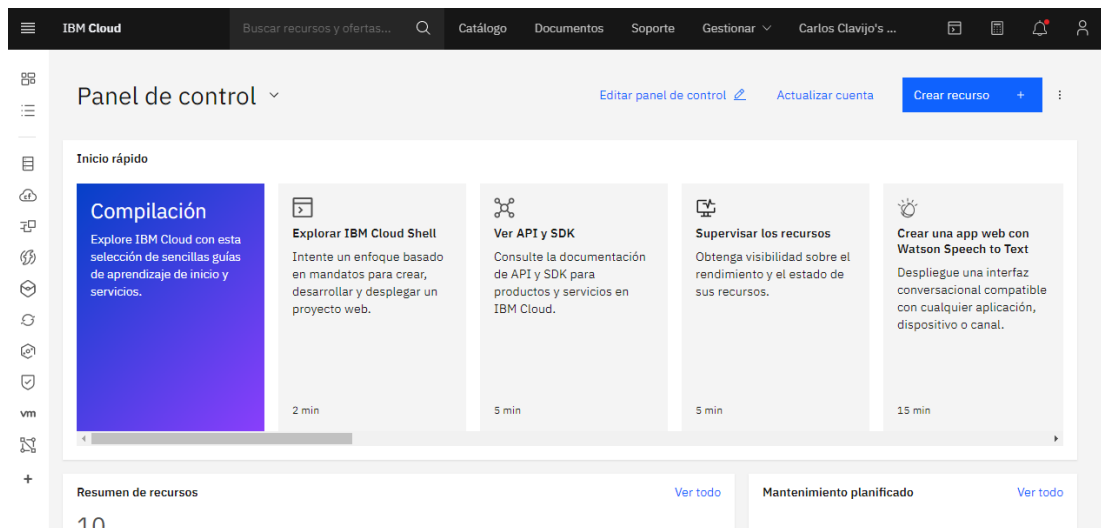
Interfaz de usuario de plataforma Node-RED. Elaborado por Clavijo Carlos.

La programación es de manera visual facilitando las tareas para nuevos proyectos, facilitando el aprendizaje con la posibilidad de conectar flujos con cualquier otro lenguaje de programación siendo más sencillo para el usuario, una de sus principales ventajas es que realiza tareas que pueden ser complejas algo que con otros lenguajes llevaría muchas líneas de código, en Node-RED los nodos son subrutinas específicas que tienen entradas y salidas donde pueden recibir mensajes por MQTT o encender un led por activación de un pulsador entre otras cosas(Gómez Giraldo 2021).

#### 1.4.6. IBM Cloud

La plataforma IBM Cloud que se indica en la figura 1.4 combina plataforma como servicio (PaaS) con infraestructura como servicio (IaaS) para proporcionar una experiencia integrada. La plataforma se escala y admite tanto a pequeños equipos y organizaciones de desarrollo como a grandes empresas. Implementada globalmente en centros de datos de todo el mundo, la solución que crea en IBM Cloud se pone en marcha rápidamente y funciona de forma fiable en un entorno probado y compatible (IBM Cloud 2021a).

Figura. 1.4. Interfaz de IBM Cloud.



Panel de control en IBM cloud. Elaborado por Clavijo Carlos.

### **1.4.7. Cloud Foundry**

Cloud Foundry es una plataforma de aplicaciones en la nube de código abierto que otorga una variedad de nubes, marcos para desarrolladores y servicios de aplicaciones. Cloud Foundry hace que sea más rápido y fácil construir, probar, implementar y escalar aplicaciones. Es un proyecto de código abierto y está disponible a través de una variedad de distribuciones de nube privada e instancias de nube pública. está diseñado para ser configurado, implementado, administrado, escalado y actualizado en cualquier proveedor de IaaS en la nube (Cloud Foundry Foundation 2021).

### **1.4.8. IBM Watson Platform IoT**

IBM Watson IoT Platform es un servicio totalmente gestionado y alojado en la nube que facilita la derivación de valor de los dispositivos de Internet de las cosas. La solución IBM Watson IoT Platform proporciona un conjunto de servicios de IBM Cloud como una única oferta de SaaS administrada por IBM que le permite recopilar y analizar el rendimiento del producto (IBM Cloud 2021).

Los dispositivos son asignados a organizaciones, los cuales definen el dominio de la información de estos dispositivos, cada organización representa una instancia de la plataforma, existen dos tipos de dispositivos los gestionados y no gestionados. Los gestionados conceden al dispositivo conectarse al Device Management de la plataforma mediante el protocolo Device Management. De esta forma, este dispositivo puede ser gestionado de forma remota, los no gestionados pueden conectarse a la plataforma por otros protocolos como MQTT o HTTPS, pero no pueden gestionar (MARTINEZ 2017).

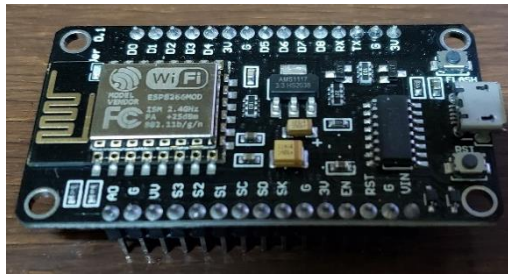
## **1.5. Hardware utilizado**

### **1.5.1. NodeMCU ESP8266**

Para este proyecto fue escogido el módulo NodeMCU esp8266 por su facilidad de conexión con el protocolo TCP-IP, permitiendo la comunicación sin cables a través de un router WiFi, además es muy accesible por su bajo coste y tamaño, contiene un pin analógico y nueve pines digitales con I2C. y SPI, además cuenta con 4MB de memoria FLASH (Muñoz et al. 2019), la figura 1.5 indica la placa mencionada.



Figura. 1.5 Placa NodeMCU ESP8266

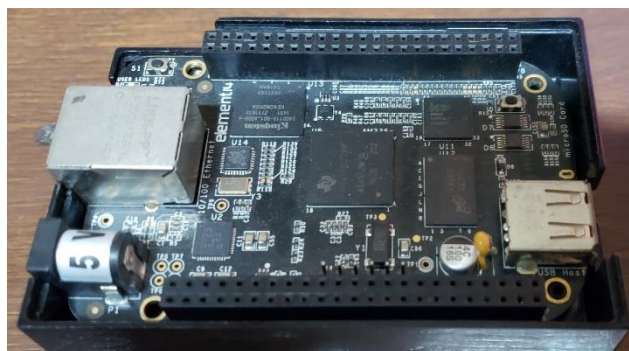


Controlador seleccionado. Elaborado por Clavijo Carlos.

### 1.5.2. BeagleBone Black (BBB)

Fue escogida como un medio de adquisición y procesamiento de datos para este trabajo puesto que es idóneo para proyectos que requieren software de alto nivel y una aplicación electrónica, de esta manera se puede aplicar a proyectos que requieran gestiones automatizadas ya sea en el hogar, pantallas inteligente, red de sensores, cualquier trabajo conectado a internet interactivo, esta placa tiene un gran poder de procesamiento para realizar todo aquello que se pueda imaginar de un dispositivo integrado (ver Figura 1.6) (Molloy 2014).

Figura. 1.6 Placa BeagleBone Black



Placa de desarrollo seleccionada. Elaborado por Clavijo Carlos.

### 1.5.3. Sensor SHARP GP2Y0A51SK0F

El sensor de distancia SHARP que mide objetos de 2 a 10 cm fue escogido debido a que la temperatura ambiental y la duración de uso no afectan a la detección de objetos a distancia debido a la adopción del método de triangulación, este compuesto por una

combinación integrada PSD (detector sensible a la posición), un diodo emisor de infrarrojos y un circuito de procesamiento de señales (ver Figura 1.7) (Sharp 2002).

Figura. 1.7 Sensor SHARP



Sensor de proximidad Pololu de 2 a 10 cm. Elaborado por Clavijo Carlos.

#### 1.5.4. Sensor BME280

El BME280 es un sensor de baja potencia que consume solo  $3,6 \mu\text{A}$  en funcionamiento y  $0,1 \mu\text{A}$  en modo de reposo. Tiene una precisión de  $\pm 0.5 \text{ }^\circ\text{C}$  para temperatura,  $\pm 3\%$  para humedad relativa (Wu et al. 2019), integra en un solo dispositivo sensores de presión atmosférica, temperatura y humedad relativa con gran precisión, bajo consumo energético, para este proyecto solo se usará la de temperatura y humedad relativa. En cuando al sensor de humedad relativa presenta un desempeño sobresaliente comparado a los sensores DHT11. Se conecta directamente a un microcontrolador a través de I2C. o SPI (ver Figura 1.8).

Figura. 1.8 Sensor BME280



Sensor de temperatura y humedad por I2C. Elaborado por Clavijo Carlos.

### 1.5.5. Sensor AM2301A

El sensor de temperatura y humedad AM2301A que se muestra en la figura 1.9 será usado dentro del galpón para tener otra referencia con respecto a las condiciones internas del galpón, su rango de medición es de  $-40$  a  $80^{\circ}$  C para la temperatura con una precisión de  $\pm 0.3^{\circ}$ C, para la humedad relativa es de 0 a 100% de HR con una precisión de  $\pm 3\%$  (Chen et al. 2017). Este sensor es un producto integrado con bus digital de un solo cable de datos y tiene confiabilidad industrial, El bit de comprobación se aplica en la inspección final de una lectura, donde se puede evitar el error de lectura, debido a que AM2301A actúa como un representante típico del dispositivo basado en un patrón de comunicación de bus de un cable, otra propiedad técnica de AM2301A minimiza su impacto en el sistema de trabajo (Zhao et al. 2018).

Figura. 1.9 Sensor AM2301A



Sensor de temperatura y humedad digital. Elaborado por Clavijo Carlos.

### 1.5.6. Sensor MQ135

Este sensor permite identificar cierto tipo de gas que se encuentra en el ambiente, el mismo posee un micro tubo de cerámica y un calentador, este último proporciona las condiciones necesarias para activar el sensor, aunque no nos indicara directamente las partículas por millón del gas nos devolverá una cantidad proporcional a la resistencia que produce cierto gas (Pinzón et al. 2018), es usado en equipos de control de calidad del aire para edificios, oficinas, etc. Detectan  $\text{NH}_3$ ,  $\text{NO}_x$ , alcohol, benceno, humo,  $\text{CO}_2$ , se muestra el sensor en la figura 1.10.

Figura. 1.10 Sensor de gas MQ135



Sensor de monóxido de carbono y amonio. Elaborado por Clavijo Carlos.

### 1.5.7. Servomotor MG995

Este servo fue escogido por su pequeño tamaño y alta eficiencia en el torque ya que a 6 V proporciona un torque de 10 kgf\*cm, torque suficiente para el trabajo que se va a realizar (ver Figura 1.11).

Figura. 1.11 Servo MG995



Servomotor de alto torque y respuesta rápida. Elaborado por Clavijo Carlos.

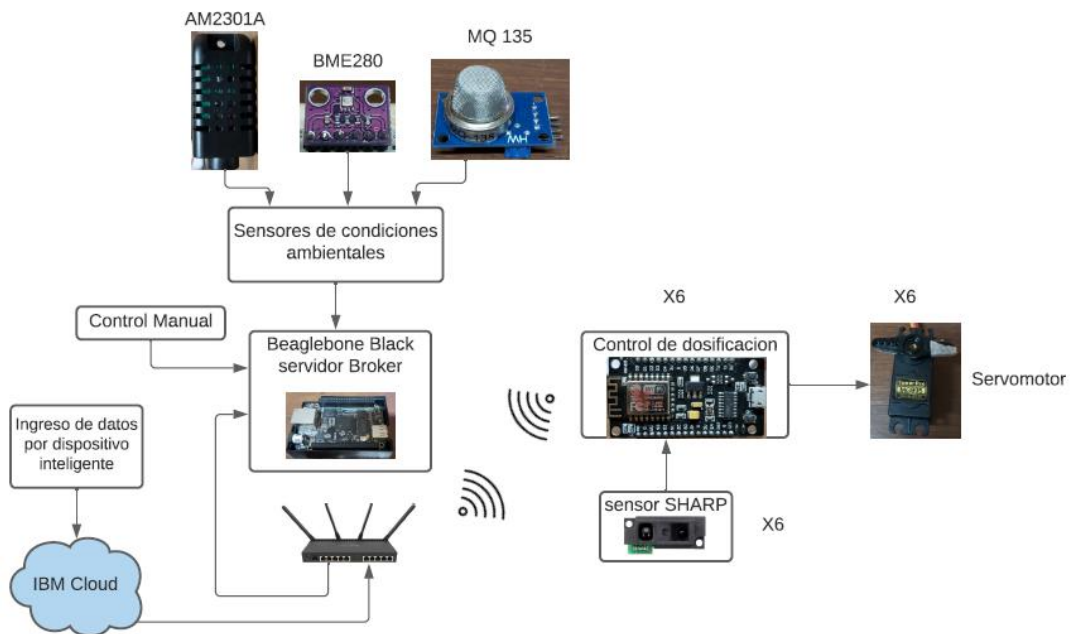
## CAPÍTULO 2

### 2. DISEÑO DE HARDWARE

#### 2.1 Arquitectura del Hardware

El sistema propuesto se describe brevemente en esta sección, como se representa en la figura 2.1, donde los sensores ambientales envían su información a una placa NodeMCU la cual envía los datos recopilados por MQTT al servidor bróker local. Mediante un dispositivo inteligente se ingresa los datos de semana que serán usados por las placas de control difuso, el servidor bróker se encarga de todo lo que sucede con el sistema.

Figura. 2.1 Arquitectura del Hardware

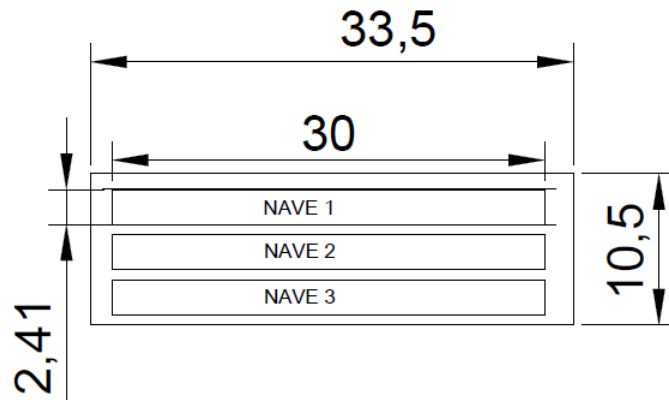


Esquema del hardware incorporado. Elaborado por Clavijo Carlos.

#### 2.2 Descripción del galpón

La granja Avícola rosita tiene una capacidad instalada de 5000 gallinas, existen tres naves (ver Figura 2.2) de seis filas cada una (tres cada lado), se desarrolló un dosificador inteligente para la nave uno donde se colocaron rieles para que la estructura sea pueda movilizar a lo largo del galpón.

Figura. 2.2 Medidas del galpón



Diseño estructural del galpón avícola Rosita con unidades representadas en metros. Elaborado por Clavijo Carlos.

### 2.3 Diseño estructural

Por el poco espacio entre naves se propuso una estructura móvil con dos tolvas colocadas en la parte superior, cada una consta con la capacidad de 90 Kg de balanceado (ver figura 2.3 y 2.4), tres tubos de acero de 3 pulgadas sirven como conducto hacia cada canal de cada fila, la cantidad de balanceado que será colocada ira disminuyendo por cada tubo dependiendo del balanceado que haya en cada canal, dicha estructura se movilizará a lo largo del galpón donde cada tolva dosificará a tres filas, logrando así optimizar recursos que se necesitaron para la creación de dicha estructura.

Figura. 2.3 Tolva de dosificación

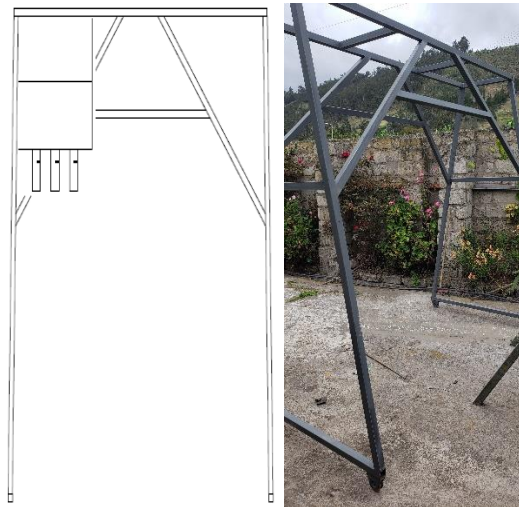


Diseño de tolvas dosificadoras. Elaborado por Clavijo Carlos.

Las tolvas dosificadoras tienen soportes para poder colocar los servomotores en los que fueron agregadas tol en forma de circular que realizan la función de compuerta de apertura y cierre al flujo de balanceado.

La estructura móvil fue diseñada de tal manera que contenga las dos tolvas en la parte superior una en cada extremo, además de ser el soporte para las tolvas, contendrá la caja de control difuso en la cual está la batería y su respectiva placa PCB con la conexión de servomotores y sensores. Para la movilización de dicha estructura se instalaron ruedas para riel en V en el piso del galpón.

Figura. 2.4 Estructura vista frontal y lateral



Diseño de estructura móvil. Elaborado por Clavijo Carlos.

## 2.4 Diseño de control para alimentación

La intención de este proyecto fue optimizar la manera en la que se alimentaba a las aves. En las tolvas se colocaron seis servomotores y seis sensores SHARP infrarrojos, uno por cada fila, donde la función de los sensores es medir la distancia que en ese momento hay en el canal de alimentación con la referencia, para así colocar menos cantidad, esa información es enviada a una placa NodeMCU la cual efectúa el control difuso controlando la apertura del servo motor. El esquema del proceso efectuado se representa en la figura 2.5.

Figura. 2.5. Esquema de control de alimentación.

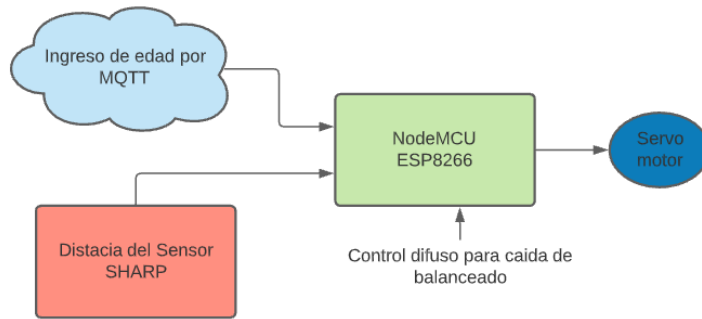
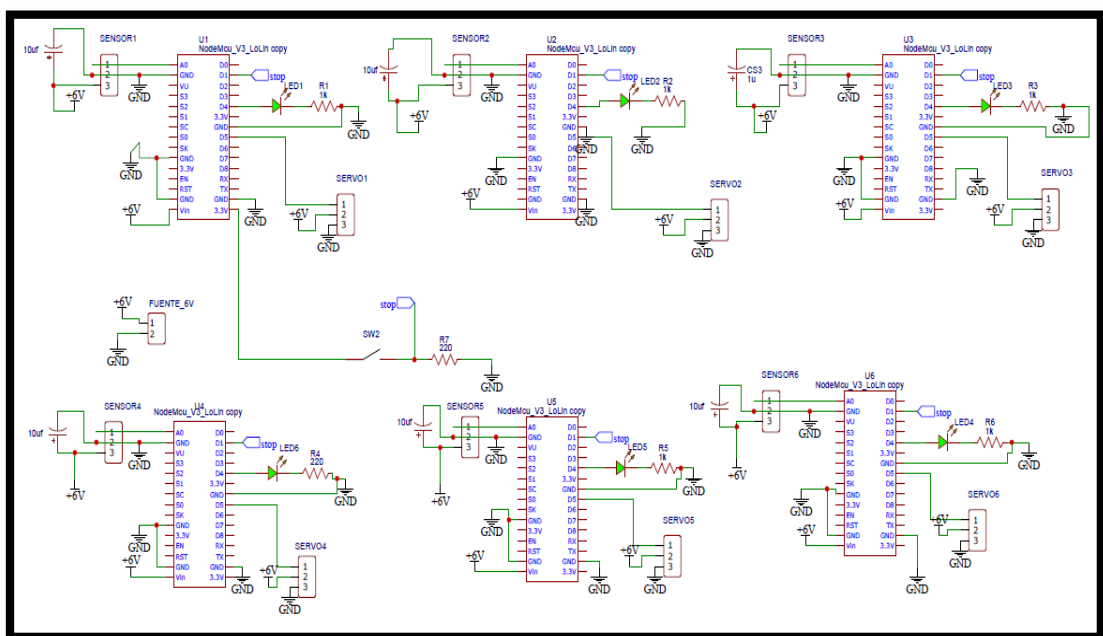


Diagrama de control para dosificación. Diseño de tolvas dosificadoras. Elaborado por Clavijo Carlos.

Para la creación del PBC se usó el programa EasyEDA en el cual se diseñó el esquema correspondiente como se representa en la figura 2.6, para el sensor SHARP se colocó un capacitor de 10 uf como recomienda el fabricante para eliminar ruidos provenientes de fuente, la placa será alimentada por una batería recargable de 6 voltios 4.5Ah, la cual funciona perfecto para los seis controladores, actuadores y sensores, conociendo que la estructura será móvil y las placas NodeMCU envían información por protocolo WiFi no se necesita cableado fuera de la estructura, la placa PCB. con su batería será colocada en una pequeña caja de control.

Figura. 2.6 Controlador difuso.



Esquema para controladores difusos. Elaborado por Clavijo Carlos.



## 2.5 Recepción de datos desde sensores ambientales

Aunque la información que entreguen los sensores no afecta al control a realizarse, es muy importante tener referencia o conocer los datos sobre temperatura, humedad y calidad del aire. Por ello se decidió colocar tres tipos de sensores, dos para temperatura y humedad, y uno para medir gases, la ubicación de dichos sensores se observa en la figura 2.7.

La ubicación se determinó por el índice de calor en ese punto con pruebas en diferentes días, diferente horario y diferentes posiciones en el galpón, se concluyó que el centro del galpón es el lugar que más índice de calor tiene, al igual que en ese punto la percepción de gas es más fuerte que en otro punto del galpón. La prueba del índice de calor se lo realizo con el sensor AM2301A antes de haber iniciado con este proyecto y los resultados se observan en la tabla 2.1.

Tabla 2.1 Índices de calor

Día	Inicio del galpón	Centro del galpón	Final del galpón
Día 1 (9 am)	17.4 °C	18.6 °C	18.1 °C
Día 2 (12 pm)	20.8°C	21.7°C	21.4°C
Día 3 (9 pm)	13.5°C	15.1°C	14.3°C

Tabla para especificar punto con más índice de calor. Elaborado por Carlos Clavijo

Figura. 2.7 Ubicación de sensores

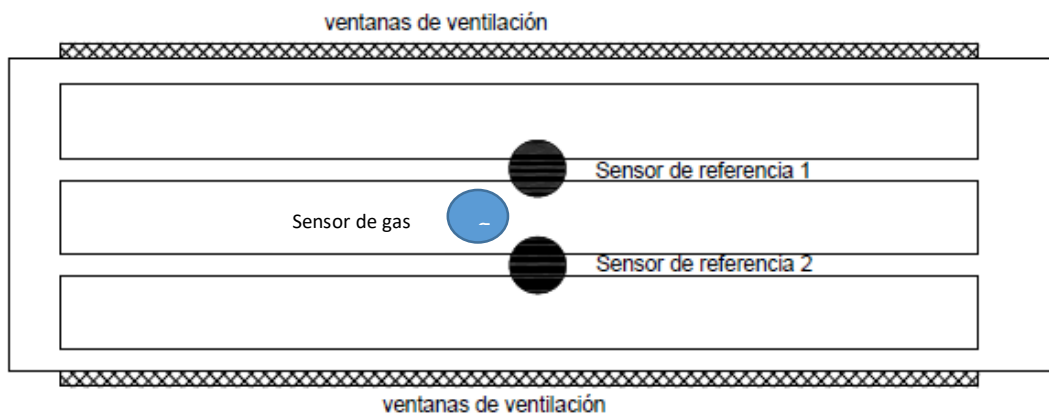
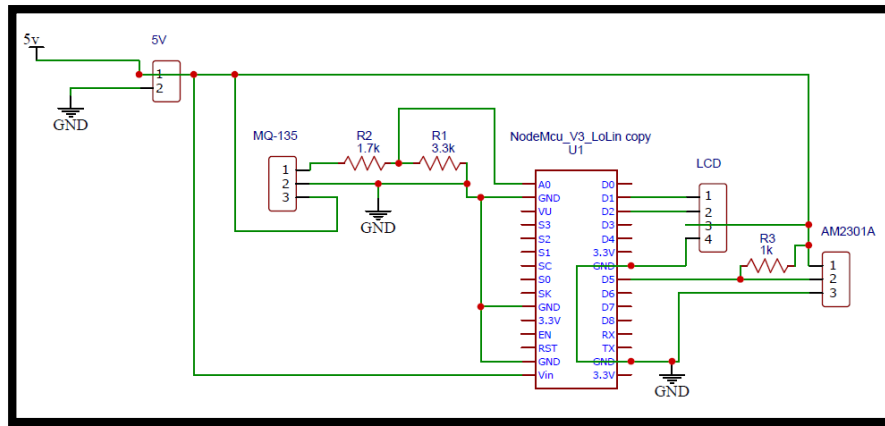


Diagrama de ubicación de sensores en el galpón. Elaborado por Carlos Clavijo

El esquema de recepción de información por parte de los sensores se observa en la figura 2.8, la cual indica la forma en que se conectaron los sensores en la placa siendo esta otra placa PCB. para el sistema.

Figura. 2.8 Recepción de sensores



Esquema de conexión de sensores. Elaborado por Carlos Clavijo.

## CAPITULO 3

### 3. DISEÑO DE SOFTWARE

#### 3.1 Control Difuso

##### 3.1.1. Entradas y Salida Difusa

Para empezar con el proyecto se realizó la simulación del control fuzzy en Matlab con sus entradas y salida difusas, el sensor de distancia SHARP será quien nos proporcione información siendo una de las entradas difusas y por lo tanto siendo el universo de discurso distancia que se puede observar en la figura 3.1 además de su ecuación 3.1, la segunda entrada difusa será la edad en semanas que será ingresada por el usuario siendo el universo de discurso 2 que se visualiza en la figura 3.2 con su ecuación 3.2 que indica el rango del universo de dicha entrada.

$$U_{\text{distancia}} = 2 \leq X \leq 7 \text{ (cm)}$$

Ec. (3.1)

Figura. 3.1. Entrada Difusa 1 Distancia.

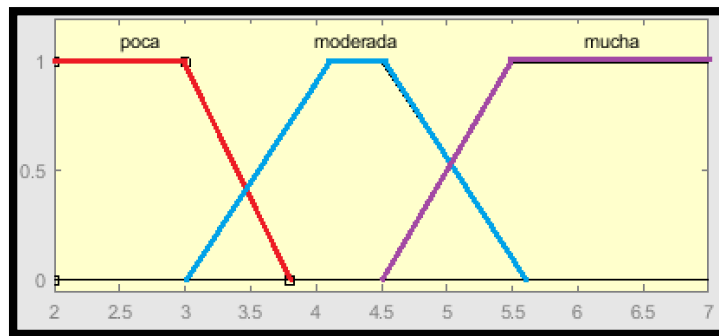


Gráfico de conjuntos del universo Distancia. Elaborado por Clavijo Carlos

$$U_{\text{semanas}} = 12 \leq X \leq 22 \text{ (semanas)}$$

Ec. (3.2)

Figura. 3.2. Entrada Difusa 2 Semanas

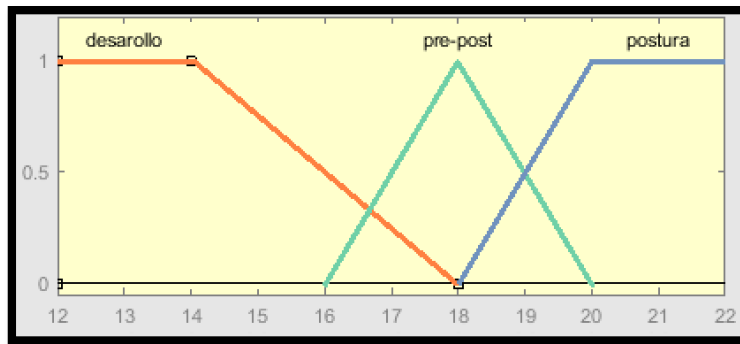


Gráfico de conjuntos del universo Semanas. Elaborado por Clavijo Carlos

La salida difusa será representada por un servo motor el cual será controlado por la placa NodeMCU, este tendrá el trabajo de apertura y cierre en el flujo del balanceado proporcionando la cantidad necesaria según la información del sensor y la edad en semanas de las gallinas, la figura 3.3 indica los conjuntos de la salida difusa y la ecuación 3.3 indica el rango del universo de salida.

$$U_{\text{Salida}} = 0 \leq X \leq 25 \text{ (grados)}$$

Ec. (3.3)

Figura. 3.3. Salida Difusa Servo.

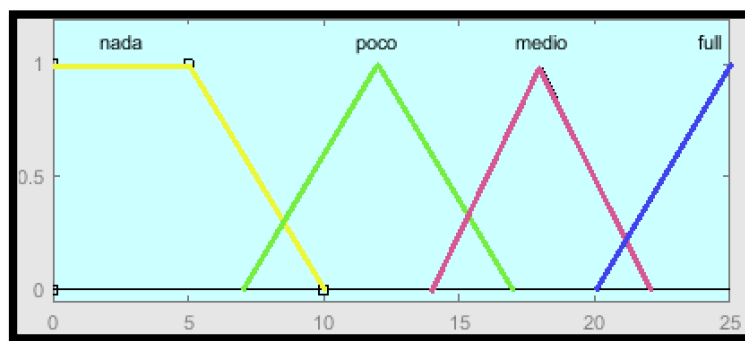


Gráfico de conjuntos de universo Servo. Elaborado por Clavijo Carlos

### 3.1.2. FAM y Reglas lógicas.

Para asociar las entradas con las salidas se diseñó la siguiente tabla de múltiples entradas y una salida, ver tabla 3.1., donde las reglas resultantes son las indicadas en la figura 3.4, el método de inferencia es por Mamdani.

Tabla 3.1. FAM.

E.D: 1 \ E.D: 2	DESARROLLO	PRE-POST	POSTURA
POCA	NADA	POCO	POCO
MODERADA	POCO	MEDIO	MEDIO
MUCHA	MEDIO	MEDIO	FULL

Tabla de matriz de asociación difusa. Elaborado por Clavijo Carlos

Figura. 3.4. Reglas lógicas.

1. If (distancia is poca) and (semanas is desarrollo) then (motor is nada) (1)
2. If (distancia is poca) and (semanas is pre-post) then (motor is poco) (1)
3. If (distancia is poca) and (semanas is postura) then (motor is poco) (1)
4. If (distancia is moderada) and (semanas is desarrollo) then (motor is poco) (1)
5. If (distancia is moderada) and (semanas is pre-post) then (motor is medio) (1)
6. If (distancia is moderada) and (semanas is postura) then (motor is medio) (1)
7. If (distancia is mucha) and (semanas is desarrollo) then (motor is medio) (1)
8. If (distancia is mucha) and (semanas is pre-post) then (motor is medio) (1)
9. If (distancia is mucha) and (semanas is postura) then (motor is full) (1)

Creación de reglas lógicas para el control. Elaborado por Clavijo Carlos

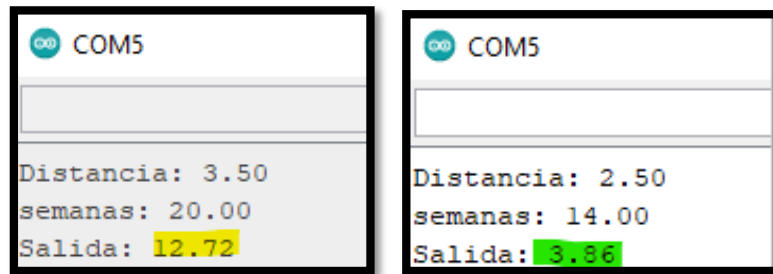
### 3.1.3. Programa Fuzzy

El código de programa Fuzzy fue desarrollado con la librería fuzzy.h desarrollada por Elmer Meneses que es compatible con ESP8266, donde se declara los universos de discurso como variables flotantes en arreglos, al igual que los conjuntos difusos, se crea variables para distancia y semanas que serán ejecutadas como funciones, la inferencia que realiza esta librería es tipo Mamdani y su defusificación es por centroide donde transforma la salida difusa en un número real el cual pertenece a las coordenadas de las x del centro de área (Ec. 3.4) de dicho conjunto difuso de salida.

$$Z = \frac{\sum_{i=1}^N x_i * U_{xi}}{\sum_{i=1}^N U_{xi}} \quad \text{Ec. (3.4)}$$

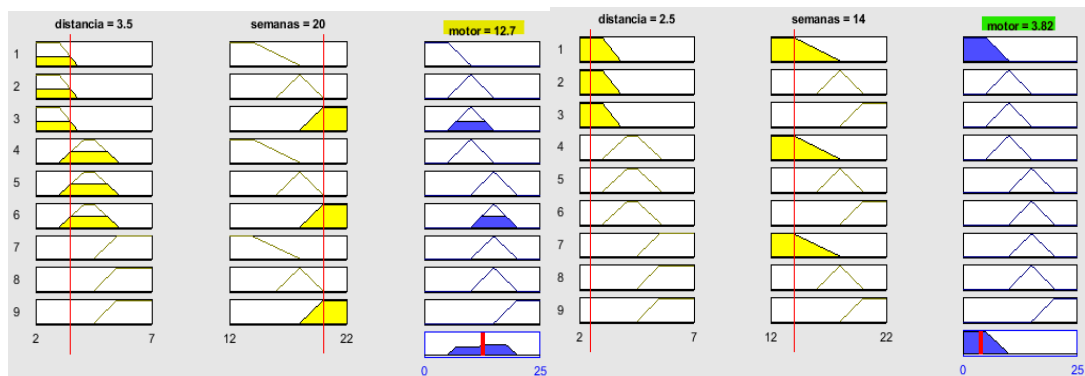
Los resultados que se obtienen son muy similares a los arrojados por el toolbox Fuzzy de Matlab donde para ambos programas se especificó valores de entradas como distancia=3.5 cm y en semanas=20 dando como resultado en la salida 12.72 en Arduino.ide y 12.7 en el toolbox de Matlab, de la misma manera se cambió los valores de entrada y su resultado son muy parecidos como se puede ver en la figura 3.5 y 3.6.

Figura. 3.5. Respuesta fuzzy en Arduino.ide



Resultado en consola de Arduino controlador fuzzy. Elaborado por Clavijo Carlos

Figura. 3.6. Respuesta en toolbox de Matlab.

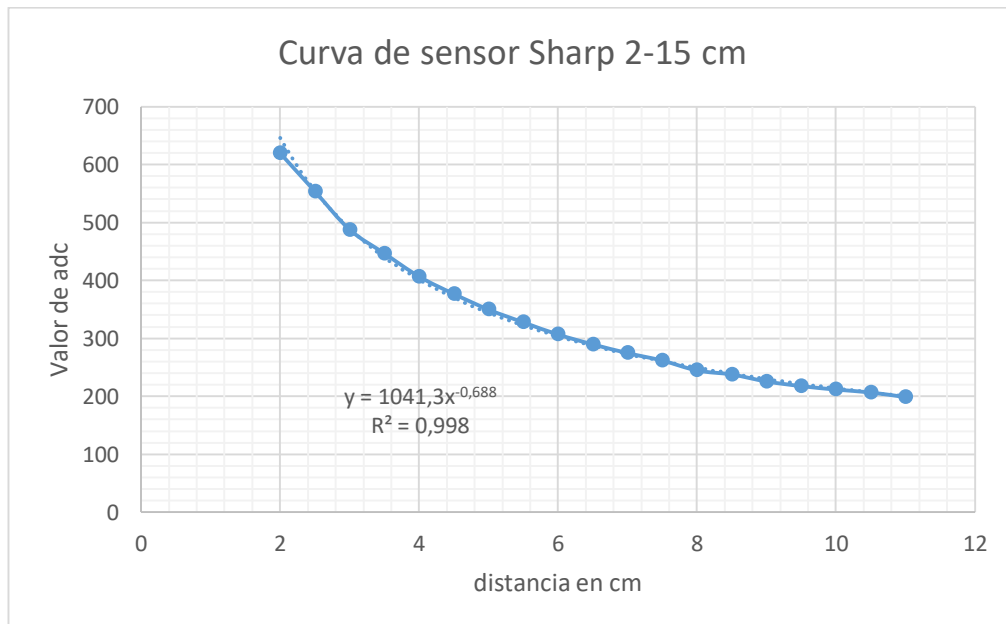


Resultado en toolbox fuzzy de MATLAB. Elaborado por Clavijo Carlos

### 3.2. Calibración de Sensor SHARP

Una vez definido el sistema difuso, se empieza hacer la calibración de los sensores analógicos SHARP GP2Y0A51SK0F que tienen un rango de 2 a 15 cm de distancia, para esta calibración se realizó el uso del programa Arduino.ide donde obtenemos el valor de ADC arrojado por el sensor, colocando un objeto a cierta distancia del sensor que va ir aumentando cada cinco milímetros empezando en 2 cm y terminando en 14 cm esto se realiza para obtener la curva característica del sensor y la ecuación resultante que se ven reflejada en la figura 3.7 y ecuación 3.5.

Figura. 3.7. Curva de sensor Sharp.



Curva resultante del sensor SHARP. Elaborado por Clavijo Carlos

$$y = 1041,3x^{-0,688} \quad \text{Ec. (3.5)}$$

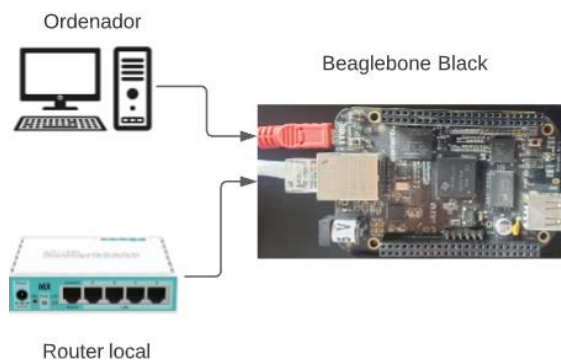
Despejando x de la ecuación 4 se puede obtener el valor de la distancia en (cm) que está midiendo el sensor, con ese valor trabajara el programa fuzzy ya que la entrada difusa 1 es de 2 a 7 cm. Para la segunda entrada será ingresada desde un dispositivo inteligente por protocolo MQTT que será explicado más adelante. La librería Servo.h desarrollada por Michael Margolis es la usada para poder controlar el servo donde se ingresa en números enteros el grado entre 0 a 180 grados, en este caso está limitada de

0 a 25 grados que es la apertura suficiente para una porción máxima de balanceado en el canal

### 3.3. Servidor local de MQTT

Dejando por el momento a un lado la placa NodeMCU donde ya está funcionando el control fuzzy, se realizará la configuración de un servidor local en la placa BeagleBone Black donde se podrá comunicar de forma bidireccional con el router local y las placas ESP8266. Para que la comunicación sea exitosa por medio del programa PuTTY se instala MQTT-WebSocket conectado por medio de un cable USB a un computador y conectando un cable de ethernet RJ45 en la placa BeagleBone y el otro extremo a un puerto ethernet del router local para tener acceso a internet ver figura 3.8, iniciado la sesión en PuTTY se observa como usuario `debian@beaglebone:~$` otorgando permisos de súper usuario con el comando `sudo su`, se puede iniciar con la instalación de MQTT con `mosquitto` ingresando `apt-get install mosquitto` y luego con el mismo comando agregando `-clients` para comprobar la instalación de mosquitto en la BeagleBone se publicara un mensaje como se observa en la figura 3.9.

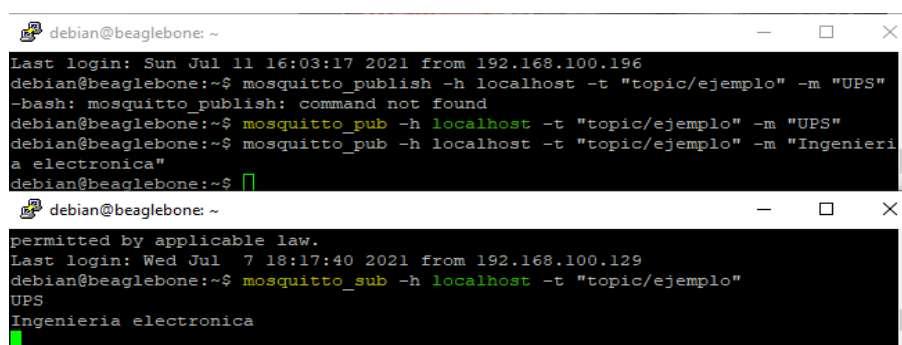
Figura. 3.8. Conexión a BeagleBone Black.



Esquema de conexión de la placa BBB. Elaborado por Clavijo Carlos



Figura. 3.9. Comprobación de mosquitto.



```
debian@beaglebone: ~  
Last login: Sun Jul 11 16:03:17 2021 from 192.168.100.196  
debian@beaglebone:~$ mosquitto_publish -h localhost -t "topic/ejemplo" -m "UPS"  
-bash: mosquitto_publish: command not found  
debian@beaglebone:~$ mosquitto_pub -h localhost -t "topic/ejemplo" -m "UPS"  
debian@beaglebone:~$ mosquitto_pub -h localhost -t "topic/ejemplo" -m "Ingenieria  
a electronica"  
debian@beaglebone:~$ █  
  
debian@beaglebone: ~  
permitted by applicable law.  
Last login: Wed Jul 7 18:17:40 2021 from 192.168.100.129  
debian@beaglebone:~$ mosquitto_sub -h localhost -t "topic/ejemplo"  
UPS  
Ingenieria electronica  
█
```

Comprobación por cmd en BBB de pub-sub con mosquitto. Elaborado por Clavijo Carlos

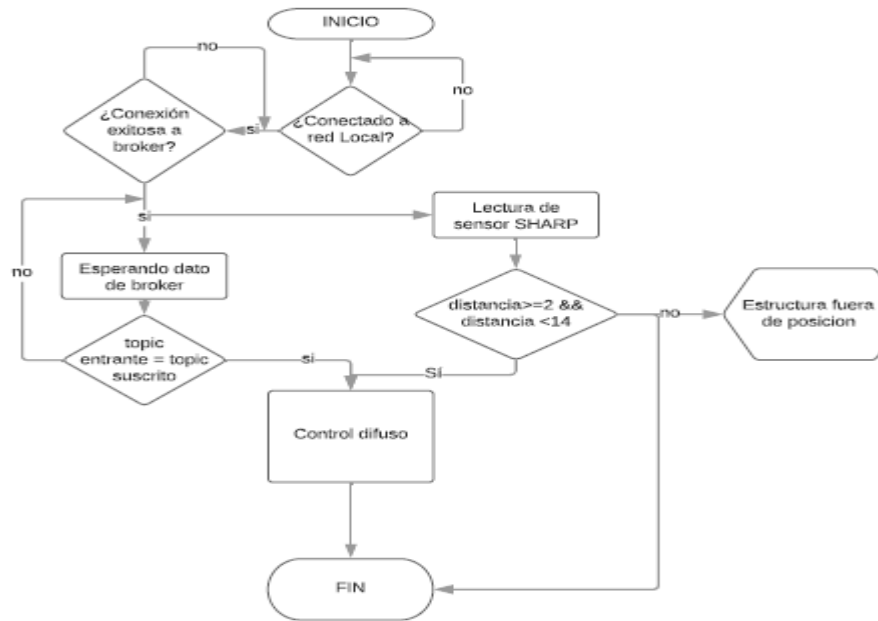
### 3.4. MQTT en NodeMCU

Ya instalado el servidor local de MQTT en la placa BeagleBone Black, devuelta en la placa NodeMCU se instalará la librería PubSubClient.h de Nick O'Leary, la cual servirá para la comunicación MQTT con el broker donde es necesario conocer el nombre de la red local del router y su contraseña, pero el paso más importante es colocar la ip asignada por el router al broker, ya que a esa dirección ip llegaran todos los mensajes que estén registrados dependiendo los tópicos suscritos. Cada placa NodeMCU está suscrito a un solo topic y publica con el mismo topic, está suscrito para recibir el dato de las semanas desde la nube y publica cuando este haya terminado el recorrido de colocar balanceado o se haya oprimido el botón de emergencia.

#### 3.4.1. Algoritmo implementado en NodeMCU.

El algoritmo implementado en la placa NodeMCU está representada por la figura 3.10, cuando el programa inicia intenta la conexión a la red local y solo cuando esté conectada se conectará al servidor bróker, el mensaje que se ha enviado desde el broker serán con distintos tópicos para cada placa el control inicia solo si el mensaje llega con el topic correspondiente al igual que si el sensor detecta distancias incoherentes no arranca la parte de control.

Figura. 3.10. Algoritmo implementado en NodeMCU.



Demostración del algoritmo implementado en placas de control. Elaborado por Clavijo Carlos

### 3.5. Conexión entre Node-RED local e IBM Cloud

En IBM cloud se instaló Node-RED a nivel de cloud foundry, plataforma como un servicio (PaaS), adicionalmente en el servicio de IBM Watson IoT platform se agregó a la placa BeagleBone Black (figura 3.11) de la misma manera se creó una clave API la cual servirá para lograr una conexión exitosa entre el dispositivo y Node-RED en la nube, esto se logra por el nodo IBM IoT en el cual se ingresa la clave API y los detalles del dispositivo creado, logrando una conexión segura entre el dispositivo físico y la nube.

Figura. 3.11. Dispositivo agregado en IBM Watson.

Información de conexión	
Información de conexión básica sobre este dispositivo.	
ID de dispositivo	central
Tipo de dispositivo	BeagleBone_black
Fecha de adición	4 de jul. de 2021 12:18 PM
Añadido por	carl.alejo7@gmail.com
Estado de conexión	Desconectado
	Conectado por última vez: 4 de jul. de 2021 1:16 PM
	Dirección del cliente: 157.100.173.238 Señal segura
	Duración: unos segundos
	Datos transferidos: 236 B

Información detallada del dispositivo enlazado. Elaborado por Clavijo Carlos

### 3.5.1. Envío de datos desde IBM Cloud

Para el envío y recepción de datos se usa el formato JSON (figura 3.12) el cual ordena la información y permite la extracción con facilidad por etiquetas, de esta manera el receptor escoge la etiqueta deseada y la usa sin importar que tipo de variable sea (Int, float, String)

Figura. 3.12. Envío de dato en formato JSON.

```

1 msg.payload={ enviar : msg.payload : Object
2   d:{
3     "disp1":msg.payload  ▼ object
4   }
5 }
6 return msg;           ▼d: object
                        disp1: 16
  
```

Estructura del mensaje enviado hacia la nube. Elaborado por Clavijo Carlos

El mensaje en formato JSON es enviado por el nodo IBM IoT y se ve reflejado en los sucesos del dispositivo creado en IBM Watson como se indica en la figura 3.13.

Figura. 3.13. Datos recibidos por IBM Watson.

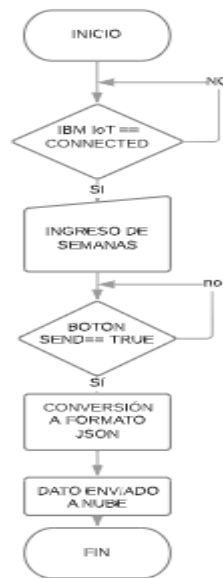
Suceso	Valor	Formato	Último recibido
estado	{"d":{"disp1":19}}	json	hace unos segundos
estado	{"d":{"disp6":18}}	json	hace unos segundos
estado	{"d":{"disp5":18}}	json	hace unos segundos
estado	{"d":{"disp4":14}}	json	hace unos segundos
estado	{"d":{"disp3":17}}	json	hace unos segundos

Formato de recepción de dato. Elaborado por Clavijo Carlos

### 3.5.2. Algoritmo para envío de datos desde IBM Cloud

Para el envío de datos el nodo IBM IoT de salida debe estar en estado “connected”, una vez verificado ese estado se ingresa el dato deseado y se lo envía para que sea transformado a formato JSON para luego ser recibido por la nube de IBM, la figura 3.14 muestra el algoritmo.

Figura. 3.14. Algoritmo de envío de dato a la nube.



Demostración del algoritmo implementado para el envío de edad en semanas. Elaborado por Clavijo Carlos

### 3.5.3. Recepción de datos desde IBM Cloud

Como se ha dicho la información llega en formato JSON y sale en formato JSON para lo cual se debe convertir a un dato según sea la etiqueta, como se observa en la figura 3.15 se crea una variable que se le asigna al dato que ingrese en JSON como {"d": {"disp1": dato}}, donde 'dato' será el valor que sea usado para ser enviado por MQTT al cliente.

Figura. 3.15. Conversión de JSON a mensaje.

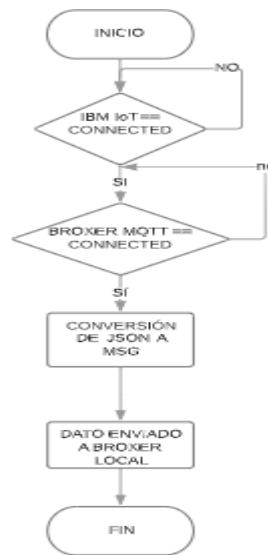
```
1 var valor = msg.payload.d.disp1;
2 msg.payload =valor;
3 return msg;
```

Obtención del valor desde JSON. Elaborado por Clavijo Carlos

### 3.5.4. Algoritmo para recepción de datos desde IBM Cloud

De la misma manera para la recepción de datos como se observa en la figura 3.16, el nodo de IBM IoT en Node-RED debe estar en estado "connected", a este nodo llegarán todos los mensajes que se encuentren en la nube que esta aliada al dispositivo, se debe interpretar el mensaje que llega en formato JSON para que sea enviado por MQTT.

Figura. 3.16. Algoritmo para recepción de datos desde la nube



Demostración del algoritmo implementado para el recepción de datos. Elaborado por Clavijo Carlos

### 3.6. Programación en Node-RED local

En la placa BBB usando <ip BBB:1880> con el puerto 1880 ingresa a la herramienta Node-RED de manera local el cual será el programa principal utilizado en este proyecto donde se empieza a programar la recepción de datos, envío de datos, ejecución de acciones entre otros, la interfaz tiene nodos muy valiosos e importantes que serán usados para la programación. La BBB tiene como prioridad enviar el dato de la edad en semanas de las gallinas a las placas NodeMCU, para poder ejecutar el control difuso, esto puede llevarse a cabo mediante el ingreso por un dispositivo inteligente o de manera manual desde el tablero central. También recibe información por I2C. del sensor BME280 el cual indica la temperatura y humedad relativa del ambiente, adicional mente activa el movimiento de la estructura.

#### 3.6.1. Temperatura y humedad con BME280

Para la recepción de la temperatura y humedad en el ambiente se usa el ya mencionado sensor BME280 que en Node-RED tiene su propio nodo que funciona por I2C. el cual necesita el número de bus y la dirección que ha sido asignada, para ello se debe conocer la dirección asignada por la placa al sensor, con el comando `I2C.detected -r 2` en la consola de la placa BBB se observara la dirección que fue asignada y se puede verificar en la figura 3.17, la cual será ingresada en el nodo de BME280.

Figura. 3.17 Dirección I2C del sensor BME280 en BBB.

```
debian@beaglebone:~$ i2cdetect -r 2
WARNING! This program can confuse your I2C bus, cause data loss and worse!
I will probe file /dev/i2c-2 using receive byte commands.
I will probe address range 0x03-0x77.
Continue? [Y/n] y
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  76  --  --  --  --  --  --  --  --  --
debian@beaglebone:~$
```

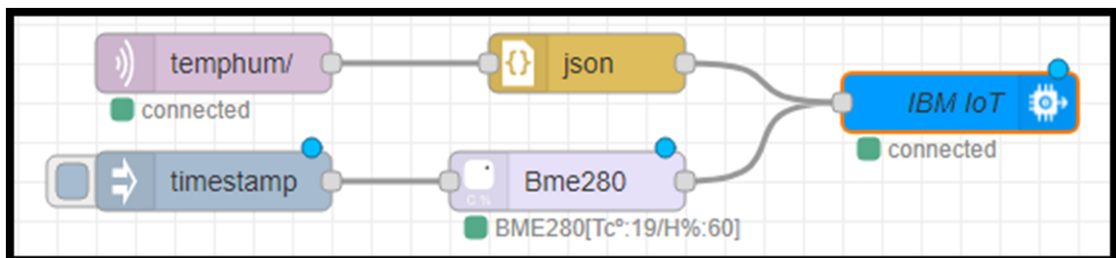
Muestreo de direcciones I2C. en BBB. Elaborado por Clavijo Carlos

Ingresando el bus número 2 y la dirección 0x76 en el nodo ya se observan datos que serán enviados por el nodo de IBM IoT out (ver figura 3.18) a la nube donde se observa en un gráfico los valores de temperatura y humedad del galpón desde algún dispositivo inteligente. Este sensor fue catalogado como un sensor de referencia uno, que estará ubicado en un punto céntrico del galpón donde se siente que hay un encierro de aire y una mayor sensación térmica como se mostró en la figura 2.7.

### 3.6.2. Temperatura, humedad e índice de calor con AM2301A

Una placa NodeMCU es usada como receptor del sensor AM2301A el cual nos entrega el valor de la temperatura y humedad relativa, para el envío hacia Node-RED local se realizó una serialización de las variables para ser enviadas por MQTT hacia la placa BBB donde será enviada hacia la nube, este sensor nos entrega dichos valores en un cierto punto del galpón siendo un sensor de referencia dos para obtener en ese punto y poder saber que parámetros se están dando durante las 24 horas, el flujo usado en Node-RED para el envío de datos sobre los sensores se visualiza en la figura 3.18.

Figura. 3.18 flujo de sensores.



Flujo para recepción de sensores en Node-RED. Elaborado por Clavijo Carlos

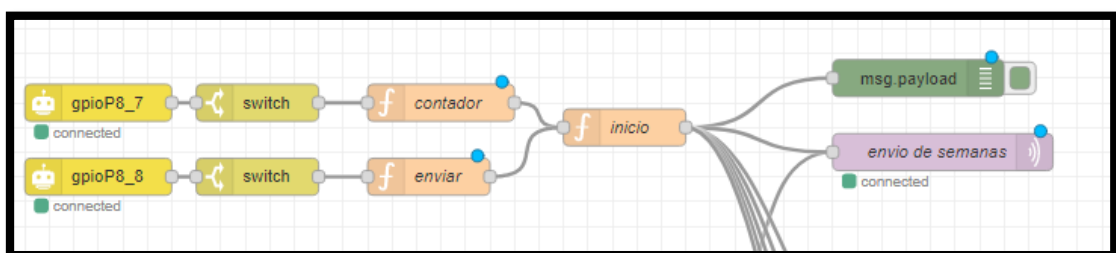
### 3.6.3. Gases en el ambiente.

En el mismo mensaje de MQTT del sensor AM2301A llega la información desde la placa NodeMCU de la cantidad de gases en partes por millón (ppm) existentes en el galpón estos gases son los más comunes en los desechos de las aves tales como el amonio y monóxido de carbono, el sensor usado para la detección de gases es el MQ-135. El mensaje de JSON con el que llegan dichos datos son “CO” para el monóxido de carbono y “NH4” para el amonio.

### 3.6.4. Envío de datos de forma manual

En el caso de que no haya conexión a internet por algún motivo, la placa BBB es el servidor bróker y mientras este tenga alimentación y esté conectado al router local existirá comunicación MQTT con las placas de control, para ello se colocó dos pulsadores, un pulsador sirve para ajustar el valor de la edad en semana de las gallinas y el otro pulsador sirve para enviar el dato por MQTT a las placas de control una vez que se haya ajustado las semanas, para ello se usaron dos nodos de función la primera función sirve como contador en ascendente para ajustar el valor de las semanas en trece y veintiuno, y el otro nodo de función guarda el valor de semanas y espera a que el botón enviar sea presionado para que sea enviado a las placas de control, el flujo se indica en la figura 3.19.

Figura. 3.19 Flujo para envío de datos manual.



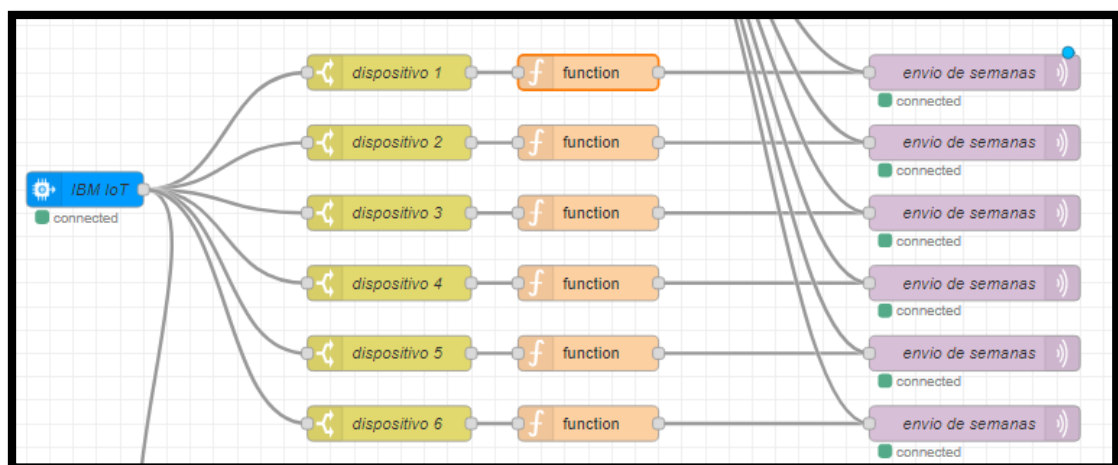
Visualización de envío de datos manual. Elaborado por Clavijo Carlos

La información se verá reflejada en una pantalla LCD 16x2, donde la primera línea indica el valor a ingresar mientras la segunda línea el valor que fue enviado a las placas de control.

### 3.6.5. Recepción de datos desde IBM IoT

Esta parte del flujo recibe la misma información de semanas, pero desde un dispositivo inteligente y para ello es usado el nodo IBM IoT IN en el cual se enlaza por medio de una API KEY al dispositivo creado en IBM Watson y lee todo lo que se encuentre en aquel dispositivo, al recibir todos los datos es necesario usar solo la información que interesa debido a eso se usa el nodo switch con el cual se separa los datos dependiendo al topic con el que lleguen, la información que ingresa por el nodo de IBM IoT llega en formato JSON y para transformar esa información a un valor que le sirva a la placa de control se usa un nodo de función, en la figura 3.20 se observa el flujo.

Figura. 3.20 Flujo de recepción de datos desde IBM Watson.



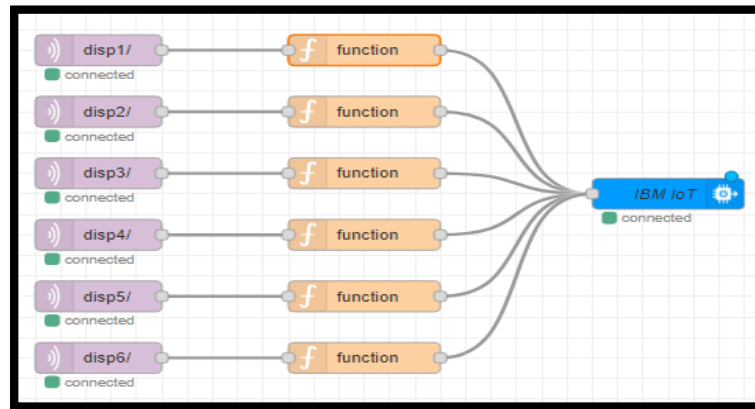
Visualización de nodos para la recepción de datos. Elaborado por Clavijo Carlos

### 3.6.6. Envío de datos a IBM IoT

Desde las placas de control se envía un dato que indica el estado de las mismas, es decir si la estructura está en posición de dosificación enviara un dato en true, si la estructura termino o no está en posición mandara un dato de falso, los datos son recibidos por protocolo MQTT con diferentes tópicos correspondientes a cada placa de control, con un nodo de función se convierte el dato recibido desde MQTT en formato JSON y es enviado por el nodo IBM IoT OUT el cual debe tener la misma API KEY del dispositivo agregado en IBM Watson, los nodos se observan en la figura 3.21.



Figura. 3.21 Envío de datos a IBM IoT



Demostración del flujo para envío de datos. Elaborado por Clavijo Carlos

### 3.7. Programación en Node-RED de IBM cloud

En este punto del proyecto se empieza a programar la parte de Node-RED en la nube con lo que algunos pasos serán los mismos de lo que se hizo de manera local, además aquí se presentará el dashboard que será el medio de interacción con el usuario. Se enviará los datos hacia el Node-RED local al igual que se recibirá del mismo.

#### 3.7.1. Recepción de información de Temperatura, humedad y gases.

La información que está en la nube subida por los sensores se encuentra en formato JSON como se indica en la figura 3.22.

Figura. 3.22 Información subida del sensor en la nube.

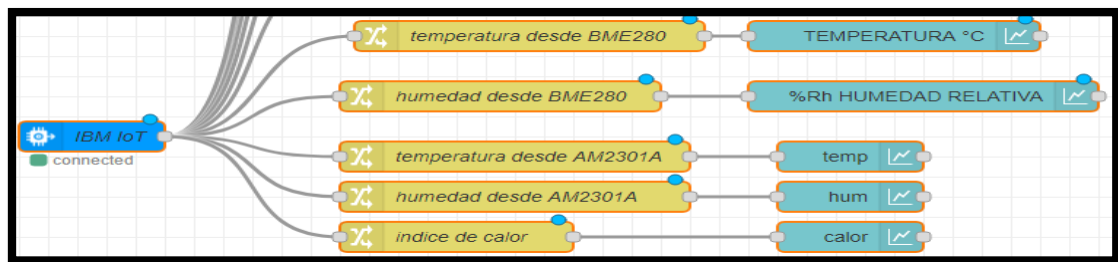
Suceso	Valor	Formato	Último recibido
estado	{"temperature_C":17.65,"humidity":39.4112670...	json	hace unos segundos
estado	{"temperature_C":17.65,"humidity":39.4274764...	json	hace unos segundos
estado	{"temperature_C":17.66,"humidity":39.4977413...	json	hace unos segundos

Formato de llegada de datos de sensor a nube. Elaborado por Clavijo Carlos

Para obtener los datos, de la misma manera se usó el nodo IBM IoT in donde se recibe toda la información que llegue al nodo, esta información se la clasificó o escogió

dependiendo los datos que interesan usando un nodo change, estos se verán visualizados en gráficos correspondientes a cada valor escogido (figura 3.23), los datos que se verán reflejados son dos provenientes del sensor BME280 representando la temperatura y humedad, y tres del sensor AM2301A que de la misma manera representaran la temperatura y humedad pero adicionalmente el índice de calor.

Figura. 3.23 flujo de recepción de temperatura y humedad.



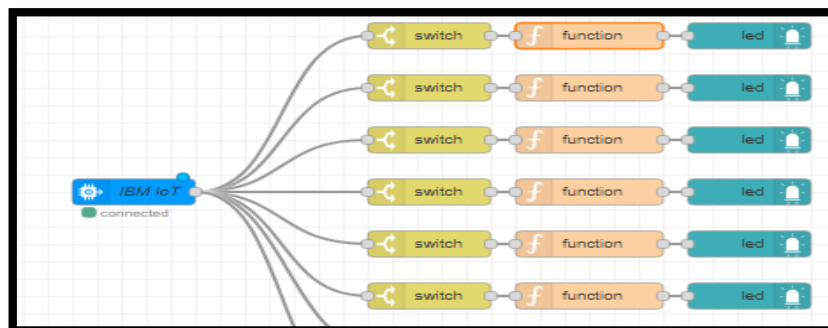
Visualización de nodos recepción de datos de sensores. Elaborado por Clavijo Carlos

El sensor de gas MQ-135 envía información sobre el monóxido de carbono y el amonio existente en el ambiente.

### 3.7.2. Recepción de datos desde placas NodeMCU

Como ya se mencionó antes las placas de control envían datos de estado, esta información llega a IBM Watson y es leída por el nodo de IBM, el procedimiento es el mismo. Se separa la información y se convierte el JSON para que el nodo led se encienda dependiendo del estado de cada placa (figura 3.24).

Figura. 3.24 Flujo de estado de placas de control



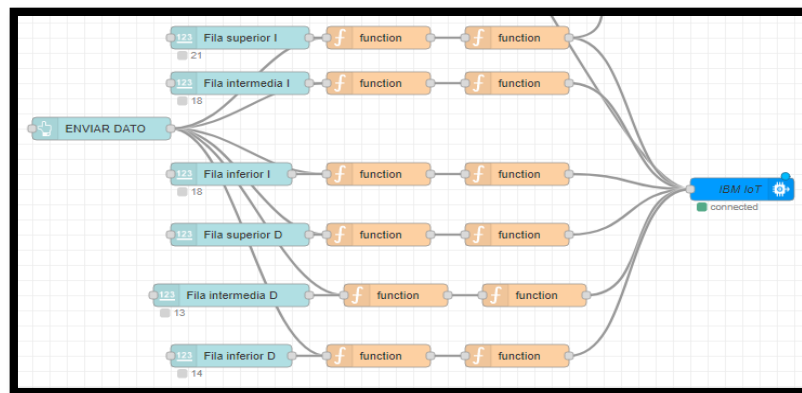
Visualización de nodos para el estado de placas. Elaborado por Clavijo Carlos

### 3.7.3. Envío de edad en semanas a placas NodeMCU

Para el envío de la edad en semanas de las gallinas se usaron nodos del dashboard para que el usuario pueda ingresar el valor deseado con el que van a trabajar las placas de control el flujo se puede visualizar en la figura 3.25, por lo general, en una nave de jaulas se ubican gallinas de la misma edad y es muy raro que se coloquen de diferente edad en una misma nave, pero si existen casos en los que por optimizar espacio se colocan de diferente edad en la misma nave, pero en diferentes filas, para ello se ingresa el dato de semanas de cada fila.

Una vez ingresado todos los datos el usuario da clic u oprime el botón enviar dato, para que los datos lleguen todos a la misma vez para que las placas de control difuso puedan empezar todos al mismo tiempo.

Figura. 3.25. Envío de dato a nube.

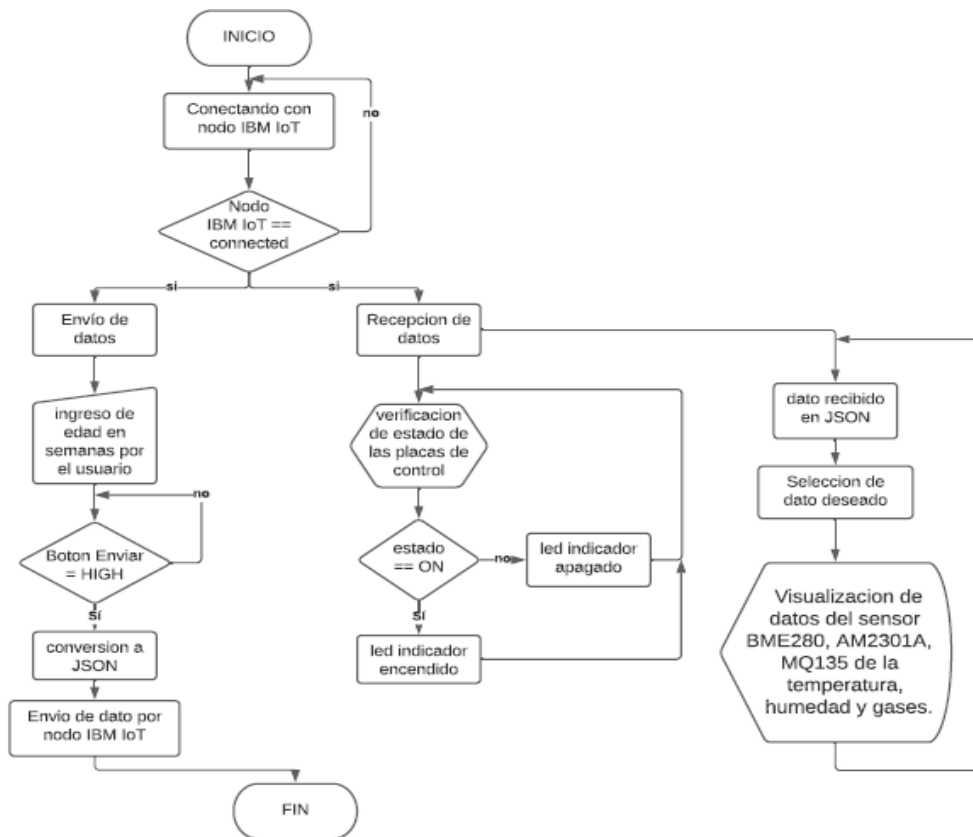


Visualización de nodos para envío de semanas hacia nube. Elaborado por Clavijo Carlos

### 3.7.4. Algoritmo de Node-RED en la nube

El proceso que se ejecuta en el Node-RED de la nube se visualiza en la figura 3.26.

Figura. 3.26. Algoritmo de Node-RED en la nube



Demostración del algoritmo implementado en Node-RED de la nube. Visualización de nodos recepción de datos de sensores. Elaborado por Clavijo Carlos

## CAPÍTULO 4

### 4. IMPLEMENTACIÓN Y PRUEBAS

#### 4.1 Implementación de la estructura.

La estructura fue diseñada para las medidas de la nave 3, la cual es ideal para el galpón debido a que no consume espacio lateral, que es el más usado por el personal del galpón en el proceso de recolección de huevos, las ruedas colocadas en la parte inferior permiten una movilidad sin esfuerzo siendo dicha estructura fácil de deslizar a lo largo del galpón. Las tolvas colocadas en la parte superior tienen la ventaja de no interrumpir la luz solar a ningún sector de la nave cuando la estructura está esperando a ser operada, la figura 4.1 muestra la estructura dentro del galpón.

Figura. 4.1 Estructura con tolvas.



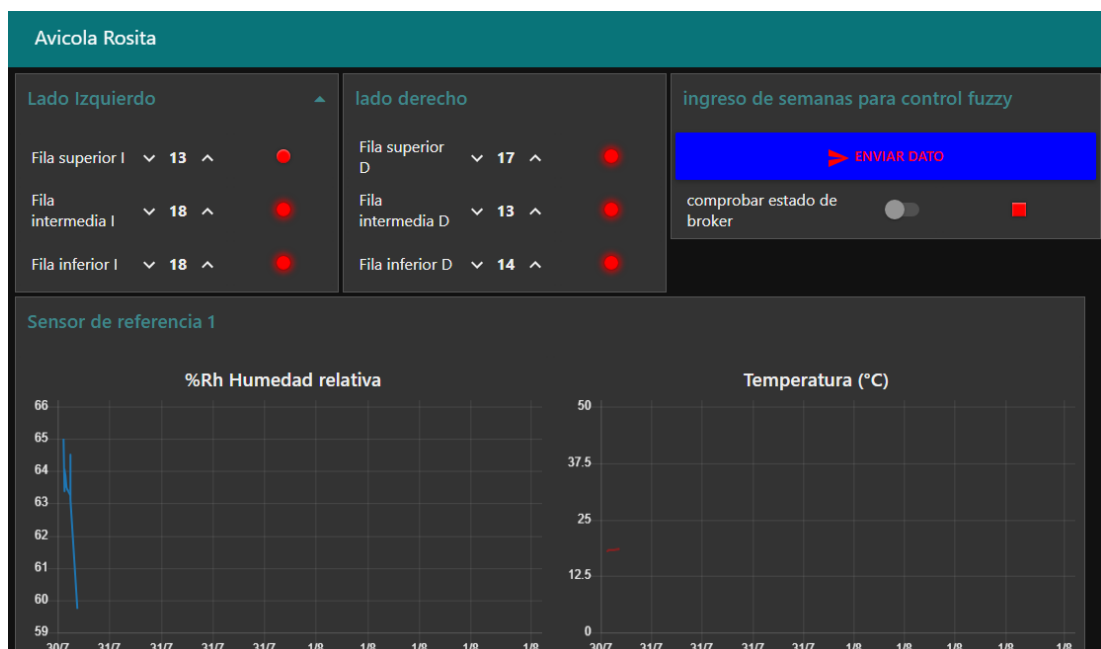
Estructura móvil para dosificación de balanceado. Elaborado por Carlos Clavijo.

#### 4.2 Interfaz de usuario

Para que este proyecto sea amigable al usuario se creó una página web [avicularosita.web.app](http://avicularosita.web.app) (ver anexo 1) la cual solicita el ingreso de un usuario y su contraseña, solo de esta manera el usuario puede acceder al control y monitoreo del

galpón, la página fue realizada en JavaScript y HTML el cual fue subido a Firebase para un acceso desde cualquier red, una vez ingresado los datos correctos la página direcciona al dashboard de Node-RED en la nube donde se puede visualizar toda la interfaz que controla y monitorea el galpón. En la figura 4.2 se observa la interfaz que se visualizara al usuario.

Figura. 4.2 Interfaz gráfica para el usuario



Dashboard de Node-RED que interactúa con el usuario. Elaborado por Carlos Clavijo

El bloque nombrado “lado izquierdo” permite el ingreso de la edad en semanas para tres placas de control difuso, como su nombre lo indica estas están ubicadas en la parte izquierda de la nave, de la misma manera actúa el bloque llamado “lado derecho”. El bloque “ingreso de semanas para control fuzzy” consta de dos botones el primer botón “enviar dato” envía los 6 datos de semanas hacia la nube para ser recibido por el bróker local. Los bloques de “sensores” muestran los datos obtenidos desde el bróker local por cada día las 24 horas.

### 4.3 Visualización de sensores

Como se vio en el capítulo 2 figura 2.7, las ubicaciones de los sensores son en los puntos donde más calor se siente, estos valores se verán visualizados en el dashboard

de Node-RED, el catálogo de crianza de las aves indica condiciones medioambientales específicas (ver tabla 4.1) para que el bienestar y rendimiento de las aves sea el más óptimo. la temperatura, humedad y nivel de gases tóxicos en el ambiente son de suma importancia.

Tabla 4.1 Condiciones ambientales optimas

Variable	Condiciones ambientales
Temperatura	18 – 20 (°C)
Humedad	60 – 70 (%Hr)
CO (Monóxido de Carbono)	< 40 ppm
NH3 (Amoníaco)	< 20 ppm

Tabla especificando los valores recomendados por el catálogo de crianza de las aves. Elaborado por Carlos Clavijo.

Aunque en la avícola Rosita no se tiene un sistema de aire acondicionado, calefacción o algún sistema automatizado para la ventilación y control de temperatura. Los datos que arrojan los sensores ayudaran a los propietarios a predecir posibles problemas con la mortalidad de las aves o producción en los que se puedan ver envueltos con las condiciones ambientales del galpón.

Los sensores están conectados a una placa NodeMCU que se encarga solo de procesar la información de sensores y dicha información es enviada por MQTT hacia el bróker para luego ser subida a la nube y el usuario pueda monitorear en tiempo real en que condiciones ambientales esta su galpón, la visualización de la caja contendora de sensores y placa BBB se indica en la figura 4.3.

Figura. 4.3 caja de bróker y sensores.



Gráfico de caja de servidor broker y placa de sensores. Elaborado por Clavijo Carlos.

#### 4.4 Implementación de controladores difusos

En el capítulo 2 se creó una placa PCB. donde se colocaron las placas NodeMCU para el control de dosificación, esta PCB. fue implantado en un gabinete eléctrico con una batería recargable que es su fuente de alimentación, la caja tiene seis luces indicadoras que muestran el estado de las placas de control y dos indicando si existe un paro de emergencia o los sensores están fuera de posición, la misma caja tiene un interruptor para apagar el sistema si fuese necesario. Dicho gabinete se lo puede apreciar en la figura 4.4

Figura. 4.4 Caja de controladores difusos

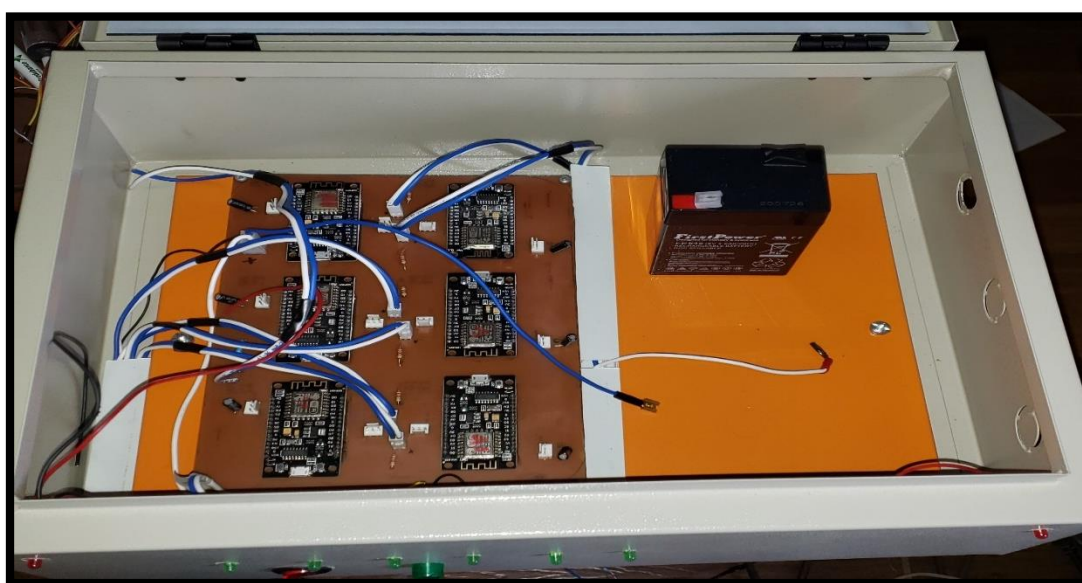


Gráfico de controladores difusos. Elaborado por Carlos Clavijo.



Al ingresar un dato de 21 semanas a cualquier fila, y el sensor de la misma fila detecta una distancia de 6 cm el servo motor gira 21 grados dejando caer 120 g de balanceado en 2 segundos en un espacio de 10 x 5 cm.

#### 4.5 Implementación de red WLAN en el galpón

El punto más cercano a internet se encuentra a 70 metros del galpón, por lo cual se optó por instalar un router repetidor el cual provee de conexión internet a la placa BBB que será el intermediario entre las placas de control y la nube como se observa en la figura 4.1, como ya se mencionó en el capítulo 2 la ventaja de las placas NodeMCU es que funcionan con protocolo TCP-IP y tienen un gran alcance siendo así que la ubicación del router WiFi será en la mitad del galpón.

Figura. 4.5 esquema de conexión WLAN

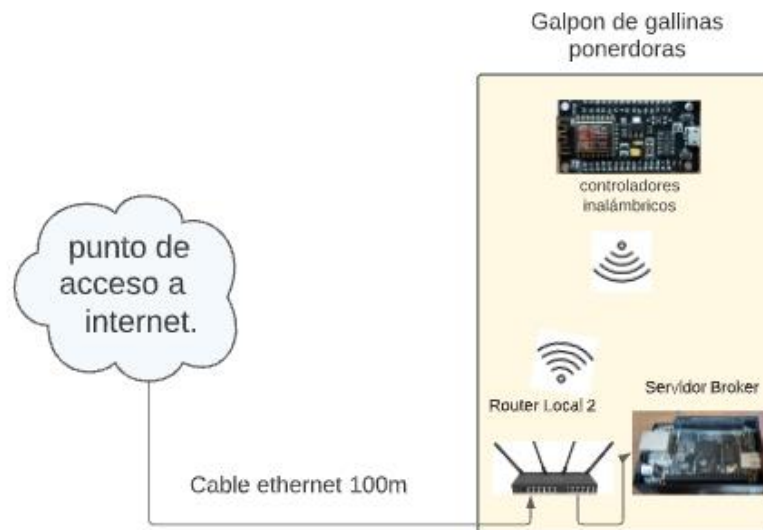


Diagrama de ubicación de router y aparatos inalámbricos y alámbricos. Elaborado por Carlos Clavijo.

#### 4.6 Prueba de control difuso.

Al iniciar el control difuso los servomotores regulan su ángulo de apertura constantemente, siendo 25 grados el ángulo de flujo máximo de dosificación para una caída de 60 gramos por cada 12 cm, es decir por cada jaula dosifica 300 gramos lo que equivale al alimento para 5 aves, como el catálogo de crianza de aves indica que desde las 21 semanas se debe colocar 120g x ave al día el sistema está diseñado para colocar

los 120 gramos pero en dos turnos uno a las 7 a.m. y otro a las 2 p.m. según decida el usuario.

Para verificar el funcionamiento del control se realizaron diversas pruebas en las que a propósito se colocaron ciertas cantidades de balanceado para observar cómo reacciona el control difuso, ingresando diferentes edades en semanas empezando desde la semana trece hasta la semana veintiuno, de la misma manera en el canal se empezó por tenerlo completamente vacío a tenerlo con cierta cantidad de balanceado, en la tabla 4.2 se puede verificar los resultados que se obtuvieron teniendo en cuenta que es de un solo turno y el promedio de 10 pruebas.

Tabla 4.2 Prueba de dosificación

Balanceado en canal por 12 cm	Semanas	Resultado	Ideal
Vacío	13	29.7 gramos	32 gramos
Vacío	18	41.2 gramos	38 gramos
Vacío	21	58.4 gramos	60 gramos
10 gramos	13	20.4 gramos	32 gramos
10 gramos	18	31.1 gramos	38 gramos
10 gramos	21	51.4 gramos	50 gramos
30 gramos	13	4.2 gramos	32 gramos
30 gramos	18	6.9 gramos	38 gramos
30 gramos	21	31.3 gramos	60 gramos

Resultados de un promedio de 10 pruebas con distinto nivel de balanceado en el canal de dosificación.  
Elaborado por Carlos Clavijo

#### **4.7 Prueba de dosificación.**

Una vez comprobado el control difuso se procedió a realizar la dosificación completa. Como todas las aves son de la misma semana se ingresaron datos como semana 21, datos que fueron ingresados por un celular conectado a internet. Para obtener una dosificación ideal de 60 gramos por cada 12 cm, como era de esperarse en ciertas partes a lo largo del canal existía comida por lo cual el servomotor controló el flujo de balanceado a dosificar, el trabajador antes del proyecto colocaba 62 kilogramos de

balanceado en toda la nave, los mismo 62 kilogramos fueron dosificados de forma uniforme en toda la nave y sobrando 136 gramos equivalente a más de una porción para un ave.

## **CONCLUSIONES**

Las visitas técnicas y lectura de diversas revistas avícolas beneficiaron a la ejecución de este proyecto debido a la forma en la que se alimenta a gallinas ponedoras en diversas granjas del cantón Salcedo, incentivando a buscar soluciones económicas y con incursión tecnológica para una mejor producción.

La construcción de la estructura dosificadora tal como se propuso en el capítulo 2 no afecto al espacio utilizado ya por los propietarios, cumpliendo así el propósito de dosificar a toda la nave del galpón Rosita, con servo motores que actuaron como apertura y cierre en el caudal de balanceado.

El control difuso propuesto cumplió con el propósito de controlar el flujo de balanceado dependiendo de la cantidad del mismo que se encuentre en el canal de comida de las aves, siendo este una alternativa viable al momento de optimizar recursos en una granja avícola, causando un ahorro por turno de aproximadamente 70 gramos y dosificando de manera uniforme.

El uso de Node-RED facilitó la comunicación con plataformas de IoT que en este caso fue resuelto con IBM cloud, siendo útil para el monitoreo desde un dispositivo inteligente de temperatura, humedad y niveles de gases que pueden afectar a la salud o productividad de las aves, además de poder ejecutar el control de dosificación de manera remota desde cualquier parte del mundo.

La tecnología usada en esta granja sin duda mejoro su manera de trabajo evitando esfuerzos físicos extremos o constantes, además siendo una parte fundamental el tener indicadores del estado ambiental de su galpón pudiendo deducir ciertas enfermedades en las aves y posibles causas dependiendo de los estándares de crianza del catálogo de las mismas.

### **RECOMENDACIONES**

Considerando la calidad del aire del galpón y las recomendaciones del catálogo de las aves, es factible colocar un sistema de ventilación de gases debido a que una alta presencia de gases pueda perjudicar a la producción de las aves y al estado de salud a largo plazo del trabajador.

La incorporación de más sensores en diversos puntos del galpón puede dar más información sobre el estado del galpón, para poder tomar las acciones pertinentes.

## BIBLIOGRAFÍA

- Arivubrakan, P., and K. Prema. 2020. "The Routing Based Protocol Technique for Enhancing the Performance Metrics Using MQTT in the Internet of Things." *Materials Today: Proceedings* (xxxx). doi: 10.1016/j.matpr.2020.11.070.
- Astill, Jake, Rozita A. Dara, Evan D. G. Fraser, Bruce Roberts, and Shayan Sharif. 2020. "Smart Poultry Management: Smart Sensors, Big Data, and the Internet of Things." *Computers and Electronics in Agriculture* 170(December 2019):105291. doi: 10.1016/j.compag.2020.105291.
- Balaji, R., A. V. R. Mayuri, N. Ramadevi, and R. Anirudh Reddy. 2020. "Advanced Implementation Patterns of Internet of Things with MQTT Providers in the Cutting Edge Communications." *Materials Today: Proceedings* (xxxx). doi: 10.1016/j.matpr.2020.11.090.
- Batuto, Adrian, Timothy Bryan Dejeron, Paolo Dela Cruz, and Mary Jane C. Samonte. 2020. "E-Poultry: An IoT Poultry Management System for Small Farms." *2020 IEEE 7th International Conference on Industrial Engineering and Applications, ICIEA 2020* 738–42. doi: 10.1109/ICIEA49774.2020.9102040.
- Chen, Fang, Linlin Qin, Xiaofeng Li, Gang Wu, and Chun Shi. 2017. "Design and Implementation of ZigBee Wireless Sensor and Control Network System in Greenhouse." *Chinese Control Conference, CCC* 8982–86. doi: 10.23919/ChiCC.2017.8028786.
- Cloud Foundry Foundation. 2021. "Conceptos de Cloud Foundry | Documentos de Cloud Foundry." Retrieved July 19, 2021 (<https://docs.cloudfoundry.org/concepts/index.html>).
- CONAVE. 2019. "Información Sector Avícola." *Conave.Org*. Retrieved

(<https://www.conave.org/informacion-sector-avicola-publico/>).

- Debauche, Olivier, Saïd Mahmoudi, Sidi Ahmed Mahmoudi, Pierre Manneback, Jérôme Bindelle, and Frédéric Lebeau. 2020. "Edge Computing and Artificial Intelligence for Real-Time Poultry Monitoring." *Procedia Computer Science* 175(2019):534–41. doi: 10.1016/j.procs.2020.07.076.
- Ghazal, Bilal, Khaled Al-Khatib, and Khaled Chahine. 2017. "A Poultry Farming Control System Using a ZigBee-Based Wireless Sensor Network." *International Journal of Control and Automation* 10(9):191–98. doi: 10.14257/ijca.2017.10.9.16.
- Gómez Giraldo, Nerley Adrián. 2021. "Diseño de Un Sistema de Monitoreo de Un Prototipo de Máquina Industrial, Para La Presentación de Datos En Tiempo Real Con Recursos de La Nube de IBM En Colombia." *Universidad Nacional Abierta Y A Distancia* 53(9):6.
- GONZALEZ CRUZ, CLAUDIA AIDE. 2012. "Diseño de Una Red de Sensores Para Control de Iluminación Basado En Lógica Difusa."
- Hooda, D. S., and Vivek. Raich. 2016. "Fuzzy Logic Models and Fuzzy Control : An Introduction." *Fuzzy Logic Models an Fuzzy Control* 374–85.
- IBM Cloud. 2021a. "Documentos de IBM Cloud." Retrieved July 9, 2021 (<https://cloud.ibm.com/docs/overview?topic=overview-what-is-platform>).
- IBM Cloud. 2021b. "IBM Watson IoT Platform." Retrieved July 9, 2021 (<https://www.ibm.com/docs/en/watson-iot-platform?topic=product-overview-features>).
- Islam, Md. Mahfujul, Shaharya Sourov Tonmoy, Sazzad Quayum, Al Russel Sarker, Sumaiya Umme Hani, and Mohammad Abdul Mannan. 2019. "Design and Implementation of Automated Poultry Farm with Distinguish Features." 273–76. doi: 10.1109/icrest.2019.8644464.
- Lu, Chien Yu, Fei Hsu Chen, Wen Chiung Hsu, Yu Qiang Yang, and Te Jen Su. 2020. "Constructing Home Monitoring System with Node-RED." *Sensors and Materials* 32(5):1701–10. doi: 10.18494/SAM.2020.2686.
- MARTINEZ, RODRIGO. 2017. "Comparativa y Estudio de Plataformas IoT."

*UNIVERSITAT POLITECNICA DE CATALUNYA* 87(1,2):149–200.

MEZA TTITO, EMILIO ENRIQUE. 2019. “SISTEMA DE CONTROL AUTOMÁTICO DE NIVEL DE AGUA EN LA CÁMARA DE CARGA BASADO EN LA LÓGICA DIFUSA PARA LA CENTRAL HIDROELÉCTRICA DE MACHUPICCHU.” *UNIVERSIDAD NACIONAL SAN ANTONIO ABAD DEL CUSCO*.

Mileva, Aleksandra, Aleksandar Velinov, Laura Hartmann, Steffen Wendzel, and Wojciech Mazurczyk. 2021. “Comprehensive Analysis of MQTT 5.0 Susceptibility to Network Covert Channels.” *Computers and Security* 104:102207. doi: 10.1016/j.cose.2021.102207.

Molloy, Derek. 2014. *Exploring BeagleBone*. 2nd, ilustra ed. edited by 2019 John Wiley & Sons.

Muñoz, Yeny Y., Omar D. Castrillón, Luis F. Castillo, and Carlos López. 2019. “Análisis de La Escena En La Cocina Por Medio de Sensores IoT Diseñados Basados En El Microcontrolador Node MCU ESP8266 y Conectados Al Servidor ThingSpeak.” *Información Tecnológica* 30(5):173–90. doi: 10.4067/s0718-07642019000500173.

Nicolas, Ron Daniel M., Wei S. Zhou, Shota C. Kitamura, and Mary Jane C. Samonte. 2019. “An IoT Monitoring Assistant for Chicken Layer Farms.” *ICTC 2019 - 10th International Conference on ICT Convergence: ICT Convergence Leading the Autonomous Future* 71–75. doi: 10.1109/ICTC46691.2019.8939845.

Node-RED, Js Foundation. 2021. “Node-RED.” Retrieved March 7, 2021 (<https://Node-RED.org/>).

Omomule, Taiwo Gabriel, Olusola Olajide Ajayi, and Adebola Okunola Orogun. 2020. “Fuzzy Prediction and Pattern Analysis of Poultry Egg Production.” *Computers and Electronics in Agriculture* 171(November 2019):105301. doi: 10.1016/j.compag.2020.105301.

Pereira, Wariston Fernando, Leonardo da Silva Fonseca, Fernando Ferrari Putti, Bruno César Góes, and Luciana de Paula Naves. 2020. “Environmental Monitoring in a Poultry Farm Using an Instrument Developed with the Internet

- of Things Concept.” *Computers and Electronics in Agriculture* 170(February):105257. doi: 10.1016/j.compag.2020.105257.
- Pillajo, Carlos, and Roberto Hincapie. 2018. *Wireless Network Control Systems De La Teoría a La Práctica*. Primera. Quito: Editorial Universitaria Abya-Yala.
- Pinzón, Antonio, Miguel Castillo, Edwin González, José Araúz, and Vladimir Villarreal. 2018. “Sistema de Detección de Sustancias y Partículas Contaminantes Para El Ambiente a Través de Sensores Arduino.” *Revista de Iniciación Científica* 4(1):55–59. doi: 10.33412/rev-ric.v4.1.1868.
- Ponce, Pedro. 2010. *Inteligencia Artificial Con Aplicaciones a La Ingeniería*.
- Rosales Tapia, Salome, and David Segovia. 2017. “Estudio de Mercado Avícola Enfocado a La Comercialización Del Pollo En Pie, Año 2012-2014.” 14(1):55–64.
- Sharp. 2002. “Datasheet GP2Y0A51SK0F.” 8.
- Wu, Fan, Christoph Rüdiger, Jean-Michel Redouté, and Mehmet Rasit Yuce. 2019. *A Wearable Multi-Sensor IoT Network System for Environmental Monitoring*. Springer International Publishing.
- Zhao, Zhihe, Jiaheng Wang, Chenxu Fu, Zhenbang Liu, Dawei Liu, and Bailiang Li. 2018. “Design of a Smart Sensor Network System for Real-Time Air Quality Monitoring on Green Roof.” *Journal of Sensors* 2018. doi: 10.1155/2018/1987931.



## ANEXOS

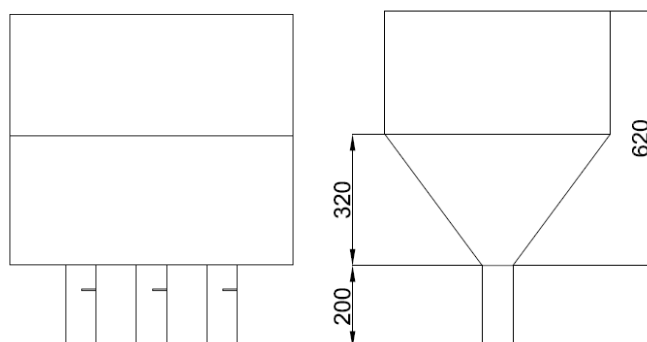
Anexo 1.

Página Web para ingreso a monitoreo y control



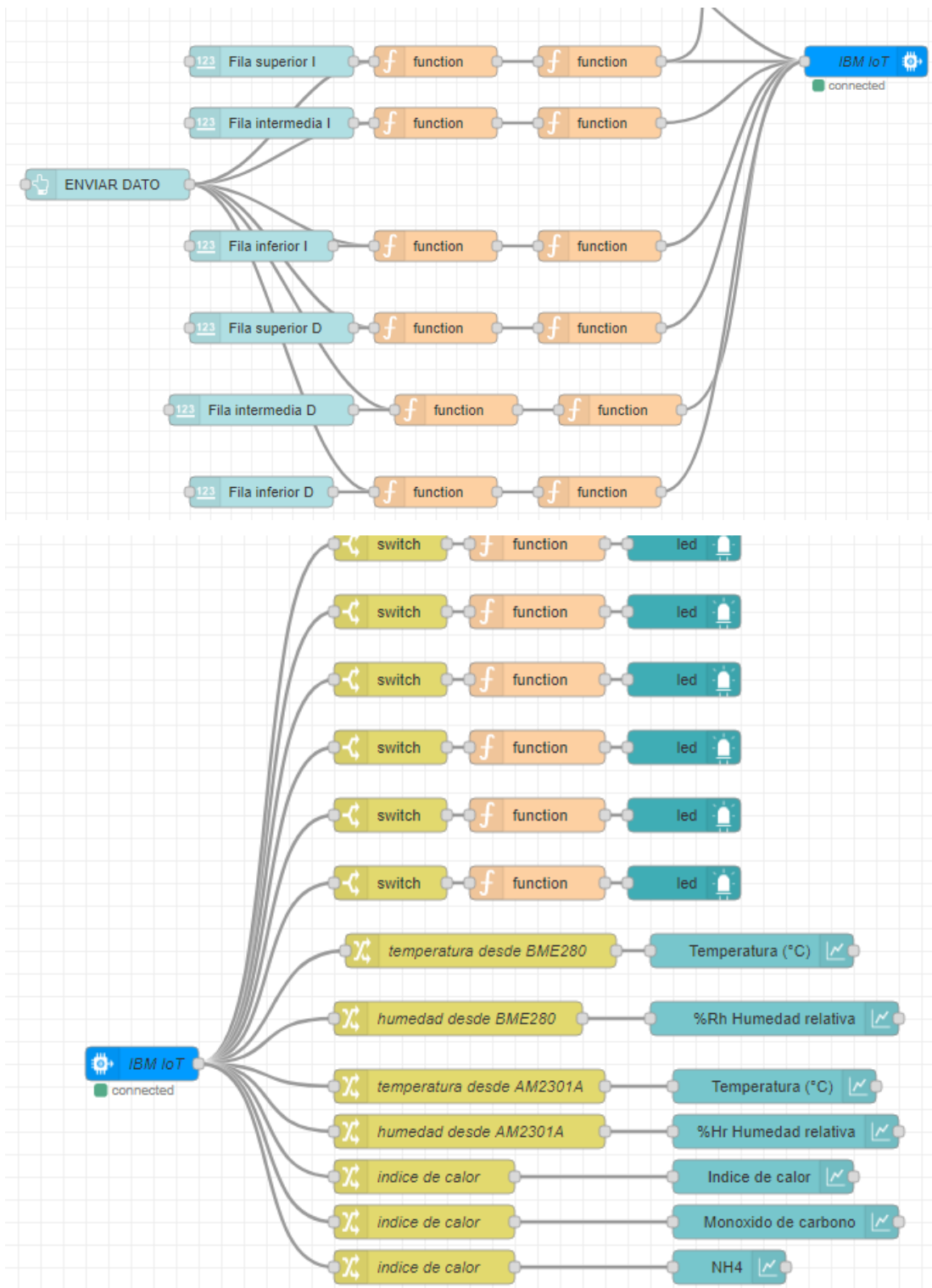
Anexo 2.

Medidas de tolvas dosificadoras



### Anexo 3.

#### Flujo en Node Red de IBM cloud.



## Anexo 4.

### Flujo en Node-RED local

