



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

**DISEÑO Y PROTOTIPO DE UNA RED DE SENSORES INALÁMBRICA PARA
AUTOMATIZACIÓN DE UN SISTEMA ACUAPÓNICO**

Maynor Giovanni Ballina Escobar

Asesorado por el Ing. Iván René Morales Argueta

Guatemala, marzo de 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO Y PROTOTIPO DE UNA RED DE SENSORES INALÁMBRICA PARA
AUTOMATIZACIÓN DE UN SISTEMA ACUAPÓNICO**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

MAYNOR GIOVANNI BALLINA ESCOBAR

ASESORADO POR EL ING. IVÁN RENÉ MORALES ARGUETA

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO ELECTRÓNICO

GUATEMALA, MARZO DE 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Luis Diego Aguilar Ralón
VOCAL V	Br. Christian Daniel Estrada Santizo
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympto Paiz Recinos
EXAMINADOR	Ing. Byron Odilio Arrivillaga Méndez
EXAMINADOR	Ing. Armando Alonzo Rivera Carrillo
EXAMINADOR(A)	Inga. Wendy Nora Miranda López
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO Y PROTOTIPO DE UNA RED DE SENSORES INALÁMBRICA PARA AUTOMATIZACIÓN DE UN SISTEMA ACUAPÓNICO

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 15 octubre del 2017.



Maynor Giovanni Ballina Escobar

Guatemala, 9 de noviembre de 2018

Ingeniero
Julio Solares Peñate
Coordinador Área de Electrónica
Escuela de ingeniería Mecánica Eléctrica
Facultad de Ingeniería
Universidad de San Carlos

Estimado ingeniero Solares:

Hago de su conocimiento por este medio que he concluido la revisión del trabajo de graduación del estudiante **Maynor Giovanni Ballina Escobar** titulado "**DISEÑO Y PROTOTIPO DE UNA RED DE SENSORES INALÁMBRICA PARA AUTOMATIZACIÓN DE UN SISTEMA ACUAPÓNICO**".

Indicando que el mismo cumple con los propósitos y objetivos con los que fue propuesto, por lo que, doy mi **APROBACIÓN** al mismo. Indicando además que, tanto el suscrito como el estudiante Ballina Escobar, somos los únicos responsables por el contenido del mismo.

Quedo en la mejor disposición de ampliar cualquier concepto relacionado con el trabajo en cuestión.

Reciba un cordial saludo,



Iván Rene Morales Argueta
ASESOR

Iván René Morales Argueta
Ingeniero Electrónico
Colegiado 12489



FACULTAD DE INGENIERIA

Guatemala, 22 de noviembre de 2018

Señor Director
Ing. Otto Fernando Andrino González
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

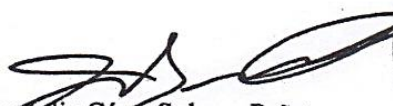
Señor Director:

Por este medio me permito dar aprobación al Trabajo de Graduación titulado **DISEÑO Y PROTOTIPO DE UNA RED DE SENSORES INALÁMBRICA PARA AUTOMATIZACIÓN DE UN SISTEMA ACUAPÓNICO**, desarrollado por el estudiante **Maynor Giovanni Ballina Escobar**, ya que considero que cumple con los requisitos establecidos.

Sin otro particular, aprovecho la oportunidad para saludarlo.

Atentamente,

ID Y ENSEÑAD A TODOS


Ing. Julio César Solares Peñate
Coordinador de Electrónica





REF. EIME 05. 2019.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto bueno del Coordinador de Área, al trabajo de Graduación del estudiante: **MAYNOR GIOVANNI BALLINA ESCOBAR** Titulado: **DISEÑO Y PROTOTIPO DE UNA RED DE SENSORES INALÁMBRICA PARA AUTOMATIZACIÓN DE UN SISTEMA ACUAPÓNICO**, procede a la autorización del mismo.


Ing. Otto Fernando Andriano González



GUATEMALA, 11 DE FEBRERO. 2019.

Universidad de San Carlos
de Guatemala



Facultad de Ingeniería
Decanato

DTG. 163.2019

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **DISEÑO Y PROTOTIPO DE UNA RED DE SENSORES INALÁMBRICA PARA AUTOMATIZACIÓN DE UN SISTEMA ACUAPÓNICO**, presentado por el estudiante universitario: **Maynor Giovanni Ballina Escobar**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

Ing. Pedro Antonio Aguilar Polanco
Decano



Guatemala, marzo de 2019

/gdech

ACTO QUE DEDICO A:

Mis padres	Mynor Ballina y Teresa Escobar. Por su apoyo incondicional y estar presentes siempre en mi vida.
Mi tía	Blanca Estela Ballina, por ser la segunda madre que siempre me apoyo y creyó en mí.
Mi esposa	Doris Roldán. Por ser mi acompañante de vida y el soporte en los momentos difíciles.
Mi hermano	Alex Ballina. Por su apoyo incondicional y motivarme a seguir adelante.
Mi tío	Salomón Ballina. Por ser una influencia positiva en mi vida.
Mi primo	Guillermo Ballina. Por ser mi influencia a nivel personal y profesional.
Amigo y mentor	Iván Morales. Porque sin él nada de esto fuera posible.
Ricardo Marroquín	Por sus consejos y apoyarme en los momentos difíciles de mi vida.

Todas esas personas

Quienes, a pesar de no ser mencionadas, han sido parte importante de mi vida personal y laboral, ustedes saben quiénes son.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por brindarme la oportunidad de educarme y trabajar.
Facultad de Ingeniería	Por brindarme el conocimiento académico y de vida.
Departamento de Física	Por darme motivación e influencia que marcó el rumbo de mi vida, permitirme desenvolver en el ámbito laboral y brindarme las mejores amistades, en especial al ingeniero Walter Álvarez.
Biblioteca Central	Por ser mi fuente de trabajo y darme la oportunidad de poder trabajar y estudiar al mismo tiempo.
Mis amigos de LAGO	Por apoyarme y brindarme excelentes experiencias de vida. En especial a Iván Morales, Eddy Solares, Guillermo García, Pablo Rosch y Rodrigo López.
Mis amigos	Porque las experiencias que he vivido con ustedes han sido importantes en mi vida. En especial a Sofía Menchú, Maco Sajquin, Jerson Culajay, Jorge Sen y Cristian Rosales.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
LISTA DE SÍMBOLOS	XI
GLOSARIO	XIII
RESUMEN	XV
OBJETIVOS	XVII
INTRODUCCIÓN	XIX
1. CONCEPTOS PRELIMINARES	1
1.1. Acuaponía	1
1.1.1. Sistema acuapónico	1
1.1.2. Elementos de un sistema acuapónico	2
1.1.3. Parámetros de un sistema acuapónico	3
1.2. Sensores	6
1.2.1. Tipos de sensores	6
1.2.1.1. Sensor de humedad	8
1.2.1.2. Sensor de temperatura	8
1.2.1.3. Sensor de pH	9
1.2.1.4. Sensor ultrasónico	10
1.2.1.5. Sensor de luminosidad	11
1.3. Red de sensores inalámbrica	11
1.3.1. Tipos de red de sensores	12
1.3.2. Topología de redes de sensores	13
1.4. MQTT	14
1.5. Buses y protocolos de comunicación serial	17
1.5.1. UART	18

1.5.2.	I ² C	19
1.5.3.	SPI	21
1.5.4.	1 Wire.....	22
1.6.	Microcontrolador	23
1.6.1.	ESP32S.....	24
1.7.	Sistema de control	27
1.7.1.	Sistema de control de lazo abierto y lazo cerrado...	28
1.7.2.	Tipos de control	30
2.	SELECCIÓN DE SENSORES	33
2.1.	Humedad del ambiente y temperatura ambiente	33
2.2.	Humedad del suelo	37
2.3.	Luminosidad.....	41
2.4.	Nivel de agua	44
2.5.	Temperatura del agua	46
2.6.	pH	48
2.7.	Caudal.....	54
3.	DISEÑO DEL SISTEMA ACUAPÓNICO.....	57
3.1.	Estanque.....	58
3.1.1.	Sensores y actuadores del estanque.....	60
3.2.	Cama de sembrado.....	66
3.2.1.	Sensores de la cama de sembrado	68
3.3.	Módulo de control y almacenamiento	73
4.	DISEÑO DEL SISTEMA DE COMUNICACIÓN.....	77
4.1.	Mensaje MQTT	79
5.	RECOLECCIÓN DE DATOS Y CONTROL DE LOS NODOS	87

5.1.	Nodo plantas	87
5.1.1.	Monitoreo constante	87
5.1.2.	Llenando	90
5.2.	Nodo tanque	91
5.2.1.	Monitoreo constante	92
5.2.2.	Llenando	93
5.3.	Broker.....	95
5.3.1.	Monitoreo constante	97
5.3.2.	Llenando	98
6.	PROTOTIPO DEL SISTEMA ACUAPÓNICO	101
6.1.	Contenedores e inmobiliario	103
6.1.1.	Mesa de soporte	103
6.1.2.	Tanque.....	104
6.1.3.	Camas de sembrado.....	106
6.1.4.	Bomba de agua.....	109
6.2.	Módulos de sensores.....	111
6.3.	Notificaciones y monitoreo.....	120
6.4.	Calibración y pruebas	123
	CONCLUSIONES	129
	RECOMENDACIONES.....	131
	BIBLIOGRAFÍA.....	133
	APÉNDICES.....	139
	ANEXOS	173

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Sistema acuapónico	2
2.	Escala de pH.....	5
3.	Temperatura vs oxígeno disuelto	5
4.	Topologías de redes de sensores	14
5.	Arquitectura MQTT.....	15
6.	MQTT topics	16
7.	Trama de datos UART	19
8.	Bus I ² C.....	20
9.	Condiciones del bus I ² C	21
10.	Comunicación de bus SPI	22
11.	Bus 1-Wire	23
12.	ESP32.....	24
13.	Diagrama de bloques	25
14.	Sistemas de control de lazo abierto y cerrado	29
15.	Sistemas de control digital	29
16.	DHT11	34
17.	Gráfica de humedad y temperatura variante	36
18.	Gráfica de humedad y temperatura, realizando mediciones cada 30 minutos en un período de 24 horas	37
19.	YL-69 y YL-38	38
20.	Gráfica de humedad del suelo en tierra.....	39
21.	Gráfica de humedad del suelo en grava.....	40
22.	GY-30	42

23.	Respuesta al espectro de luz.....	43
24.	Gráfica de luxes en un período de 24 horas	43
25.	HC-SR04	44
26.	Operación HC-SR04.....	45
27.	Gráfica del comportamiento de la altura con respecto al tiempo de llenado y vaciado del contenedor.....	46
28.	DS18B20	47
29.	Gráfica de temperatura del agua en condiciones variables.....	48
30.	Impacto del pH en los nutrientes.....	49
31.	Sensor de pH SEN0161.....	50
32.	Gráfica de medidor de pH electrónico vs EasyStrips	52
33.	Tabla de pH de elementos del hogar	53
34.	Gráfica de pH de elementos caseros	53
35.	Sensor FS3000A	54
36.	Comportamiento dinámico del sensor FS300A.....	55
37.	Propuesta del sistema acuapónico	57
38.	Estanque, mediciones en cm	58
39.	Propuesta del montaje del tubo PVC en la cama de sembrado	59
40.	Diagrama de bloques para el nodo de sensores del tanque	60
41.	Bomba de agua	61
42.	Esquema del circuito para el nodo de sensores del tanque	62
43.	Esquema del PCB para el nodo de sensores del tanque	63
44.	Caja de nodo de tanque, mediciones en mm	64
45.	Diagrama de flujo del nodo del tanque.....	65
46.	Cama para sembrado, mediciones en cm.....	66
47.	Propuesta de montaje del tubo PVC en la cama de sembrado	67
48.	Diagrama de bloques para el nodo de sensores para plantas	68
49.	Esquema del circuito para el nodo de sensores para plantas	69
50.	Esquema del PCB para el nodo de sensores para plantas	70

51.	Caja de nodo de plantas, mediciones en mm	71
52.	Diagrama de flujo del nodo de sembrado	72
53.	Raspberry Pi Zero W	73
54.	Diagrama de entidades	74
55.	Diagrama de flujo del nodo de control	75
56.	Diagrama de conexión hacia el <i>router</i>	77
57.	Diagrama de conexión MQTT, de <i>broker</i> a clientes	78
58.	Diagrama de <i>topics</i> utilizados.....	79
59.	Mensaje en json del <i>broker</i> para todos los estados	81
60.	Mensaje en json del control contante del tanque	82
61.	Mensaje en json del llenado desde el tanque	83
62.	Mensaje en json del control contante del nodo plantas	84
63.	Mensaje en json del llenado del nodo plantas	85
64.	Diagrama de flujo de monitoreo constante	88
65.	Diagrama de flujo del modo llenando	90
66.	Diagrama de flujo de monitoreo constante	92
67.	Diagrama de flujo de llenado.....	94
68.	Diagrama de control de humedad	95
69.	Diagrama de control de pH	96
70.	Diagrama de flujo, control de monitoreo constante.....	97
71.	Diagrama de flujo, control de modo llenando	99
72.	Plano de la mesa en cm.....	103
73.	Mesa de soporte	103
74.	Tanque vacío	104
75.	Tanque con separador	104
76.	Altura inicial del agua	105
77.	Tilapias para el sistema acuapónico.....	105
78.	Contenedor vacío.....	106
79.	Separador y tanque con separador	107

80.	Dos contenedores instalados en la mesa de soporte.....	107
81.	Plano del drenaje en mm	108
82.	Contenedor con reductor y drenaje instalado.....	108
83.	Contenedores con piedra pómez	109
84.	Plano de tubería de distribución en mm.....	110
85.	Tubería instalada con sensor de caudal	110
86.	Tubería y bomba instalada	111
87.	Contenedores impresos.....	112
88.	Placa de circuito impreso de tanque y plantas	113
89.	Placas con componentes.....	114
90.	Instalación de las placas de los módulos	115
91.	Módulo plantas terminado e instalado.....	116
92.	Tubo con puntas de prueba	117
93.	Módulo tanque terminado e instalado	117
94.	Salida del agua por medio de los tubos	118
95.	Caída del agua por medio del drenaje	118
96.	Sistema acuapónico completo	119
97.	Raspberry Pi Zero W, instalada	120
98.	Flujo de Node Red.....	121
99.	Página web de monitoreo	121
100.	Base de datos actualizable en tiempo real.....	122
101.	Correo enviado por el sistema	122
102.	Calibración por medio del módulo pH	123
103.	Soluciones de pH para calibración.....	124
104.	Resultado de las láminas de prueba pH 4 y pH 7, respectivamente ..	124
105.	Medición de tiempos en el sistema	125
106.	Gráfica de reacción de nodos, llenado y vaciado	126
107.	Nodo plantas 10	127
108.	Nodo plantas 11	127

109.	Nodo tanque 11.....	128
------	---------------------	-----

TABLAS

I.	Rangos generales de tolerancia de calidad de agua.....	3
II.	Especificaciones técnicas	35
III.	Milivoltios vs pH	51
IV.	Lista de materiales varios.....	101
V.	Lista de materiales electrónicos	102

LISTA DE SÍMBOLOS

Símbolo	Significado
ABS	<i>Acrylonitrile Butadiene Styrene</i>
BNC	<i>Bayonet Neill-Concelman</i>
cm	Centímetro
dBm	Decibelio-milivatio
RTD	Detector de temperatura por resistencia
GPIO	<i>General Purpose Input/Output</i>
GHz	Gigahercio
°C	Grados centígrados
H	Hidrógeno
HR	Humedad relative
ISO	International Organization for Standardization
IoT	<i>Internet of things</i>
IP	<i>Internet protocol</i>
Json	<i>JavaScript Object Notation</i>
kHz	Kilohercio
KΩ	Kilo ohmios
LDR	<i>Light Dependent Resistance</i>
l/h	Litros por hora
l/min	Litros por minuto
Mbps	Megabits por segundo
MHz	Megahercios
MPa	Megapascal
MQTT	<i>Message Queue Telemetry Transport</i>

mA	Miliamperio
mAh	Miliamperio-hora
μA	Microamperio
μF	Microfaradio
mm	Milímetro
Ms	Milisegundo
nA	Nanoamperio
nF	Nanofaradios
NAT	Nitrógeno amoniacal total
NPS	Nominal Pipe Size
OD	Oxígeno disuelto
PVC	Policloruro de vinilo
CPVC	Policloruro de vinilo clorado
pH	Potencial de hidrógeno
PH2O	Presión parcial del vapor de agua
QoS	<i>Quality of service</i>
RTC	<i>Real Time Clock</i>
R	Resistencia
s	Segundos
SSL	<i>Secure Sockets Layer</i>
SCL	<i>Serial Clock Line</i>
SDA	<i>Serial Data Line</i>
T	Temperatura
TCP	<i>Transmission-Control-Protocol</i>
VAC	Voltaje Corriente Alterna
VDC	Voltaje Corriente Directa
V	Voltio
W	Watts
WSN	<i>Wireless Sensor Networks</i>

GLOSARIO

Bluetooth	Es una especificación industrial para redes inalámbricas de área personal creado por Bluetooth Special Interest Group, Inc., que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2.4 GHz.
Broker	En el protocolo de comunicación MQTT es el nodo central que contiene una capacidad de trabajar y controlar un gran número de clientes.
Gateway	Puerta de enlace que es el dispositivo que actúa de interfaz de conexión entre aparatos o dispositivos.
Nitrobacter	Es un género de bacterias gram negativas, son quimios autotróficas que participan activamente en el ciclo del nitrógeno.
Open source	Es un modelo de desarrollo de software basado en la colaboración abierta.
Ping	Es una utilidad diagnóstica en redes de computadoras que comprueba el estado de la comunicación del <i>host</i> local con uno o varios equipos remotos de una red IP.

Watchdog Timer

Es un mecanismo de seguridad que provoca un *reset* del sistema en caso de que este se haya bloqueado.

Wifi

Es una tecnología que permite la interconexión inalámbrica de dispositivos electrónicos.

RESUMEN

Este documento recopila la propuesta de automatización para mantener el control constante de un sistema acuapónico, implementando Internet de las cosas para resolver los problemas de conectividad, comunicación y obtener los parámetros necesarios para ver el comportamiento de un sistema acuapónico a pequeña escala.

En el primer capítulo se desarrollan los fundamentos teóricos necesarios para implementar un sistema acuapónico y determinar los elementos necesarios para poder automatizar el mismo. Inicia con la descripción de un sistema acuapónico y se enlista los elementos y conceptos técnicos necesarios para el desarrollo del diseño electrónico.

En el segundo capítulo se realiza la selección de sensores a utilizar para diseñar el prototipo, enlistando las características básicas por las cuales se elige cada uno de los sensores, al final de la descripción se realizan pruebas individuales a cada elemento.

En el capítulo tres se realiza el diseño básico del prototipo, eso incluye el mobiliario donde se instalará el sistema acuapónico, además del diseño de los módulos de sensores y los componentes de control para cada etapa de este sistema.

El capítulo cuatro explica el formato de los mensajes y los elementos básicos que estos manejan para la interpretación del *broker*. El capítulo cinco continúa con los elementos básicos de control para determinar los tipos de

mensaje, además de explicar las etapas que tiene el sistema acuapónico y la forma de recolección de datos de cada nodo.

El capítulo seis muestra de forma visual la implementación del prototipo que controla el sistema acuapónico, explicando las etapas de construcción del sistema y los módulos, además se muestra de forma gráfica el comportamiento del sistema conforme su funcionamiento.

OBJETIVOS

General

Diseñar una red de sensores inalámbrica para la automatización de un sistema acuapónico.

Específicos

1. Diseñar estaciones modulares de sensores controlados por medio de un microcontrolador, el cual se conectará de forma inalámbrica a un nodo central.
2. Utilizar diversos sensores que muestren parámetros importantes dentro del sistema acuapónico.
3. Diseñar un sistema inalámbrico basado en el protocolo MQTT.
4. Crear un prototipo funcional para la calibración de los sensores y del control del sistema acuapónico.
5. Crear un sistema inalámbrico que esté alimentado por medio de baterías.

INTRODUCCIÓN

Las redes inalámbricas de sensores han crecido mucho con el pasar de los años. Estas se implementan en infraestructuras existentes, complementando su funcionamiento con protocolos de comunicación de bajo consumo de ancho de banda, entre estos el protocolo MQTT.

Existen diversos sistemas que se pueden optimizar o automatizar con esta tecnología, por lo general se busca obtener información de diversos sensores y publicarlos, ya sea en una página web o almacenarlos en una base para después tomar decisiones o analizar el comportamiento del sistema monitoreado por la red de sensores inalámbrica.

El sistema acuapónico es un sistema de lazo cerrado que necesita monitoreo constante y tomar decisiones con base en parámetros que se deben medir periódicamente, en este caso se busca automatizar este sistema, implementado diversos sensores que verifiquen aspectos físicos como pH, temperatura, humedad, luminosidad y distancia, para después, con estos datos, mantener el equilibrio en el sistema acuapónico por medio de un actuador, creando un sistema de control de lazo cerrado.

Se implementa el Internet de las cosas para automatizar el sistema acuapónico, creando una propuesta modular que se alimenta por medio de baterías y de fácil instalación.

1. CONCEPTOS PRELIMINARES

En la siguiente sección se determinan los conceptos principales con relación a las bases de investigación.

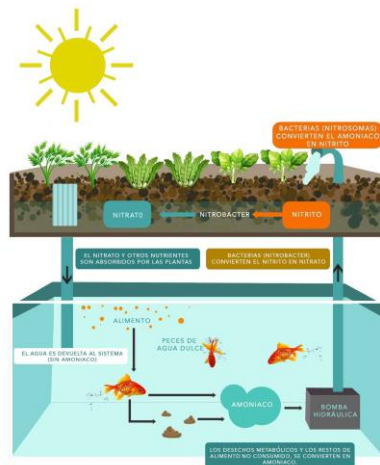
1.1. Acuaponía

La acuaponía es un sistema de sembrado que une la acuicultura y la hidroponía. Por separado, la acuicultura es la crianza de peces, cangrejos o camarones en un medio controlado; la hidroponía es el cultivo de plantas, utilizando agua en lugar de tierra, la cual se lleva a cabo en áreas urbanas o en espacios reducidos para el cultivo. Es un medio de producción sostenible de plantas y peces.

1.1.1. Sistema acuapónico

Este es un sistema cerrado que se autoalimenta de un microsistema a otro: las secreciones de los peces se acumulan en el agua (aumentando su toxicidad), lo cual es aprovechado por las plantas, dado que los desechos de los peces son descompuestos en nitritos, estos, al llevarse al sistema hidropónico, las bacterias de nitrificación los convierten en nitratos, los cuales a su vez son alimento para las plantas que los absorben por medio de sus raíces, funcionando como un biofiltro; al momento que las plantas consumen los nitratos reducen los niveles de oxidación del agua, la cual finalmente regresa hacia los peces, equilibrando este pequeño ecosistema y convirtiéndose en un sistema de lazo cerrado que maneja la recirculación del agua para reutilizarla en ambos microsistemas.

Figura 1. Sistema acuapónico



Fuente: Universidad Andrés Bello. *Explicación de sistema de acuaponía.*

<http://sustentabilidad.unab.cl/wp-content/uploads/2017/04/explicación-sistema-acuaponía.jpg>.

Consulta: 3 de agosto de 2017.

1.1.2. Elementos de un sistema acuapónico

Los sistemas acuapónicos se basan en 3 diseños básicos: técnica del *film* nutritivo, raíz flotante y camas de siembra. La variedad de diseños influye en el modo de sembrado. La técnica de *film* nutritivo se basa en una siembra sobre tubos por lo general de PVC, por los cuales corre el agua que contiene los nutrientes. El diseño de raíz flotante se basa en la siembra de balsas flotantes, estas se posicionan sobre tanques con una gran cantidad de agua para que contengan un nivel alto de profundidad. Con el diseño de lechos de sustratos se crea una variante dado que el recipiente de siembra se rellena con un tipo de material inerte.

Otro elemento importante en el sistema acuapónico es el tanque para el alojamiento de los peces, este debe tener mayor longitud que altura, para que los

peces no sufran de estrés. Estos tanques deben ser claros para poder mantener el monitoreo de las especies y su comportamiento ante cambios en su entorno.

1.1.3. Parámetros de un sistema acuapónico

Para que el sistema acuapónico trabaje de forma adecuada debe generarse un proceso llamado biofiltración, el cual se encarga de transformar el nitrógeno excretado por los organismos vivos, creando la transición total desde amoníaco a nitrato. “Para poder determinar la toxicidad del agua se debe contar con las mediciones de temperatura y pH, dado que estas están sujetas al nitrógeno amoniacal total (NAT), para que peces puedan vivir en un entorno el nivel de amoníaco existente debe ser menor a 1 mg/l.”¹

Los 5 parámetros principales que definen la calidad del agua son: temperatura, oxígeno disuelto, potencial de hidrógeno, compuestos nitrogenados y alcalinidad. Esta calidad debe ser lo más cercana a la siguiente tabla:

Tabla I. Rangos generales de tolerancia de calidad de agua

Tipo de organismo	Temperatura (°C)	pH	NAT (mg/l)	NO2 (mg/l)	NO3 (mg/l)	OD (mg/l)
Peces aguas cálidas	22-32	6-8,5	< 3	< 1	< 400	4-6
Peces aguas frías	10-18	6-8,5	< 1	< 0,1	< 400	6-8
Plantas	16-30	5,5-7,5	< 30	< 1	-	> 3
Bacterias nitrificantes	14-34	6-8,5	< 3	< 1	-	4-8

Fuente: SOMERVILLE, Christopher; et al. *Small-scale aquaponic food production – Integrated fish and plant farming*. p. 22.

¹ MASSER, Michael; et al. *Recirculating aquaculture tank production systems: management of recirculating systems*. Southern Regional Aquaculture Center Publication N° 452.

Para el sistema acuapónico los parámetros en los que se debe mantener un estricto balance y monitoreo son: la temperatura y el potencial de hidrógeno (pH), y la FAO en su guía *Small-scale aquaponic food production* recomienda:

“Los niveles recomendados para el sistema acuapónico contemplando los tres organismos.

- Temperatura: 18-30 grados Celsius
- pH: 6-7
- Nitrógeno amoniacal total: < 1 miligramo por litro
- NO₂: < 1 miligramo por litro
- NO₃: 5-500 miligramos por litro
- Oxígeno disuelto: > 5 miligramos por litro²

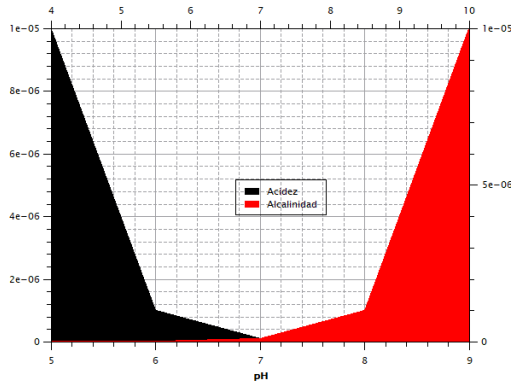
El oxígeno disuelto (OD) es importante de monitorear dado que las plantas, los peces y las bacterias del sistema acuapónico necesitan oxígeno para vivir, este parámetro debe estar entre el rango de 5 a 8 miligramos por litro, un factor importante del oxígeno disuelto es que este decrece conforme la temperatura aumenta.

El potencial de hidrógeno (pH) determina la acidez o alcalinidad de una solución, esta se encuentra en una escala logarítmica de 1 a 14, en 7 el elemento es neutral, al ser menor de 7 este es ácido, al contrario, es alcalino. Para el sistema acuapónico el rango óptimo debe estar entre 6 a 7. Esto define cuántos iones de hidrógeno (H⁺) activos existen en la solución, dado por la ecuación.

$$pH = -\log_{10}[a_{H^+}] \quad (1.1)$$

² SOMERVILLE, Christopher; et al. *Small-scale aquaponic food production. Integrated fish and plant farming*. p. 22.

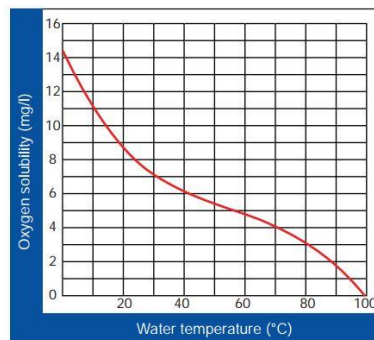
Figura 2. **Escala de pH**



Fuente: elaboración propia, empleando Qtiplot.

La temperatura puede tener un rango entre 18 a 30 grados Celsius, esta dependerá del tipo de planta y pez seleccionado para el sistema acuapónico, dado el entorno natural de los especímenes seleccionados, sin embargo, las altas temperaturas afectan a los valores de oxígeno disuelto, y el nivel de absorción de calcio en las plantas.

Figura 3. **Temperatura vs oxígeno disuelto**



Fuente: SOMERVILLE, Christopher; et al. *Small-scale aquaponic food production. Integrated fish and plant farming.* p. 23.

El nitrato inorgánico (NO_3) se forma naturalmente debido a la descomposición de proteínas, urea, entre otros. Cabe resaltar que en esta descomposición se forma amoníaco, el cual por medio de *Nitrobacter* se transforma en ácido nítrico. Los nitratos son parte esencial del compuesto del abono, estos son aprovechados por las plantas para su alimento.

1.2. Sensores

Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, para transformarlas en señales eléctricas que se puedan cuantificar o manipular. Las variables de instrumentación pueden ser, por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, pH, entre otros. Estos se realizan por medio de componentes pasivos y activos, para obtener magnitudes eléctricas, las cuales pueden ser una resistencia eléctrica, una capacitancia eléctrica, una tensión eléctrica, una corriente eléctrica, entre otros. A diferencia de un transductor, el sensor puede adaptar la señal que está midiendo por medio de circuitos electrónicos, para que un dispositivo de control la pueda interpretar sin necesidad de un circuito de acople y transforme la señal.

1.2.1. Tipos de sensores

La funcionalidad y características de un sensor dependerán de los siguientes parámetros básicos:

- Exactitud: es el grado de conformidad del valor medido al valor real.
- Precisión: es la constancia en el valor medido dado que la dispersión en los valores obtenidos en una serie de mediciones será mínima.

- Rango de funcionamiento: los valores de entrada en los cuales puede realizarse la medición deben ser precisos y exactos en todo rango de medición.
- Fiabilidad: el sensor no debe estar sujeto a fallos frecuentes durante su funcionamiento.
- Sensibilidad: la relación que indica cuánta salida se obtiene por unidad de entrada.
- Distancia operativa: qué tan distante debe estar el sensor al elemento a medir para su funcionamiento óptimo.
- Histéresis: es la diferencia máxima que se observa en los valores indicados por la señal de salida con base en incrementos o decrementos continuos.
- No linealidad: suponer que la relación entre la entrada y la salida de todos los sensores es lineal.

Se pueden definir dos tipos de sensores según la energía empleada en ellos o proporcionada por ellos:

- Activos: emiten energía a partir de la transformación realizada
- Pasivos: reciben energía para realizar su transformación

Estos se pueden clasificar por medio de su principio de funcionamiento, tales como los sensores primarios, resistivos, de reactancia variable y electromagnética, generadores y digitales.

Otro tipo de clasificación de sensores es por medio de las magnitudes que medirán, está relacionada directamente al fenómeno físico o químico que desea medir. Entre estos aspectos están posición, desplazamiento, velocidad, aceleración, fuerza, nivel, presión, caudal, temperatura, proximidad, entre otros.

1.2.1.1. Sensor de humedad

Los sensores de humedad toman su medición con base en la humedad relativa que se encuentre en el aire. La medición se realiza con base en el vapor de agua que se encuentre en una medición, siendo 0 % de humedad relativa perteneciente al aire seco y 100 % de humedad relativa perteneciente a un ambiente completamente húmedo. Este parámetro se basa en la presión parcial del vapor de agua en mezcla con el aire ($P_{(H_2O)}$) y la presión de saturación de agua a la temperatura de aire ($P^+_{(H_2O)}$), representado en la ecuación:

$$HR = \frac{P_{(H_2O)}}{P^+_{(H_2O)}} * 100 \% \quad (1.2)$$

Los más utilizados son los sensores de polímero resistivo, los cuales se fabrican en una base de polímero que es capaz de absorber agua, para obtener el valor requerido se debe medir la resistencia a través del polímero.

1.2.1.2. Sensor de temperatura

Los cambios de temperatura pueden ser determinados por expansión o contracción de sólidos, líquidos o gases, el cambio de resistencia eléctrica y fem termoeléctrica.

En los detectores de temperatura por resistencia (RTD) la temperatura de ciertos materiales aumenta de forma lineal conforme al aumento de la temperatura, el tiempo de respuesta de estos sensores es del orden de 0,5 a 5 segundos o mayores, el cambio de resistencia del material está basado en la resistencia que tiene un elemento a 0 °C R_0 y el coeficiente de temperatura de la resistencia del material α determinado por la ecuación.

$$R_t = R_0(1 + \alpha\Delta t) \quad (1.3)$$

Los termistores también funcionan como transductores para los sensores de temperatura, estos elementos están hechos de mezclas óxidometálicas, el comportamiento de la resistencia no es lineal al aumento de la temperatura. Estos componentes tienen dos coeficientes de temperatura: negativo (NTC) y positivo (PTC). La relación de la resistencia con respecto a la temperatura depende de la constante K y la resistencia del material β , lo cual se puede expresar con la siguiente ecuación:

$$R_t = K e^{\frac{\beta}{t}} \quad (1.4)$$

Donde:

$$K = R_0 e^{\frac{-\beta}{t_0}} \quad (1.5)$$

Entre otro tipo de transductores que se utilizan están los termodiodos y los termotransistores, los cuales, al cambiar la temperatura del semiconductor con impurezas, también cambia la movilidad de los portadores de carga. Eso quiere decir que la corriente que circula estará basada en la temperatura.

1.2.1.3. Sensor de pH

Este sensor utiliza un método electroquímico para medir los iones de hidrógeno potenciales (pH) en una disolución, se basa en dos electrodos que están sumergidos en dos soluciones de electrolito, mide la diferencia de potencial entre ambos electrodos con base en la ecuación de Nerst para la medición del potencial reducido:

$$E = E_0 + \frac{R*T}{n*F} \log(C_{H^+}) \quad (1.6)$$

Donde E es el potencial medido, E₀ es el potencial en condiciones estándar, R la constante de los gases ideales, T temperatura en Kelvin, n la cantidad de mol de electrones participantes en la reacción, F la constante de Faraday y C_{H⁺} la cantidad de iones de hidrógeno en la solución. También se puede representar como:

$$E = E_0 - \frac{R*T}{n*F} * pH \quad (1.6)$$

Por lo tanto, la diferencia de potencial solo dependerá de las variaciones de pH del medio externo al sensor.

1.2.1.4. Sensor ultrasónico

Los sensores de proximidad ultrasónicos usan un transductor para enviar y recibir señales de sonido de alta frecuencia, cuando un objeto intercepta la onda de sonido, el sonido es reflejado al sensor haciendo que este pueda determinar la diferencia de tiempo en que la señal fue emitida y recibida. La distancia se determina con base en el tiempo T y la velocidad del sonido c_s como lo expone la siguiente ecuación:

$$L = \frac{1}{2} T * c_s \quad (1.7)$$

Entre las ventajas de los sensores ultrasónicos se tiene que, debido a que las ondas de sonido se pueden reflejar, estas pueden detectar objetos transparentes y su medición no se ve afectada por neblina o suciedad.

1.2.1.5. Sensor de luminosidad

Los cambios en la radiación luminosa se pueden obtener por medio de materiales semiconductores como los fotodiodos, fototransistor y resistencia LDR (Light Dependent Resistance). Los fotodiodos se conectan en un circuito de polarización inversa, lo cual permite que su resistencia sea muy elevada, la incidencia de luz en la unión del diodo permite un aumento en la corriente inversa, por lo cual la resistencia del diodo decrece, esta corriente inversa es casi proporcional a la luz. En los fototransistores la luz genera una corriente en la base del transistor directamente proporcional a la intensidad luminosa. Por último, los fotorresistores, al aumentar la intensidad luminosa, la resistencia del elemento decrece de manera lineal.

1.3. Red de sensores inalámbrica

“Una red de sensores (WSN) es un grupo de dispositivos autónomos, compuestos de sensores que monitorean condiciones físicas o ambientales”³, para realizar una tarea en común. Esta se basa en dispositivos de bajo costo y bajo consumo eléctrico llamados nodos, los cuales son capaces de obtener información de su entorno, procesarla localmente y comunicarla a través de enlaces inalámbricos hasta un nodo central de coordinación.

En algunas de ellas, los nodos actúan como elementos de la infraestructura de comunicaciones al reenviar los mensajes transmitidos por nodos más lejanos hacia al centro de coordinación. La red de sensores inalámbricos está formada por numerosos dispositivos distribuidos espacialmente, que utilizan sensores para controlar diversas condiciones en distintos puntos. Los sensores pueden ser

³ National Instruments. *¿Qué es una red de sensores inalámbricos?* <http://www.ni.com/white-paper/7142/es/>. Consulta: 3 de febrero de 2018.

fijos o móviles. Por lo general cada unidad de este sistema cuenta con un microcontrolador, una fuente de energía, un receptor de radiofrecuencia y los sensores. Los transceptores deben cumplir el estándar IEEE1451.5, la información va dirigida a una estación base, la cual se convierte en un Gateway, dado que este permite la interconexión entre la red de sensores y la red TCP/IP.

Entre las características básicas de una red de sensores están: el tiempo de vida, cobertura de la red, costo y facilidad de instalación, tiempo de respuesta, precisión y frecuencia de las mediciones, seguridad y consumo energético.

1.3.1. Tipos de red de sensores

Dependiendo del entorno en el cual se aplicará la red de sensores, esta puede tener diferentes tipos:

- Terrestres: en este tipo de redes la comunicación hacia la estación base es muy eficiente, consiste de varios nodos colocados en un área específica.
- Subterráneas: este tipo de redes es más costosa que las terrestres, en términos de mantenimiento, equipo y creación. Varias de estas estaciones se encuentran debajo de la tierra, pero algunas se encuentran sobre ella, las condiciones de comunicación son difíciles dado a los altos niveles de atenuación y pérdida de señal.
- Bajo de agua: se basan en sensores o vehículos dispersos bajo de agua, la mayoría de ellos son autónomos y se encargan de recolectar información. Las grandes desventajas son: el ancho de banda, larga demora en la propagación de la onda de comunicación y fallas en los sensores, muchos de estos dispositivos están equipados con baterías de vida útil limitada.

- Multimedia: son redes que están equipadas para monitorear eventos tales como imágenes y sonidos. Las desventajas de este tipo de redes son el alto consumo de energía y la necesidad de un alto ancho de banda.
- Móviles: estas se pueden mover e interactuar con su entorno físico, tienen la habilidad de comunicarse de una forma versátil, pueden tener mejoras en la eficiencia energética y mayor capacidad de canales de comunicación.

1.3.2. Topología de redes de sensores

Para las comunicaciones por radiofrecuencia las topologías más utilizadas en este tipo de redes son estrella (*star*), jerarquía de árbol (*cluster tree*) y malla (*mesh*).

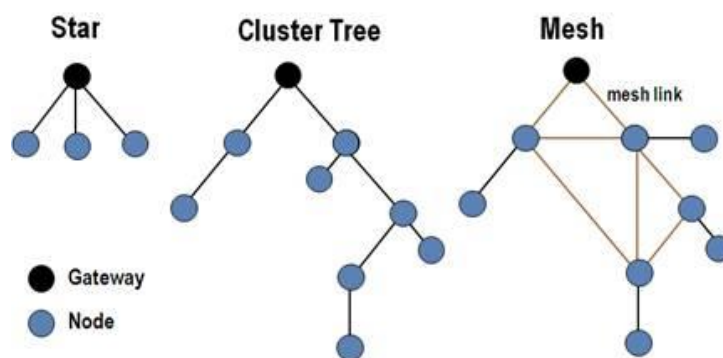
En la topología de tipo estrella cada nodo de la red se conecta a una sola puerta de acceso denominada estación base, este tipo de configuración no permite que los nodos se comuniquen entre sí, más bien la estación base maneja la comunicación para cada nodo, lo cual permite tener comunicaciones con baja latencia, pero debe estar dentro del área de transmisión de cada uno de los nodos.

La topología de jerarquía de árbol, también conocida como estrella en cascada, permite que nodos se conecten a otros nodos que están arriba dentro de la organización de árbol y estos a su vez se conectan con la estación base, esto permite que la distancia física de los nodos sea mucho mayor y la detección de errores sea fácil de obtener.

La topología del tipo malla permite la transmisión de datos entre nodos que estén dentro de su rango de alcance, si un nodo se quiere comunicar con otro

que esté lejos de su alcance, este debe comunicarse por medio de un nodo intermedio, las ventajas de este tipo de topología permiten: instalación a grandes distancias, aislamiento entre nodos y fácil detección de errores. La clara desventaja es el costo económico de esta red.

Figura 4. Topologías de redes de sensores



Fuente: National Instruments. ¿Qué es una red de sensores inalámbricos?
<http://www.ni.com/white-paper/7142/es/>. Consulta: 29 de mayo de 2018.

1.4. MQTT

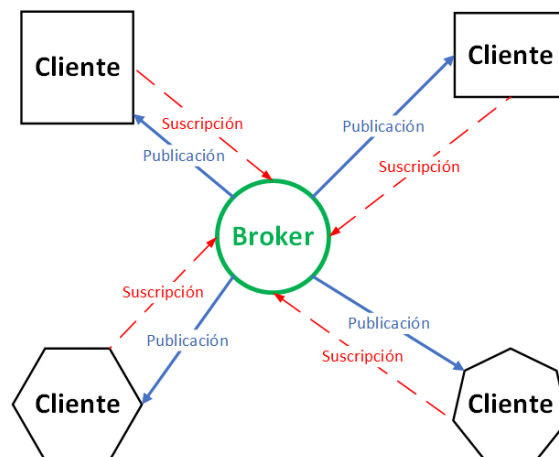
El protocolo de comunicación MQTT (Message Queue Telemetry Transport) realiza comunicación *machine-to-machine* (M2M) enfocada para el Internet de las cosas. “Está diseñado para que transporte mensajes de publicación y suscripción extremadamente ligeros limitada a un máximo de 256 MB de información, el cual trabaja sobre el protocolo TCP/IP utilizando los puertos 1883 para comunicación no encriptada y 8883 para comunicación encriptada utilizando SSL/TLS.”⁴ Es útil para conexiones con ubicaciones remotas en las que se requiere una pequeña huella de código y / o ancho de banda de red es un bien escaso. “Este protocolo

⁴ OASIS. *MQTT Version 3.1.1*. <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>. Consulta: 30 de mayo de 2018.

fue aprobado como un estándar OASIS en octubre del 2015 y como un estándar ISO en junio del 2016.”⁵

La arquitectura MQTT sigue una topología estrella, la cual contiene un nodo central (*broker*) con capacidad de trabajar un gran número de clientes, el *broker* es el encargado de gestionar la red y transmitir los mensajes, este publica la información a los clientes que previamente se han suscrito.

Figura 5. **Arquitectura MQTT**



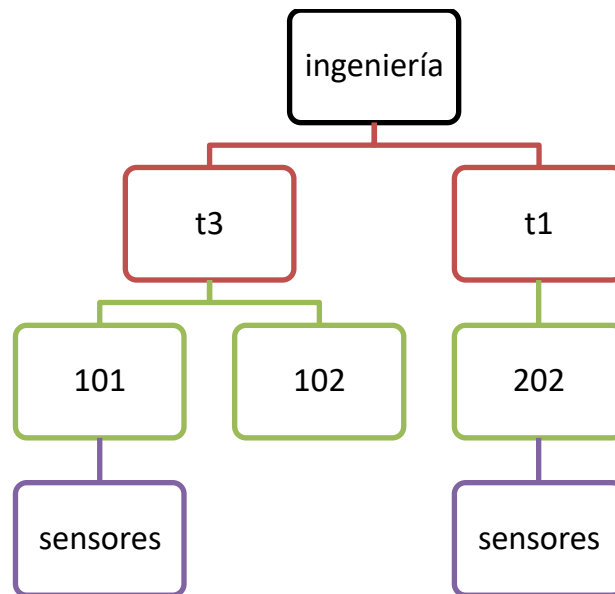
Fuente: elaboración propia, empleando Visio.

Otro elemento importante dentro de esta arquitectura es el *topic* (tema), con este elemento se centra la comunicación dado que los emisores y receptores deben estar suscritos a un *topic*. Esta comunicación puede ser de uno a uno o de uno a muchos. Todos los *topics* tienen una estructura jerárquica, la cual permite tener relaciones padre e hijo, dando la libertad que dentro de la comunicación se pueda suscribirse a un *topic* padre o a un *topic* hijo. Cada

⁵ MQTT community. *Frequently asked questions*. <http://mqtt.org/faq>. Consulta: 26 de junio de 2018.

jerarquía se separa por medio de una diagonal “/”, por ejemplo: ingeniería/t3/salón101/sensores e ingeniería/t1/salón202/sensores. Visto de una forma gráfica sería:

Figura 6. **MQTT topics**



Fuente: elaboración propia, empleando Visio.

Una sesión MQTT se divide en cuatro etapas: conexión, autenticación, comunicación y terminación.

- **Conexión:** el cliente comienza creando una conexión TCP/IP por medio de un puerto estándar.
- **Autenticación:** el cliente debe enviar los datos de usuario y contraseña, que en el caso de que la conexión no sea encriptada, la autenticación que presente el cliente al servidor está en texto plano.

- Comunicación: un cliente puede realizar operaciones como publicación, suscripción, anulación y *ping*.
- Terminación: cuando un suscriptor quiere finalizar la conexión MQTT envía un mensaje de desconexión y luego cierra la sesión.

MQTT cuenta con tres niveles de QoS (calidad de servicio, por sus siglas en inglés), dependiendo el nivel de QoS, este puede ser más confiable, pero tiende a afectar la latencia y el ancho de banda:

- QoS 0: servicio no reconocido, es una secuencia de paquete PUBLISH, el Publisher envía un mensaje al *broker* y este lo envía a sus suscriptores solamente una vez y no verifica si el mensaje ha sido recibido.
- QoS 1: servicio reconocido, este nivel utiliza la secuencia de paquetes PUBLISH/PUBACK, verifica que por lo menos una vez sea recibido el mensaje, pero puede ocasionar que el suscriptor reciba varias copias del mismo mensaje.
- QoS 2: servicio asegurado, en este nivel se utilizan dos pares de paquetes: PUBLISH/PUBREC y PUBREL/PUBCOMP. Verifica que, no importando el número de intentos, el mensaje se reciba solamente una vez.

1.5. Buses y protocolos de comunicación serial

“Un bus externo es un conjunto de líneas de señales que interconectan microprocesadores, computadoras, a los que a su vez se conectan a un periférico.”⁶ La transmisión de datos de forma serial permite que datos y señales de control se envíen bit a bit de forma secuencial a través de una sola línea. Esta transmisión de datos puede ser síncrona o asíncrona. En la transmisión

⁶ BOLTON, William. *Mecatrónica. Sistemas de control electrónico en la ingeniería mecánica y eléctrica*. p. 366.

asíncrona el receptor y emisor no están sincronizados, por lo cual requieren de bits adicionales para inicio y fin de transmisión, siendo esta menos eficiente. En la síncrona el receptor y emisor tienen una señal de temporización en común que inicia y finaliza los caracteres en cada ciclo.

Para la comunicación existen 3 tipos básicos de transmisión:

- Simplex: la transmisión solo es posible en una dirección.
- Semidúplex: la transmisión solo permite una dirección a la vez, pero puede ir hacia ambos lados.
- Dúplex completo: la transmisión puede ser hacia ambos lados de forma simultánea, por lo general se utilizan dos líneas de transmisión.

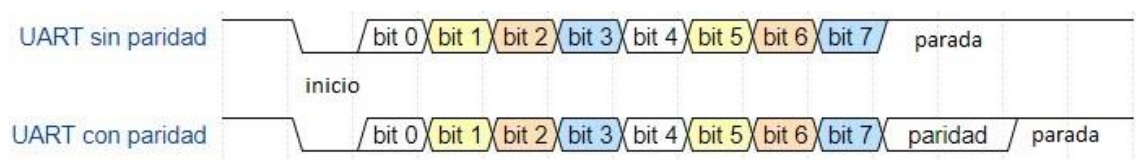
1.5.1. UART

El transmisor-receptor asíncrono universal (UART, Universal Asynchronous Receiver-Transmitter) es un dispositivo de comunicación serial asíncrona, en el cual el formato de los datos y la velocidad de transmisión puede ser configurable. Este dispositivo puede encontrarse solo pero también puede formar parte de un microcontrolador. Contiene registros de corrimiento, los cuales convierten los datos provenientes de una transmisión paralela a serie y viceversa, el dispositivo UART también tiene una variante llamada USART (Universal Synchronous/Asynchronous Receiver/Transmitter).

La comunicación asíncrona no tiene una señal de reloj propia para el envío de los datos, esta señal depende de los ciclos de reloj del microcontrolador, por lo tanto, ambos dispositivos que se comunican deben tener la misma tasa de baudios.

Para la comunicación síncrona el transmisor genera el ciclo de reloj para que el receptor utilice. En los casos que el ciclo de reloj se envía por medio de otra línea, la tasa de baudios puede llegar a ser muy alta, arriba de 4 000 000 bits por segundo. La comunicación se realiza por medio de dos líneas independientes: Tx (transmisor) y Rx (receptor), las cuales pueden tener las configuraciones simplex, semidúplex y dúplex completo. La trama de comunicación contiene un bit de inicio, los datos, bit de paridad en el cual se puede obviar y por lo general uno o dos bits de parada, la tasa de baudios estandarizada es de 9600, otros valores estándar pueden ser: 1200, 2400, 4800, 19200, 38400, 57600 y 115200.

Figura 7. Trama de datos UART

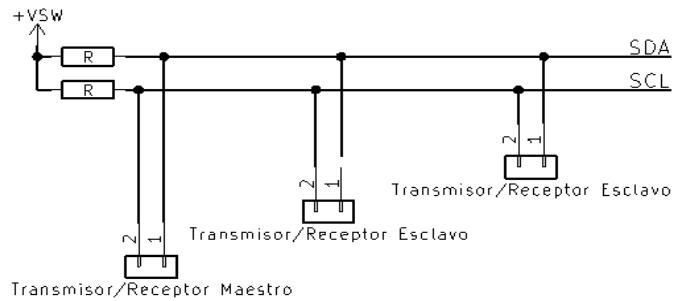


Fuente: elaboración propia, empleando WaveDrom.

1.5.2. I²C

El bus de intercomunicación con circuitos integrados (I²C, Inter Integrated Circuit) fue diseñado por Philips para la comunicación entre circuitos integrados por medio de dos cables.

Figura 8. Bus I²C

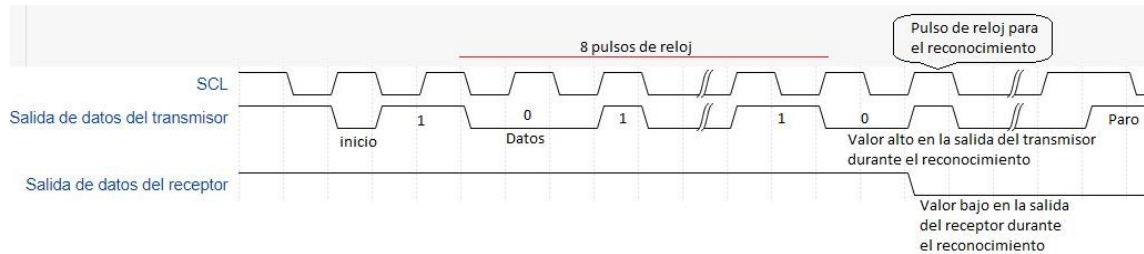


Fuente: elaboración propia, empleando KiCad.

SDA en la línea de transmisión son bidireccionales y SCL es la línea de temporización, ambas se conectan a una fuente de alimentación positiva a través de una resistencia, el dispositivo que controla la comunicación es el maestro, dado a que este verifica que para poder transmitir un mensaje el bus no esté ocupado. La comunicación se realiza bajo las siguientes condiciones:

- SDA y SCL están en alto, el bus no está ocupado.
- SDA cambia a bajo cuando SCL es alto, inicia la comunicación.
- SDA cambia de bajo a alto cuando SCL es alto, finaliza la comunicación.
- Durante la transferencia de datos, SDA es estable con SCL en alto, SDA cambia de estado cuando SCL está en bajo.
- Solo existe un pulso de SCL por cada bit de datos transmitido.
- El receptor envía un reconocimiento en el noveno bit, el cual es un nivel alto colocado en el bus por el transmisor y un valor de nivel bajo por el receptor.

Figura 9. Condiciones del bus I²C



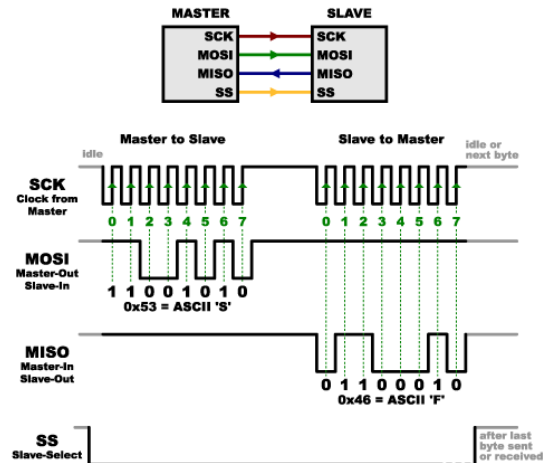
Fuente: elaboración propia, empleando WaveDrom.

1.5.3. SPI

El bus de interfaz de periféricos en serie (SPI, Serial Peripheral Interface) es usado para transmitir información entre microcontroladores y pequeños periféricos. Este utiliza líneas separadas para el reloj, los datos con comunicación a dúplex completo y una línea para seleccionar el dispositivo con el que se necesita establecer información. Esta interface requiere de cuatro señales lógicas:

- SCLK (Serial Clock): reloj serial emitido por el *master*.
- MOSI (Master Output Slave Input): emisor de datos del *master*.
- MISO (Master Input Slave Output): emisor de datos del esclavo.
- SS (Slave Select): selector de elemento esclavo del *master*, este elemento siempre está en alto y cuando inicia la comunicación con el esclavo su estado cambia a bajo.

Figura 10. Comunicación de bus SPI



Fuente: SPARKFUN. SPI. <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>.

Consulta: 29 de mayo de 2018.

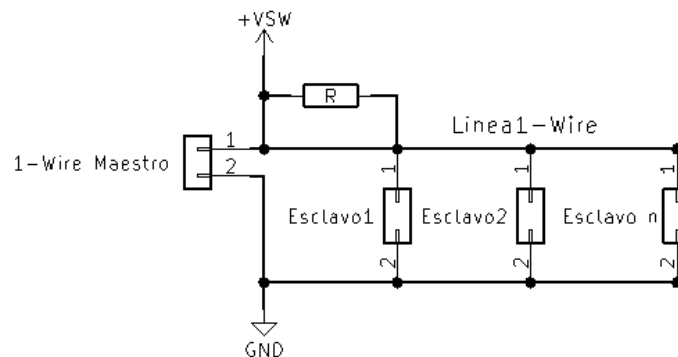
Estas cuatro señales se pueden utilizar en dos configuraciones básicas, una donde existe una línea de SS para cada dispositivo utilizado en el circuito y la configuración *Daisy-chained*, la cual utiliza una sola línea de SS conectada a todos los dispositivos y la línea de MISO se conecta en forma de cascada a la línea de MOSI del siguiente elemento.

1.5.4. 1 Wire

El dispositivo 1-*Wire* (un cable), es bus de comunicación creado por Dallas Semiconductor, el cual es similar al bus I²C, pero con tasas menores de datos y alto alcance. Usualmente se utiliza para establecer comunicación entre dispositivos de bajo costo tales como termómetros e instrumentos climáticos. La mayoría de buses 1-*Wire* incluye un capacitor de 800 pF para almacenar carga, con esto se alimenta el dispositivo al momento en el que la línea de datos se activa, dado que los dispositivos 1-*Wire* no tienen un pin para una fuente de

alimentación. Es el único sistema que trabaja bajo dos contactos, uno de datos y otro de referencia a tierra.

Figura 11. **Bus 1-Wire**



Fuente: elaboración propia, empleando KiCad.

Cada dispositivo 1-Wire tiene un número de identificación único e inalterable de 64 bits. La comunicación iniciada por el maestro tiene un tiempo de duración aproximado de 60 microsegundos. La comunicación se resume en tres fases específicas:

- Reseteo/Sincronización de dispositivos.
- Comandos de ROM, estos pueden ser leer, saltar, paridad, buscar y búsqueda de alarma.
- Comandos de memoria, estos son específicos para cada sensor.

1.6. **Microcontrolador**

Un microcontrolador es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria interna. Está compuesto de varios periféricos, los cuales cumplen tareas específicas. Este se compone de una

unidad central de procesamiento, interfaces de entrada y salida para poder manejar la comunicación hacia el mundo exterior, memoria, la cual almacena el código ejecutable y datos. Este puede contar con periféricos adicionales tales como temporizadores, interfaces de comunicación, entre otros.

1.6.1. ESP32S

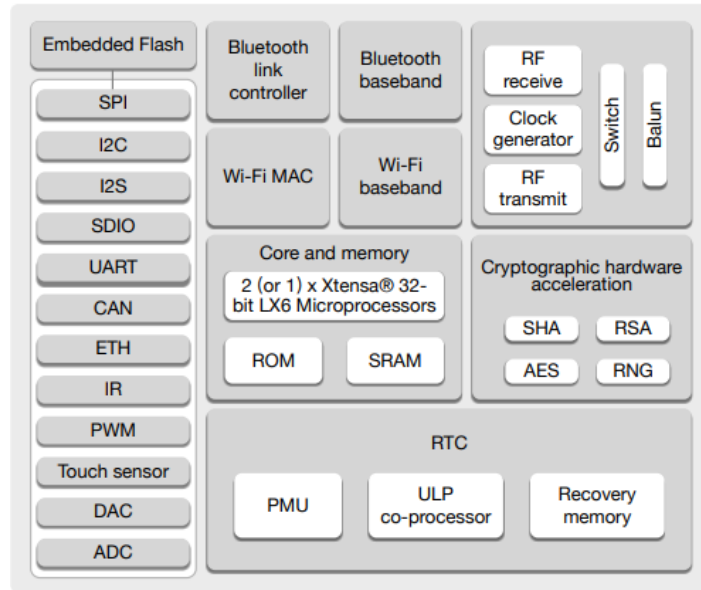
Este es un microcontrolador diseñado para IoT (Internet of Things) que tiene un modo de consumo de energía ultrabajo, es capaz de operar en ambientes industriales, soporta desde $-40\text{ }^{\circ}\text{C}$ hasta $125\text{ }^{\circ}\text{C}$, utiliza un procesador Dual-Core de 32 bits una máxima frecuencia de 240 MHz, al usar dos núcleos, permite tener uno enfocado a la comunicación y el otro a otros procesos, lo que permite que el microcontrolador cuente con elementos de comunicación tales como Wifi, Bluetooth, además de los protocolos I2C, UART, I2S, SPI e IR, permite encriptación del código en la *flash*. Debido a la flexibilidad del dispositivo y sus múltiples entornos de uso, es catalogado como un SoC (System on Chip) de bajo costo.

Figura 12. **ESP32**



Fuente: ESPRESSIF. *ESP32-DevKitC*. <https://www.espressif.com/en/products/hardware/esp32-devkitc/overview>. Consulta: 09 de junio de 2018.

Figura 13. Diagrama de bloques



Fuente: ESPRESSIF. *ESP32, Technical Reference Manual*. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Consulta: 09 de junio de 2018.

Este microcontrolador viene dentro de un módulo de desarrollo que contiene las siguientes características basadas en su hoja de datos técnicos:

Especificaciones:

- Wifi
 - Soporta los protocolos 802.11 b/g/n/d/e/i/k/r y 802.11n con una velocidad máxima de 150 Mbps
 - Utiliza un rango de frecuencia entre 2.4 GHz y 2.5 GHz
 - Soporta WMM (Wifi Multimedia)
 - Tx/Rx A-MPDU (*MAC protocol data unit*)
 - Rx A-MSDU (*MAC service data unit*)
 - 4 interfaces virtuales de Wifi
 - Soporta los protocolos de red
 - IPv4
 - IPv6
 - SSL
 - TCP
 - UDP
 - HTTP

- FTP
 - MQTT
 - Con opciones de seguridad
 - WPA
 - WPA2
 - WPA2-Enterprise
 - WPS
 - Bluetooth
 - Bluetooth v4.2 con BR (*Bluetooth Basic Rate*), EDR (*Enhanced Data Rate*) y BLE (*Bluetooth low energy*).
 - Transmisor de clase 1, 2 y 3 sin un amplificador externo.
 - Potencia de transmisión de +12 dBm
 - Receptor con sensibilidad de -97 dBm en BLE.
 - SCO/eSCO (*Synchronous Connection-Oriented/Extended*)
- Voltaje de operación entre 3.0 y 3.6 V
- Corriente de operación aproximada de 80 mA
- Temperatura en la cual puede operar entre -40 °C hasta 125 °C
- Microcontrolador
 - Unidad central de procesamiento y memoria
 - Xtensa® dual-core de 32-bit microprocesadores LX6
 - Es un microprocesador sin bloqueos en las etapas de segmentación
 - 448 kB de memoria ROM
 - 520 kB de memoria SRAM
 - 16 kB de memoria SRAM en RTC
 - Relojes y temporizadores
 - Oscilador interno de 8 MHz con calibración
 - Oscilador interno RC con calibración
 - Oscilador de cristal externo entre 2 MHz y 60 MHz
 - Dos grupos de temporizadores
 - ✓ 2 temporizadores de 64 bits
 - ✓ 1 temporizador watchdog en cada grupo
 - Temporizador RTC (*real time timer*)
 - RTC watchdog
 - Periféricos e interfaces
 - 34 puertos GPIO (entrada y salida de propósito general) programables
 - ADC de 12 bits
 - DAC de 8 bits
 - 10 detectores de contacto
 - 4 SPI
 - 2 I²S
 - 2 I²C
 - 3 UART
 - 1 host (SD/eMMC/SDIO)
 - 1 esclavo (SDIO/SPI)
 - Interface Ethernet con MAC dedicado, con soporte a DMA y IEEE 1588
 - CAN (*Controller Area Network*)
 - IR (Tx/Rx)
 - Control de motor por medio de PWM
 - 16 canales de control de LED por medio de PWM

- Sensor Hall
- Seguridad
 - Arranque seguro
 - Encriptación de memoria flash
 - 1024 bits para encriptación OPT (*one time pad*)
 - Aceleración de hardware criptográfico
 - ✓ AES
 - ✓ Hash (SHA-2)
 - ✓ RSA
 - ✓ ECC
 - ✓ Generador de números aleatorios RNG⁷.

Entre las aplicaciones comunes que puede tener este microcontrolador están: centro de sensores para IoT, cámaras de video, reproductores de música, juguetes Wifi, reconocimiento de voz, automatización en el hogar, controles industriales, entre otros.

1.7. Sistema de control

Un sistema de control es un conjunto de dispositivos encargados de administrar, ordenar, dirigir o regular el comportamiento de otro sistema, con el fin de reducir las probabilidades de fallo y obtener los resultados deseados, en pocas palabras manipular su respuesta dinámica mediante una función de transferencia, para cumplir las demandas de una tarea que se va a ejecutar. Por lo general, se usan sistemas de control para:

- Controlar las variaciones de algún valor en particular, por ejemplo, un sistema central de calentamiento donde la variable a controlar es la temperatura.
- Controlar la secuencia de eventos, por ejemplo, establecer secuencia de tiempos.

⁷ Espressif. ESP32, Technical Reference Manual. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Consulta: 09 de junio de 2018.

- Controlar si ocurre o no un evento, por ejemplo, tener un dispositivo de seguridad para que una máquina no pueda operar hasta que se le permita.

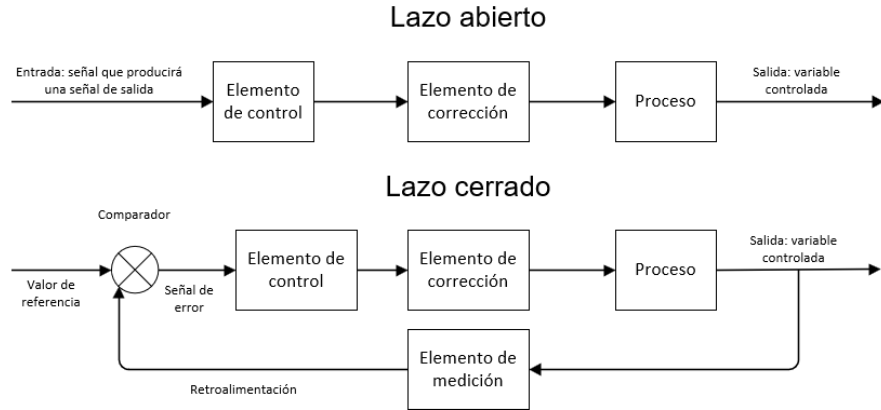
Existen dos tipos de sistemas de control: de lazo abierto y lazo cerrado, la diferencia entre ambos sistemas será el control por retroalimentación, donde las señales de salida son tomadas y tratadas para regresar como parte de la señal de entrada y se compara con el valor predeterminado (*set point*). Estos sistemas se generan con modelos matemáticos para luego convertirse en elementos físicos.

1.7.1. Sistema de control de lazo abierto y lazo cerrado

En el sistema de control de lazo abierto no se cuenta con retroalimentación, por lo cual la salida del sistema no tiene efecto en la señal de entrada. Estos sistemas tienen la ventaja de ser sistemas muy sencillos y de bajo costo con una buena confiabilidad, sin embargo, con frecuencia tienden a ser impreciso debido a que no existe corrección de errores ante cambios drásticos.

Los sistemas de control de lazo cerrado contienen retroalimentación con base en la señal de salida, por lo cual la señal de salida tiene efecto sobre la señal de entrada, modificándola para que la señal de salida se mantenga en el valor requerido. Esto hace que este tipo de sistemas sean precisos, pero incrementa la dificultad del sistema, costos y falla de componentes, dado que la cantidad de componentes que lo conforman son mucho más.

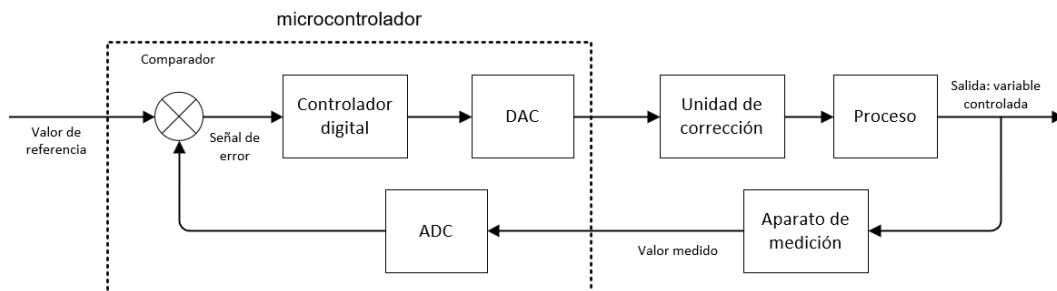
Figura 14. **Sistemas de control de lazo abierto y cerrado**



Fuente: elaboración propia, empleando Visio.

El sistema de control de lazo cerrado por lo general se divide en analógico y digital. El sistema analógico utiliza funciones continuas de tiempo, siendo el tamaño de la señal la medida de la variable. Para los sistemas digitales se considera una secuencia de señales de encendido y apagado, el valor de la variable se representa como una secuencia de pulsos de encendido y apagado. Dado que la mayoría de las señales en los sistemas de medición serán analógicas, este sistema necesita de dos componentes básicos, el ADC convertidor de analógico a digital y el DAC convertidor de digital a analógico.

Figura 15. **Sistemas de control digital**



Fuente: elaboración propia, empleando Visio.

1.7.2. Tipos de control

Dada la variación de configuraciones en las que se deben aplicar los sistemas de control, existe una diversidad de tipos de controlar un sistema.

- “Control todo o nada: el control adopta únicamente dos posiciones, para un valor único de la variable controlada, este funciona satisfactoriamente si el proceso tiene una velocidad lenta y un mínimo retardo.”⁸
- “Control flotante: mueve el elemento final de control a una velocidad constante dentro de un rango de operación.”⁹
- “Control proporcional P: existe una relación lineal continua entre el valor de la variable y la posición final del elemento de control, conocida como ganancia, con el inconveniente de la desviación permanente entre la variable una vez estabilizada con relación al *offset*.”¹⁰ La ecuación que determina el control proporcional está basada en el valor de error $e(t)$ y la ganancia proporcional K_p

$$P_{out} = K_p e(t) \quad (1.8)$$

- “Control integral I: el control integral actúa cuando existe una desviación entre la variable y el *offset*. La ecuación que determina el control integral.”¹¹

⁸ CREUS, Antonio. *Instrumentación Industrial*. p. 504.

⁹ *Ibíd.*, p. 506.

¹⁰ *Ibíd.*, p. 508.

¹¹ *Ibíd.*, p. 511.

$$I_{out} = K_i \int_0^t e(\tau) d\tau \quad (1.9)$$

- “Control derivado D: existe una relación lineal continua entre la velocidad controlada y la posición del elemento final, el control derivativo actúa cuando existen cambios en las variables. La ecuación que determina el control derivativo.”¹²

$$D_{out} = K_d \frac{d e(t)}{dt} \quad (1.10)$$

- “Control proporcional + integral + derivado PID: es la unión en un control de las tres acciones la proporcional, la integral y la derivada, este calcula la desviación o error entre un valor medido y un valor deseado.”¹³ La ecuación (1.11) que determina el comportamiento del controlador PID para la variable manipulada (VM) es la siguiente, y la ecuación (1.12) su equivalente en el dominio de Laplace donde s es la frecuencia compleja:

$$VM(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d e(t)}{dt} \quad (1.11)$$

$$L(s) = K_p + \frac{K_i}{s} + K_d s \quad (1.12)$$

¹² CREUS, Antonio. *Instrumentación Industrial*. p. 512.

¹³ *Ibíd.*, p. 514.

2. SELECCIÓN DE SENSORES

Debido a las variables básicas del sistema acuapónico, se necesita realizar mediciones del entorno tales como humedad, pH, nivel de agua, temperatura y luminosidad. Estas mediciones serán realizadas por medio de sensores electrónicos, los cuales fueron seleccionados bajo los siguientes términos:

- Costo
- El fácil acceso local
- Consumo energético
- Exactitud
- Precisión
- Poca calibración constante
- Fiabilidad
- Sensibilidad
- Histéresis

Debido a la diversidad de sensores en el mercado los factores que más influyeron la selección fueron el costo y el fácil acceso local, dado que la mayoría de sensores de bajo costo cumple con los parámetros básicos necesarios en el sistema acuapónico.

2.1. Humedad del ambiente y temperatura ambiente

Cuando se habla de la humedad del ambiente por lo general se está hablando de la humedad relativa, la cual es el vapor de agua en el ambiente y este a su vez depende de la temperatura. "La humedad relativa se debe controlar

en el sistema acuapónico dada la transpiración que influye en el déficit de vapor DVP, si el aire es muy seco la tasa de transpiración de la planta aumenta al punto de marchitarse o morir, por el contrario, si la humedad es alta la planta no absorbe agua por medio de sus raíces, esto genera falta de nutrientes en la planta.”¹⁴ Por lo tanto, la humedad relativa y la temperatura influyen en el crecimiento de las plantas, afectándolas a corto y a largo plazo.

Para la medición de la humedad del ambiente y la temperatura se optó por el sensor DHT11, siendo este un sensor de bajo costo y de fácil adquisición en el mercado guatemalteco.

Figura 16. **DHT11**



Fuente: AOSONG. *Temperature and humidity module DHT11*. p. 1.

Este es un sensor digital que provee una señal de salida digital, tiene una alta fiabilidad y alta estabilidad en tiempo de uso prolongado, es un sensor resistivo que está conectado a un microcontrolador de 8 bits, utiliza el protocolo de comunicación 1-wire bidireccional, tiene una respuesta rápida al momento de

¹⁴ PEERY, JoAnn. *¿Cómo influye la humedad en la calidad de los cultivos?* <https://www.pthorticulture.com/es/centro-de-formacion/como-influye-la-humedad-en-la-calidad-de-los-cultivos/>. Consulta: 1 de junio de 2018.

realizar mediciones y viene previamente calibrado, los parámetros principales del sensor con base en la hoja de especificaciones:

Tabla II. **Especificaciones técnicas**

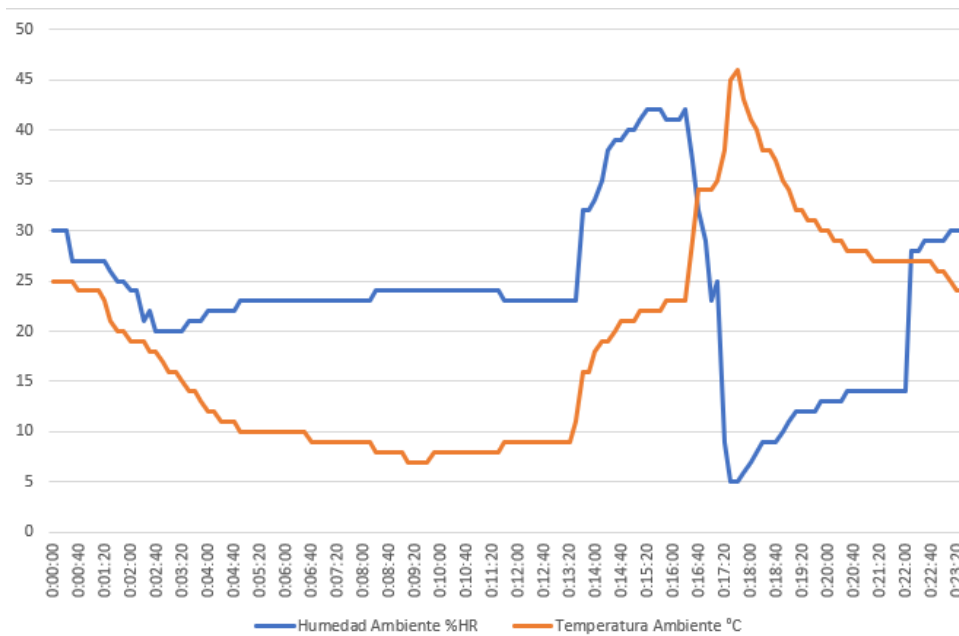
Parámetros	Condiciones	Mínimo	Nominal	Máximo
Humedad				
Resolución		1 % HR	1 % HR	1 % HR
Repetitividad			± 1 % HR	
Exactitud	0 – 50 °C			± 5 % HR
Rango de Medición	0 °C	30 % HR		90 % HR
	25 °C	20 % HR		90 % HR
	50 °C	20 % HR		80 % HR
Tiempo de respuesta	1/e(63 %) 25 °C 1 m/s V _{aire}	6 s	10 s	15 s
Histéresis			± 1 % HR	
Estabilidad a largo plazo			± 1 % HR/año	
Temperatura				
Resolución		1 °C	1 °C	1 °C
Repetitividad			± 1 °C	
Exactitud		± 1 °C		± 2 °C
Rango de medición		0 °C		50 °C
Tiempo de respuesta	1/e(63 %)	6 s		30 s
Características eléctricas				
Voltaje de alimentación	DC	3,5 V	5 V	5,5 V
Corriente de alimentación	Al medir	0,5 mA		2,5 mA
	En promedio	0,2 mA		1 mA
	En descanso	100 µA		150 µA
Tiempo de muestreo		1 s		

Fuente: Mouser Electronics. *DHT11 humidity and temperature sensor*. p. 4.

En la prueba de funcionamiento se crean condiciones extremas para analizar el comportamiento del sensor con base en cambios de temperatura y su tiempo de respuesta ante estos cambios, tomando mediciones cada 10

segundos, además se analizó el comportamiento del sensor durante un período de 24 horas, realizando mediciones cada 30 minutos.

Figura 17. **Gráfica de humedad y temperatura variante**



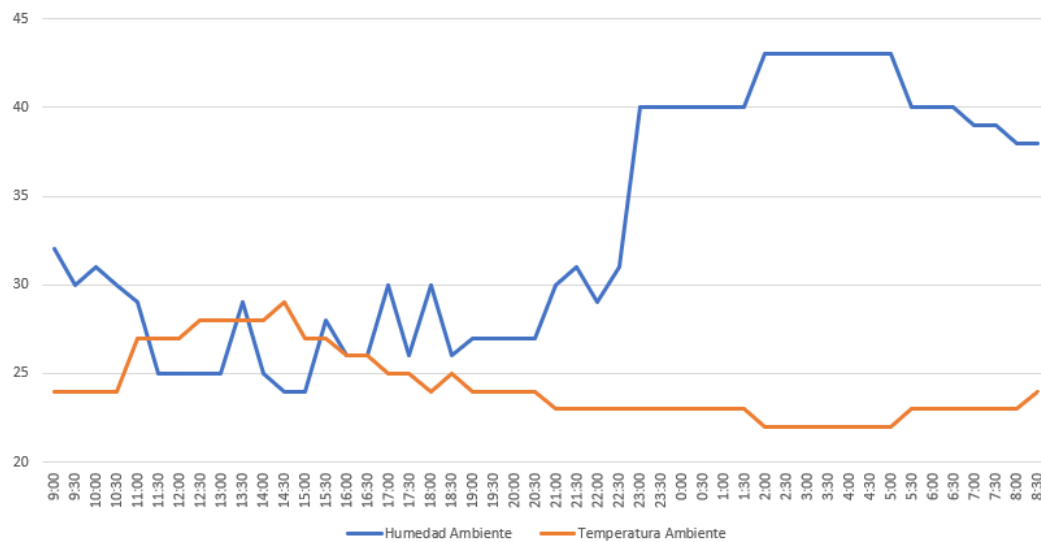
Fuente: elaboración propia.

Con base en el comportamiento del sensor al presentar cambios de temperatura, el cambio de temperatura ambiente 25 °C, a la temperatura más baja registrada 7 °C, decrece de forma gradual tal y como se tenía previsto, el cambio de temperatura de frío a calor es mucho más drástico y rápido, siendo la máxima medición registrada 48 °C. Al comparar estos datos con una termocupla, el sensor utilizado tiene mejor resolución y mejor tiempo de respuesta.

Los datos obtenidos de humedad relativa cumplen con las condiciones teóricas sobre el comportamiento de la humedad con respecto a la temperatura,

tal como lo recomienda Peery en su artículo sobre la influencia de la humedad en la calidad de los cultivos.

Figura 18. **Gráfica de humedad y temperatura, realizando mediciones cada 30 minutos en un período de 24 horas**



Fuente: elaboración propia.

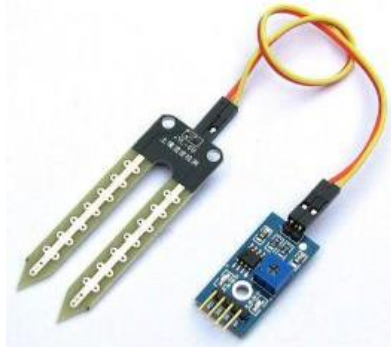
2.2. Humedad del suelo

La humedad del suelo es vital para las plantas dado que es su mecanismo para la absorción de nutrientes, saber el porcentaje de humedad del suelo permite evitar que las plantas se sequen debido a la descompensación en la turgencia, esto por la falta de humedad en el suelo, el exceso de humedad, favorece a la proliferación de hongos y enfermedades en las plantas.

Para la medición de la humedad del suelo se utiliza el sensor que está formado por los módulos YL-69 y YL-38, este sensor mide la humedad presente

en el suelo a su alrededor, empleando dos electrodos que emiten una corriente a través del suelo, midiendo la resistencia para poder determinar la humedad.

Figura 19. **YL-69 y YL-38**



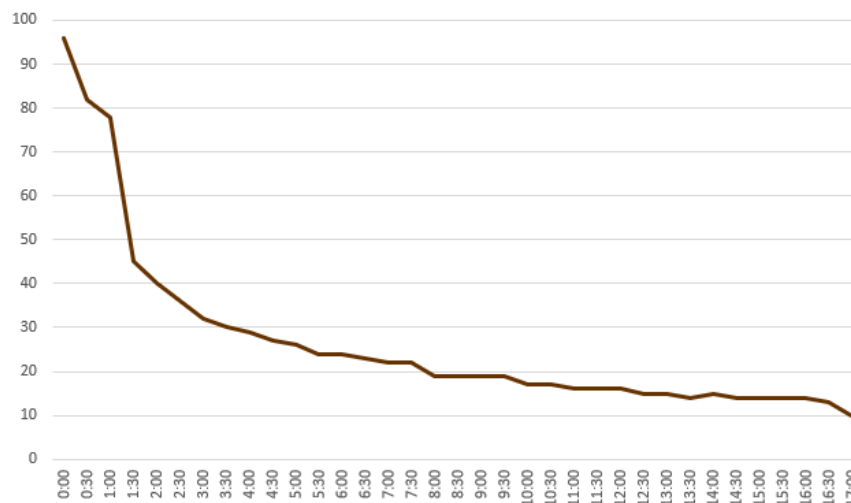
Fuente: CALDAS. YL-69. <http://www.electronicoscaldas.com/sensores-de-humedad-lluvia-inundacion/461-sensor-de-humedad-en-suelo-yl-69.html>. Consulta: 4 de junio de 2018.

Las características de ambos módulos son las siguientes:

- Medida analógica de la humedad respecto a variaciones de voltaje, el cual varía entre 0 ~ 4,2 V, el valor emitido por el sensor es proporcional a la resistencia que se puede interpretar en tres estados básicos.
 - Tierra mojada 0 – 350
 - Tierra húmeda 350 – 600
 - Tierra seca 600 – 1023
- Señal digital de superación de umbral, con led indicador, este umbral se configura por medio de un trimmer que se compara con un LM393.
- Voltaje de alimentación de 2 ~ 6 V.
- Consumo de corriente de 0 ~ 35 mA.

Con base en la hoja de especificaciones de los módulos y las indicaciones del fabricante, se realiza experimentación de valores de humedad y su comportamiento conforme el tiempo, esto permite realizar una escala de humedad útil para el sistema acuapónico y el tipo de plantas que se utiliza. Se midió el comportamiento de tierra y grava desde el punto en el que estaba completamente mojada hasta estar completamente seca, realizando mediciones cada 30 minutos.

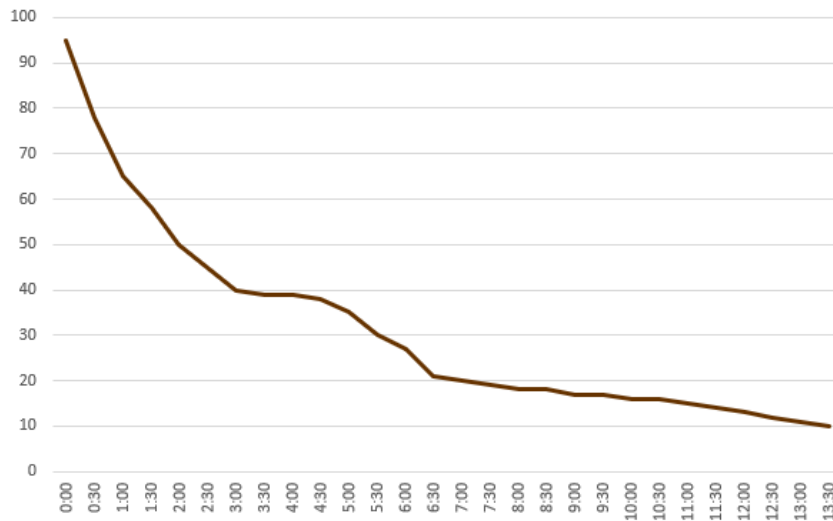
Figura 20. **Gráfica de humedad del suelo en tierra**



Fuente: elaboración propia.

Al realizarse mediciones en suelo con tierra abonada, tomó 17 horas, para una variación de humedad del 96 % al 10 % de esta, siendo un cambio drástico hasta el 30% de humedad y reduciendo de forma gradual después de esta marca.

Figura 21. **Gráfica de humedad del suelo en grava**



Fuente: elaboración propia.

Con base en las mediciones realizadas con grava bajo una temperatura aproximada de 26 °C, toma un período de 13,5 horas para variar la humedad de 95 % al 10 %, se tiene un tiempo de buena retención de agua de 5 horas, el cual es el tiempo en el que cambia de 95 % al 30 % de humedad. Con base en las recomendaciones sobre el cuidado de plantas, según el artículo técnico de la FAO,¹⁵ se determina el mínimo de humedad que tendrá cada planta a 10 %, se toma en cuenta que este tiempo tiende a variar, bajo condiciones de clima y absorción del agua por las raíces de las plantas.

Por lo cual es necesario el monitoreo constante de la humedad para determinar el momento en el que las plantas deben ser regadas.

¹⁵ SOMERVILLE, Christopher; et al. *Small-scale aquaponic food production – Integrated fish and plant farming*. p. 171.

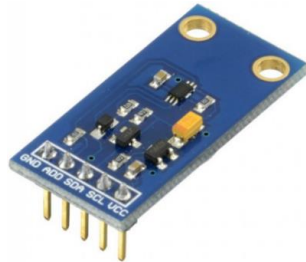
2.3. Luminosidad

Para que la fotosíntesis de las plantas se lleve a cabo se necesita ver tres factores importantes de la luz, cantidad, calidad y duración. Cuando la cantidad de luz que recibe una planta es alta, la velocidad del proceso fotosintético aumenta y llega a un punto máximo denominado punto de saturación de luz. En cuanto a la calidad de luz, se refiere a la longitud de onda de la luz recibida. El sol emite longitudes de onda entre los 280 y 2800 nm, sin embargo, las plantas fotosintetizan entre los 400 a 700 nm, por lo tanto “el proceso de fotosíntesis es muy eficiente bajo luz azul debido a su longitud de onda, otro punto crítico de absorción de luz para las hojas, es la luz roja debido a que el fitocromo dentro de las hojas es más sensible a la longitud de onda de la luz roja.”¹⁶ Dependiendo del tipo de planta, la cantidad de horas de luz varia, dado que las horas de luz día a día impactan directamente en el florecimiento de las plantas.

Para la medición de la luz se utiliza el sensor BH1750FVI, el cual se encuentra en el módulo GY-30, este es un sensor digital de bajo costo, entrega valores de medición en lux, tiene una alta precisión, su respuesta hacia el espectro visible es similar al del ojo humano, un alto rango de medición, sistema de comunicación sencillo por medio del protocolo i²C y cuenta con una calibración de fábrica.

¹⁶ CHEN, José. *Conceptos básicos de la fotosíntesis*. <https://www.pthorticulture.com/es/centro-de-formacion/conceptos-basicos-de-la-fotosintesis/>. Consulta: 4 de junio de 2018.

Figura 22. **GY-30**

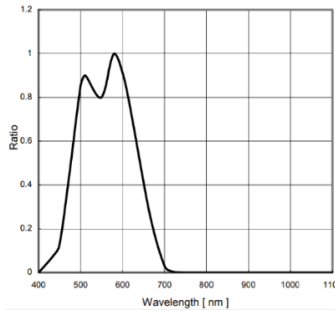


Fuente: GY-30. <https://core-electronics.com.au/bh1750fvi-digital-light-intensity-sensor-module-gy-30.html> Consulta: 11 de junio de 2018.

Las características del módulo, de acuerdo a su hoja de especificaciones técnicas, son:

- Voltaje de alimentación entre 2,4 ~ 3,6 V
- Voltaje de referencia para el puerto I²C entre 1,65 ~ Vcc V
- Consumo de corriente entre 120 ~ 190 μ A
- Longitud de onda máxima de 560 nm
- Precisión de 0,96 a 1,44 mediciones
- Incerteza en modo de alta resolución de 1 lux
- Incerteza en modo de baja resolución de 4 lux
- Rango de medición entre 1 – 65 535 lux

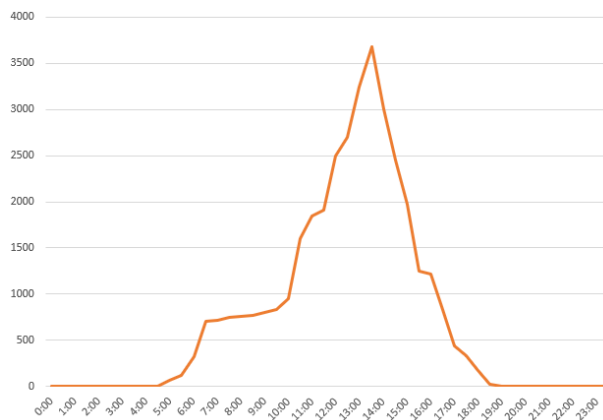
Figura 23. **Respuesta al espectro de luz**



Fuente: ROHM. *Digital 16-bit Serial Output type Ambient Light Sensor BH1750FVI*. p. 3.

Para determinar y simular las condiciones del sistema acuapónico se toman mediciones de un día completo de luz, con intervalos de 30 minutos, se inicia y finaliza la toma de datos a las 12:00 am, dichas mediciones se realizan en un ambiente sin luz artificial, la gráfica muestra el comportamiento del sensor bajo las condiciones de cada hora del día, llegando a la mayor medición a las 13:30 horas con un valor de 3678 luxes.

Figura 24. **Gráfica de luxes en un período de 24 horas**



Fuente: elaboración propia.

2.4. Nivel de agua

En cualquier tanque que se llena de forma automática se requiere la medición del agua cuando llega a un nivel óptimo, con esto se evita el desperdicio de agua al no parar el llenado a tiempo, y se mantiene un nivel de agua adecuado para la actividad que se esté realizando con este.

Para poder tener una medición de agua controlada se opta por utilizar el sensor ultrasónico HC-SR04, el cual permite medir distancias entre este y el objeto a medir, esta decisión se basa en que este sensor permite una medición fiable en diversas condiciones, es reemplazable en caso de daño, de fácil calibración y de bajo costo.

Figura 25. **HC-SR04**



Fuente: Cytron technologies. *User's manual HC-SR04*. p. 1.

Dentro de las características básicas del sensor están:

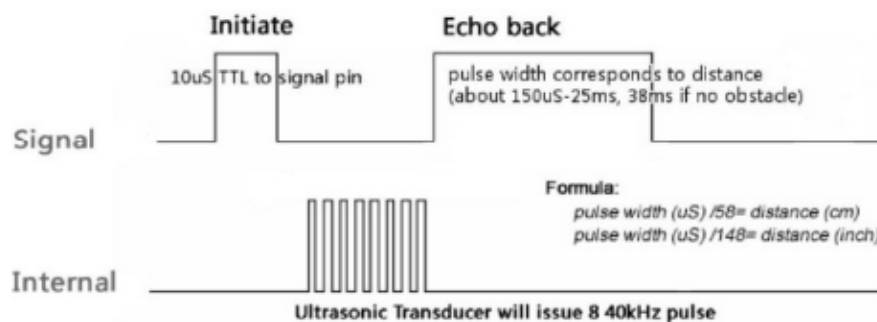
- Voltaje de alimentación: 4,5 ~ 5,5 V DC
- El consumo de corriente depende de dos estados:
 - Inactivo: 1,5 ~ 2,5 mA
 - Trabajando: 10 ~ 20 mA

- Angulo de efectividad: < 15 °
- Rango de distancia: 2 cm – 400 cm
- Resolución: 0,3 cm
- Ángulo de medida: 30 °
- Pulso del disparador de entrada: 10 μs
- Frecuencia ultrasónica: 40 kHz

El diagrama de tiempo de la figura 17 muestra el comportamiento del sensor, este luego de iniciar envía 8 pulsos ultrasónicos que se reflejarán en la superficie, el tiempo que tome en detectar el pulso determinará la distancia en centímetros, bajo la siguiente ecuación:

$$D = \frac{\text{Tiempo}}{58} \quad (2.5)$$

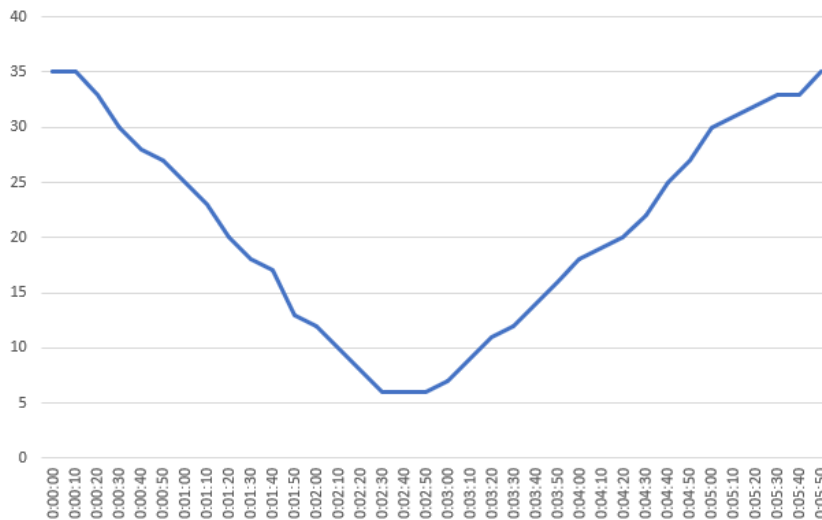
Figura 26. **Operación HC-SR04**



Fuente: Cytron technologies. *User's manual HC-SR04*. p. 7.

Se realizó en un período de 6 minutos el llenado y vaciado de un contenedor de 30 cm de alto, a una distancia de 35 cm del fondo del contenedor, para determinar la reacción del sensor a diferentes tiempos y en movimiento dinámico del agua, realizando mediciones cada 10 segundos.

Figura 27. **Gráfica del comportamiento de la altura con respecto al tiempo de llenado y vaciado del contenedor**



Fuente: elaboración propia.

2.5. Temperatura del agua

Para el sistema acuapónico el promedio de temperatura que debe tener el agua es de 18 a 30 °C¹⁷. Dado que esta determina la tasa metabólica de los peces, al aumentar la temperatura del agua la tasa de crecimiento aumenta y los procesos metabólicos y requerimientos energéticos también se incrementan. Por el contrario, con el descenso de las temperaturas decrece el potencial de crecimiento de los peces. Al aumentar la temperatura, disminuye la cantidad de oxígeno disuelto en el agua¹⁸. Para la medición de la temperatura del agua se utiliza el sensor ds18b20, este es un termómetro digital de alta precisión,

¹⁷ FAO. 2015.

¹⁸ Water Boards. *Folleto informativo. Oxígeno disuelto (OD)*. 2010. p. 4.

resistente al agua, de fácil uso, bajo costo y de fácil adquisición en el mercado guatemalteco.

Figura 28. **DS18B20**



Fuente: ADAFRUIT. *DS18B20*. <https://learn.adafruit.com/assets/3777>

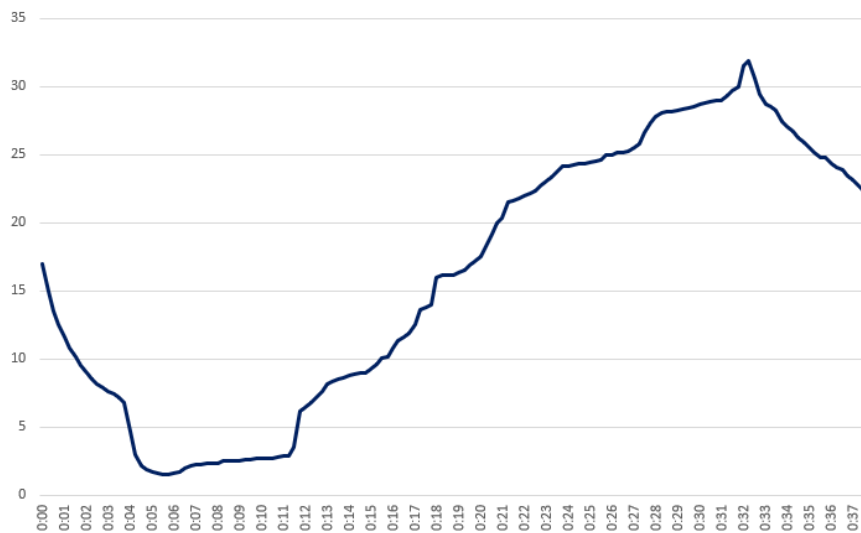
Consulta: 5 de junio de 2018.

Con base en la hoja de especificaciones las características principales de este sensor son:

- Protocolo de comunicación 1-Wire
- Voltaje de alimentación: 3,0 ~ 5,5 V
- Rango de medición: -55 ~ + 125 °C
- Resolución programable de: 9 a 12 bits
- Error del termómetro:
 - -10 ~ + 85 °C: ± 0,5 °C
 - -55 ~ + 125 °C: ± 2 °C
- Consumo de corriente:
 - En descanso: 750 ~ 1000 nA
 - Activo: 1 ~ 1,5 mA
- Tiempo de conversión: 93,75 ~ 750 ms.

Para ver el comportamiento del sensor se realizan mediciones en condiciones extremas por medio de un período de 38 minutos en intervalos de 20 segundos, analizando el comportamiento del sensor con base en los cambios de temperatura de agua con hielo a temperatura ambiente, agregando agua a 80 grados de forma gradual y agregando hielo después de pasar los 30 °C, de esa forma se aprecia el dinamismo del sensor de temperatura.

Figura 29. **Gráfica de temperatura del agua en condiciones variables**

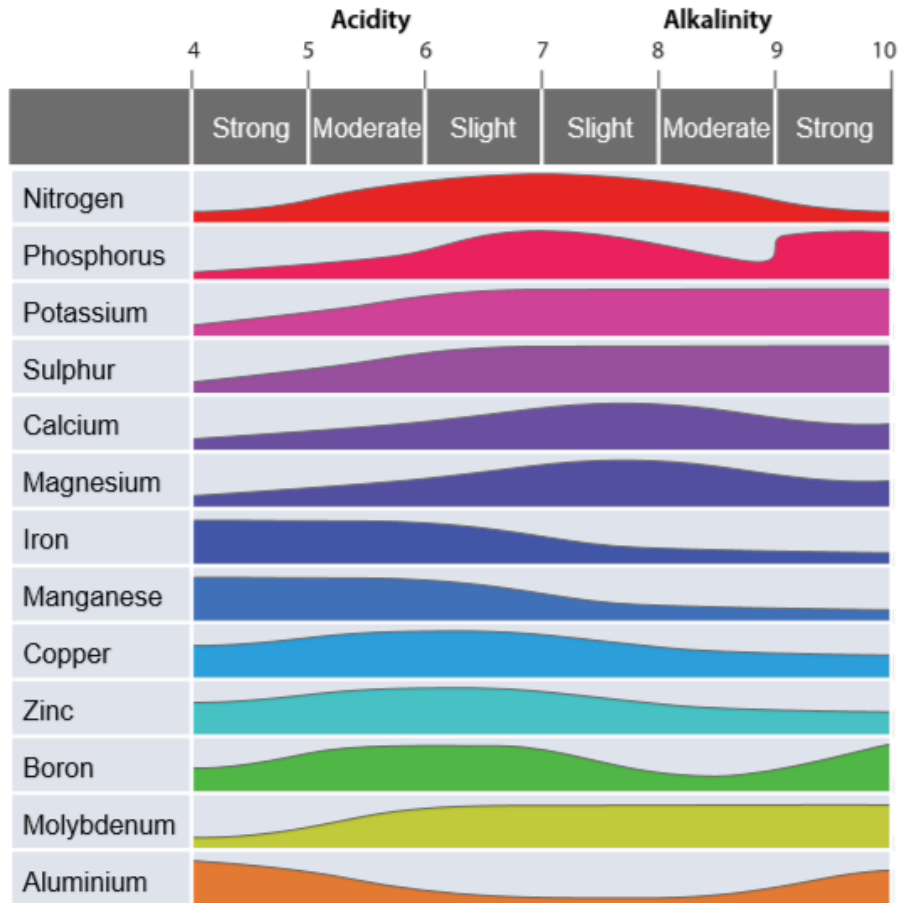


Fuente: elaboración propia.

2.6. pH

El monitoreo del pH del agua es fundamental para el sistema acuapónico, dado que mantiene a los peces en condiciones óptimas y las plantas asimilan mejor sus nutrientes en ciertas condiciones de pH. El pH para un sistema acuapónico debe estar entre 6 y 7, aunque depende del cultivo, ya que el pH tiene un impacto directo en los nutrientes, como muestra la figura 30.

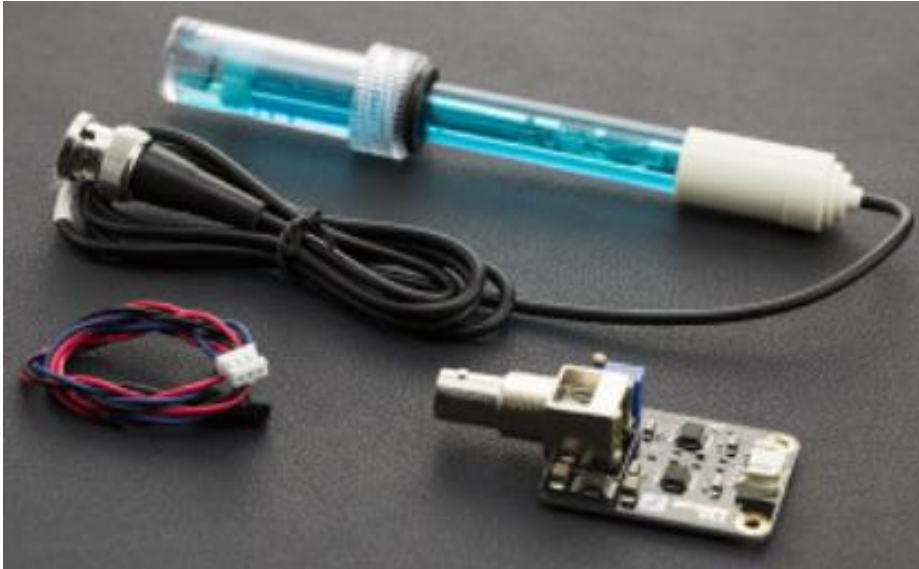
Figura 30. **Impacto del pH en los nutrientes**



Fuente: SOMERVILLE, Christopher; et al. *Small-scale aquaponic food production – Integrated fish and plant farming*. p. 91.

Para la medición del pH se utiliza el sensor pH meter (SKU: SEN0161), el cual es un sensor de bajo costo, la frecuencia de calibración es baja con respecto a otros sensores en el mercado, de fácil uso, contiene un módulo para el acondicionamiento de la señal y excelente exactitud.

Figura 31. **Sensor de pH SEN0161**



Fuente: DFROBOT. *Gravity: Analog pH Sensor Meter Kit for Arduino*.
<https://www.dfrobot.com/product-1025.html> Consulta: 5 de junio de 2018.

Con base en las especificaciones técnicas del electrodo de medición, este contiene los siguientes parámetros:

- Voltaje de alimentación 5 V
- Rango de medición de 0 a 14 pH
- Temperatura de funcionamiento 0 a 60 °C
- Exactitud de $\pm 0,1$ pH a 25 °C
- Tiempo de respuesta menor a 1 minuto
- Conector BNC
- Potenciómetro para ajustar la ganancia (para calibración)

La relación de salida de voltaje con base en el valor del pH y en mediciones a 25 °C se muestra en la siguiente tabla.

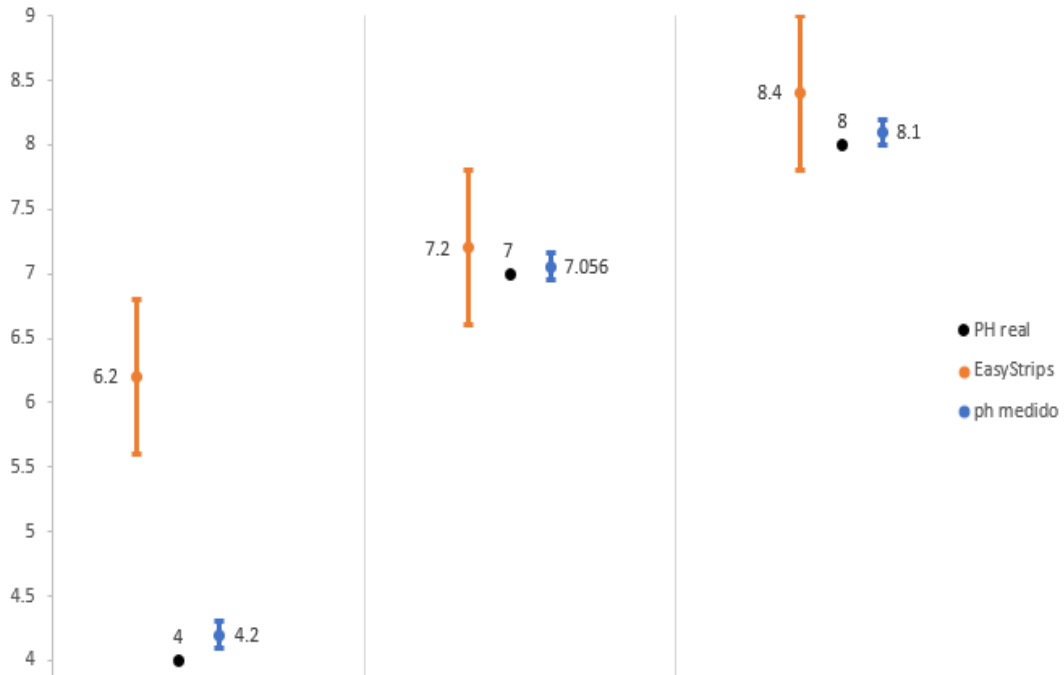
Tabla III. **Milivoltios vs pH**

Voltaje mV	pH
414,12	0
354,96	1
295,80	2
236,64	3
177,48	4
118,32	5
59,16	6
0	7
-59,16	8
-118,32	9
-177,48	10
-236,64	11
-295,80	12
-354,96	13
-414,12	14

Fuente: DFROBOT. *PH meter*(SKU: SEN0161). https://www.dfrobot.com/wiki/index.php/PH_meter_V1.1_SKU:SEN0161. Consulta: 12 de junio de 2018.

Con base en la hoja de especificaciones se miden 3 diferentes soluciones y se comparan con láminas de prueba química Tetra EasyStrips, la cual se utiliza en acuarios (véase la sección de anexos para la escala de estas láminas), tomando en cuenta la incerteza de cada elemento de medición, los resultados para el medidor de pH electrónico son precisos.

Figura 32. Gráfica de medidor de pH electrónico vs EasyStrips



Fuente: elaboración propia.

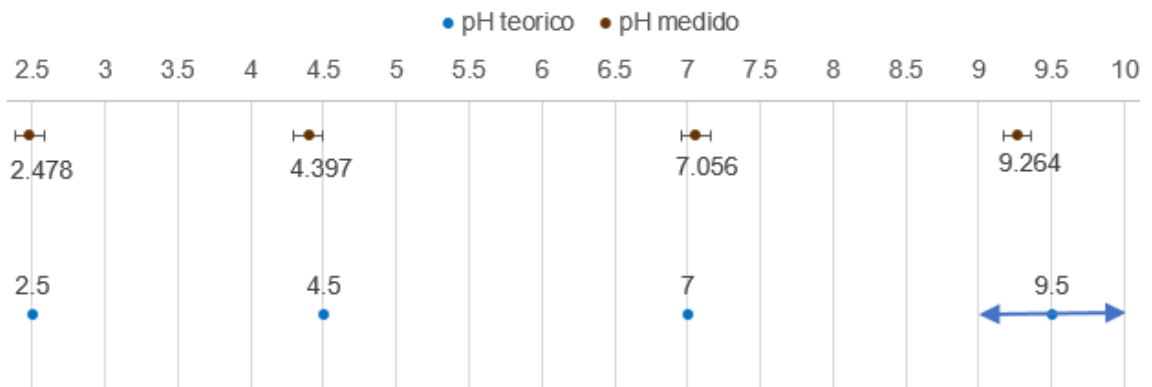
Con base en la tabla de la figura 33 se miden 4 diferentes elementos caseros para comparar el valor de pH teórico con el valor medido. Se toma muestras de: Coca Cola, cerveza, agua pura, jabón de manos. Con estas mediciones se afirma la precisión del sensor pH meter (SKU: SEN0161), para que las mediciones no se vean afectadas por contaminación cruzada, antes de cada medición se limpia el sensor con agua desmineralizada.

Figura 33. **Tabla de pH de elementos del hogar**

Refresco de cola		2.5
Vinaagre		2.9
Juqo naranja o manzana		3.0
Cerveza		4.5
Café		5.0
Té		5.5
Lluvia ácida		<5.6
Orina		5.5 - 6.5
Leche		6.5
Agua pura		7.0
Saliva humana		6.5 a 7.4
Sangre		7.35 a 7.45
Agua de mar		8.0
Jabón de manos		9.0 a 10.0

Fuente: UMC. *PH elementos de casa*. <https://webs.ucm.es/info/analitic/Asociencia/pH-Casa.pdf>
 Consulta: 5 de junio de 2018.

Figura 34. **Gráfica de pH de elementos caseros**



Fuente: elaboración propia.

2.7. Caudal

Debido a la naturaleza del sistema acuapónico, los residuos sólidos de comida y heces de los peces, tienden a tapar tuberías y las bombas de agua, las cuales después de cierto tiempo necesitan mantenimiento. Al medir el caudal que pasa por el tubo se determina si este disminuye, debido a que la velocidad de operación de la bomba de agua siempre será la misma, se puede determinar que existe algún tipo de obstrucción, ya sea en la tubería o en la bomba en sí, este se mide por medio del sensor FS3000A.

Figura 35. **Sensor FS3000A**



Fuente: AMAZON. *DealMux Hall effect wáter flow sensor FS300A*. <https://www.amazon.com/DealMux-Effect-Sensor-FS300A-Counter/dp/B072KQTWRF> Consulta: 5 de junio de 2018.

El sensor FS300A es un sensor construido con adaptadores para tuberías de $\frac{3}{4}$ " , este es un sensor de efecto *hall*, el cual, con base en sus especificaciones técnicas dadas por el proveedor, contiene los siguientes parámetros:

- Voltaje de alimentación: 4,5 V
- Rango de alimentación para su funcionamiento 5 V ~ 24 V
- Corriente máxima en funcionamiento: 15 mA
- Rango de medición de flujo: 0 ~ 60 L/min
- Temperatura de funcionamiento: 0 a 80 °C

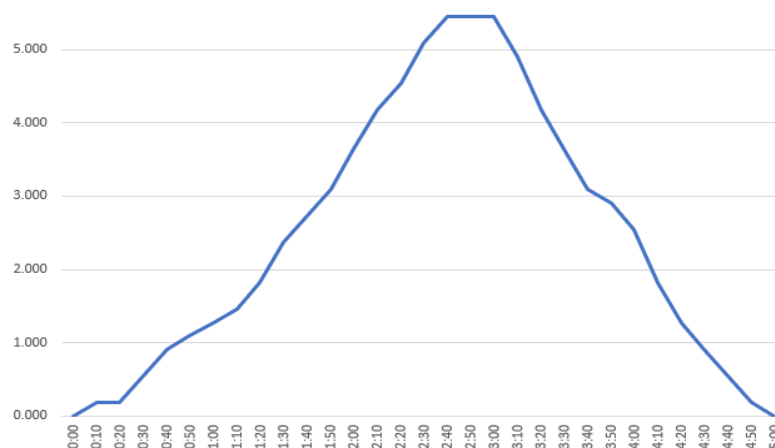
- Temperatura del líquido: 0 a 120 °C
- Presión del líquido que resiste: 0 a 2,0 MPa
- Tipo de rosca: ¾" NPS

El sensor proporciona una salida de pulsos de onda cuadrada cuya frecuencia es proporcional al caudal. El factor de conversión de frecuencia (Hz) a caudal (L/min) está determinado por el fabricante, da un factor de conversión de $K=5,5$ para el sensor FS300A, dando como resultado la siguiente ecuación:

$$f = 5,5 * Q \quad (2.6)$$

Se realiza medición del caudal de un grifo de hogar variando la cantidad de agua que sale del mismo para ver el comportamiento del sensor, las mediciones se realizaron por un período de 5 minutos en intervalos de 10 segundos. Esto muestra que el sensor es preciso en mediciones dinámicas.

Figura 36. **Comportamiento dinámico del sensor FS300A**



Fuente: elaboración propia.

3. DISEÑO DEL SISTEMA ACUAPÓNICO

Se utiliza un sistema acuapónico clásico con base en la guía *Técnicas de Acuaponía* (Candarle, 2014). Este se compone por tanque y dos camas de sembrado, se realiza a pequeña escala para la facilidad de presentación del prototipo, pero tiene el potencial de utilizarse y adaptarse a sistemas acuapónicos grandes con varios tanques y varias camas de sembrado.

Figura 37. **Propuesta del sistema acuapónico**

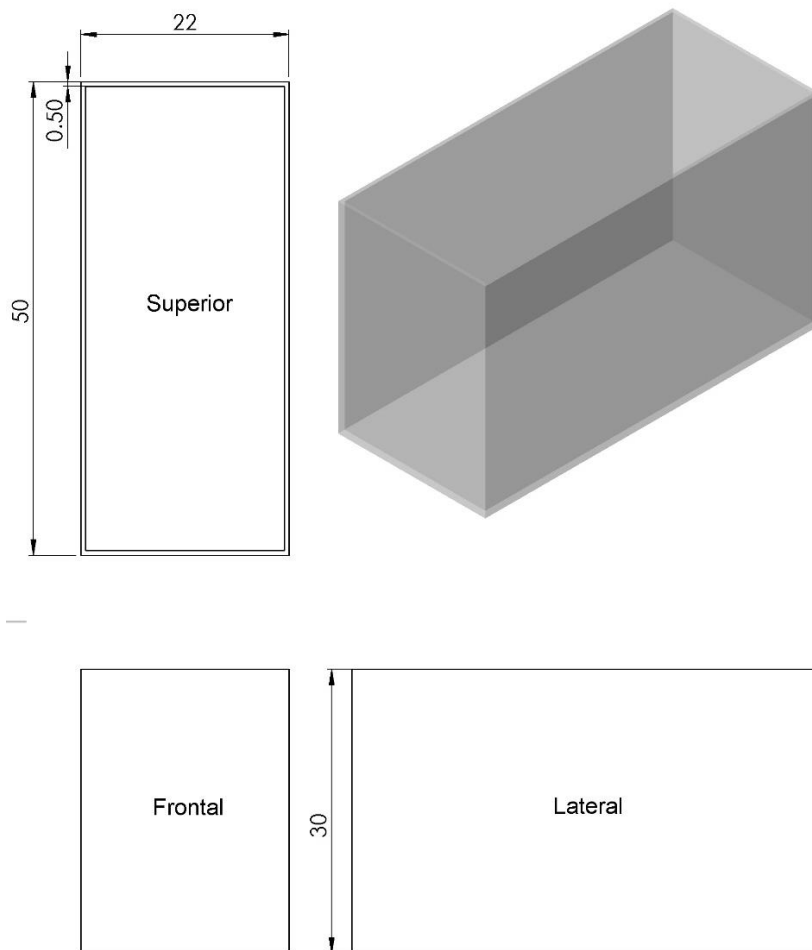


Fuente: elaboración propia, empleando Solidworks Student Edition.

3.1. Estanque

El estanque que se utiliza es una pecera de vidrio de dimensiones: 22 cm de ancho, 50 cm de largo y 30 cm de alto, con una capacidad de 0,03 m³, 33 litros de agua. Para mantener un parámetro de medición estable se utilizan 3 peces y una altura máxima de agua de 22 cm para mantener un rango de 22 a 28 litros de agua. (Candarle, 2014)

Figura 38. Estanque, mediciones en cm

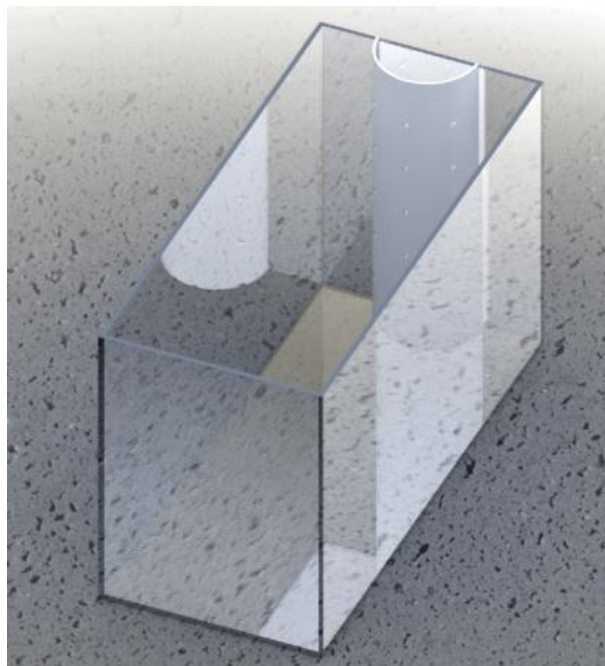


Fuente: elaboración propia, empleando Solidworks Student Edition.

Debido a que la altura del agua sube y baja, producto de la circulación de esta misma hacia las camas de sembrado, se llena el tanque a un 85 % del volumen con agua dulce, lo cual permite tener un índice de incerteza para el manejo del agua durante el proceso de circulación.

Se utiliza un tubo PVC de 4 pulgadas de diámetro partido por la mitad como medio de separación entre los seres vivos y las mediciones. De esta manera el nivel del agua dentro del estanque y la conductividad de la misma se puede medir sin ningún medio de distorsión o afectar a los peces que se encuentran en el estanque.

Figura 39. **Propuesta del montaje del tubo PVC en la cama de sembrado**



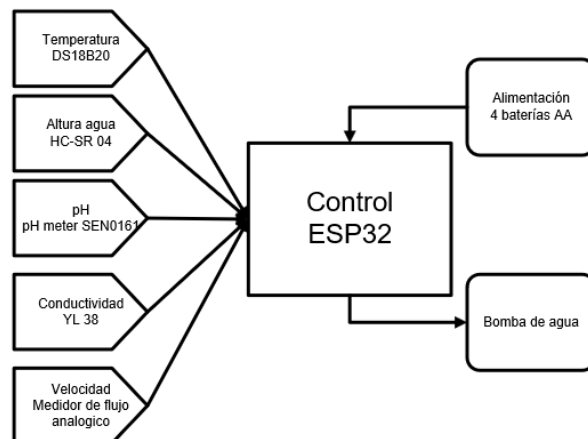
Fuente: elaboración propia, empleando Solidworks Student Edition.

3.1.1. Sensores y actuadores del estanque

Dada la necesidad de controlar los parámetros básicos para mantener el sistema acuapónico dentro del tanque y el flujo de agua que sale de este, se implementa un módulo de sensores y actuadores que se encarga de monitorear:

- Temperatura del agua con el sensor DS18B20.
- Altura del agua con el sensor HC-SR04.
- pH por medio del sensor PH meter SKU: SEN0161.
- Conductividad del agua por medio del módulo YL38 y dos cables, los cuales se miden por medio de un valor analógico.
- Velocidad de salida del agua por medio del sensor FS300A.
- Como actuador una bomba para circulación de agua.
- La alimentación está proporcionada por medio de 6 baterías AA de 1,2 VDC y 2400 mAh cada una, la bomba de agua se alimenta con 120 VAC.

Figura 40. Diagrama de bloques para el nodo de sensores del tanque



Fuente: elaboración propia, empleando Visio.

Como actuador se utiliza la bomba de agua Kintons KT:101, la cual está controlada por un relé de estado sólido y tiene las siguientes características:

- Alimentación: 110 V/ 60 Hz 5 W
- Columna de agua máxima de 1 m
- Caudal máximo 800 l/h

Figura 41. **Bomba de agua**



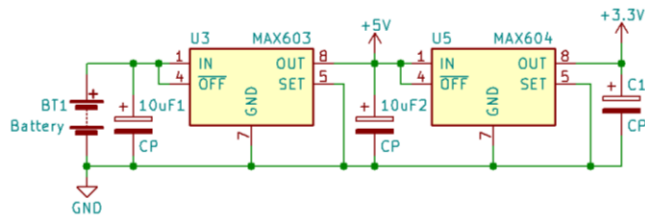
Fuente: elaboración propia, cámara Nikon coolpix B500.

El control de los sensores y el actuador es por medio del microcontrolador ESP32. Este se encarga de solicitar las mediciones de cada dispositivo, interpretarlas, procesarlas y encender la bomba de agua. Dado que este microcontrolador opera a 3,3VDC y algunos sensores operan a 5VDC, se implementa un circuito regulador de voltaje, para esto se utilizan los integrados MAX603 y MAX604, los cuales proveen voltajes de salida 5VDC y 3,3VDC respectivamente. Para disminuir el consumo de corriente, se utiliza un circuito tipo *switch* por medio de un transistor Mosfet 2N7000, el cual controla el momento

en que se enciende y se apaga un sensor. Se utilizó el programa KiCad para el diseño del esquema del circuito y su respectiva PCB.

Figura 42. Esquema del circuito para el nodo de sensores del tanque

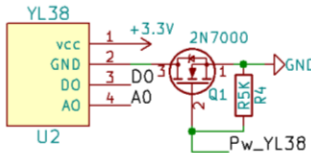
Circuito de regulación de voltaje:
Este proporciona los voltajes de alimentación para todo el nodo, depende de una entrada de voltaje DC mayor a 5V



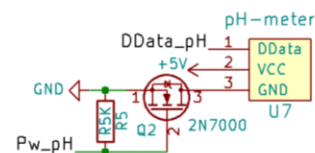
Sensor Temperatura



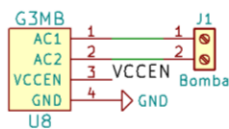
Sensor Conductividad



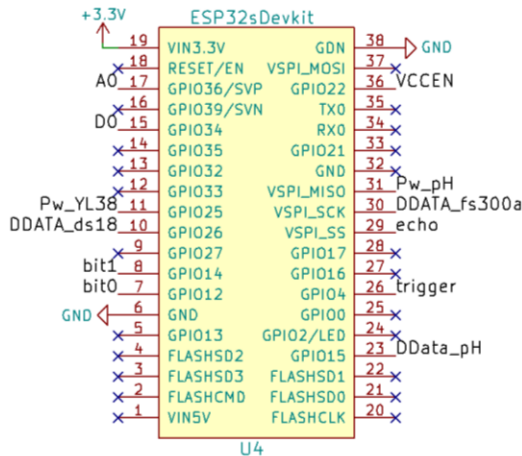
Sensor pH



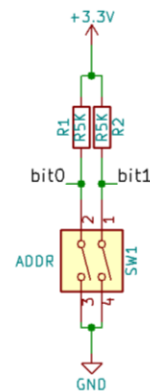
Actuador Bomba Agua



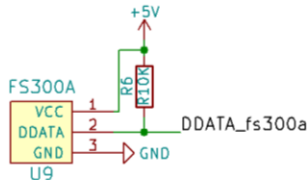
Elemento de Control:
ESP32S



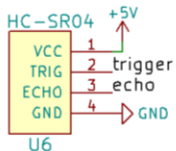
DIP SWITCH ADDRESS



Sensor de Caudal

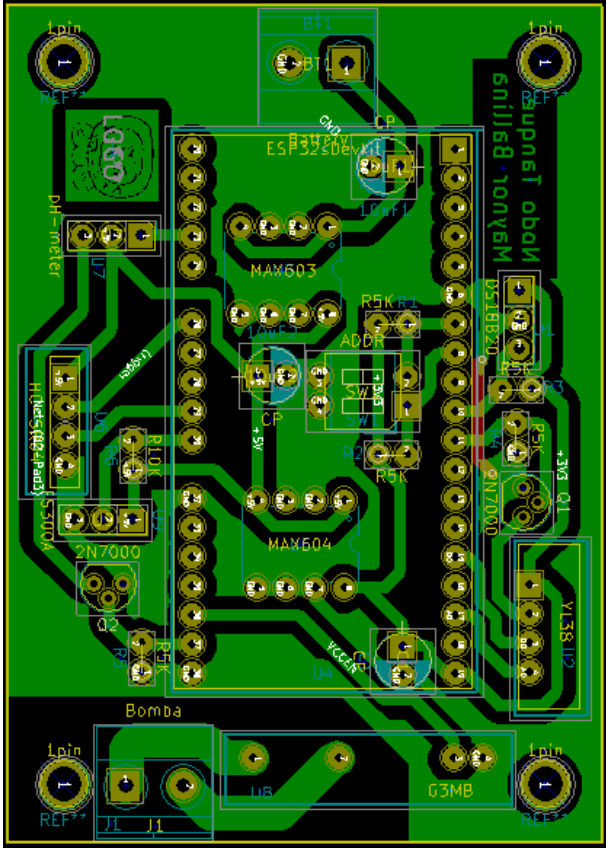


Sensor Altura Agua



Fuente: elaboración propia, empleando KiCad.

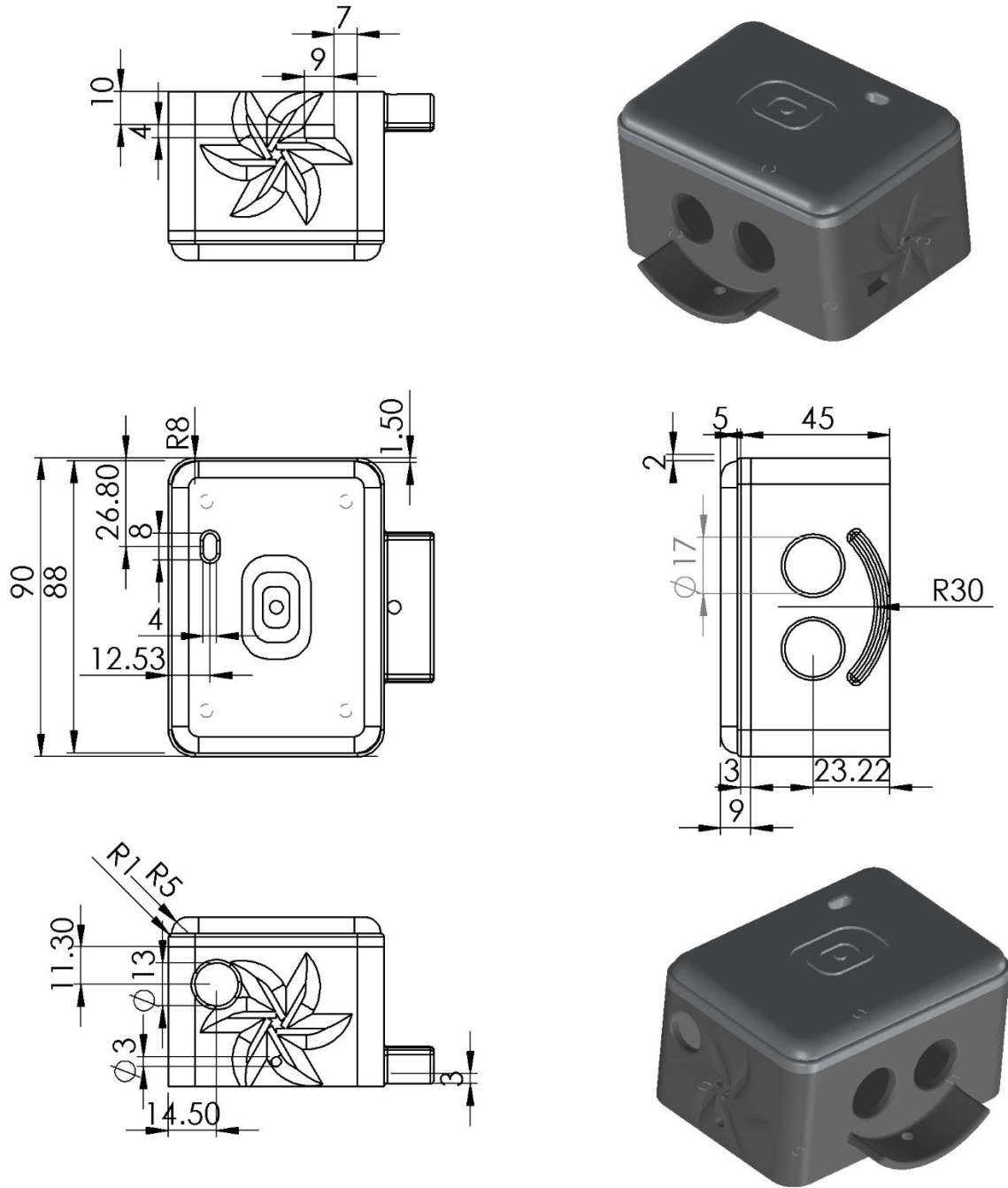
Figura 43. Esquema del PCB para el nodo de sensores del tanque



Fuente: elaboración propia, empleando KiCad.

Dado que este nodo debe estar cerca de agua para que el PCB no se dañe y facilite su instalación en el tanque, se diseña un contenedor, el cual se elaboró en ABS, las mediciones están especificadas en la figura 44.

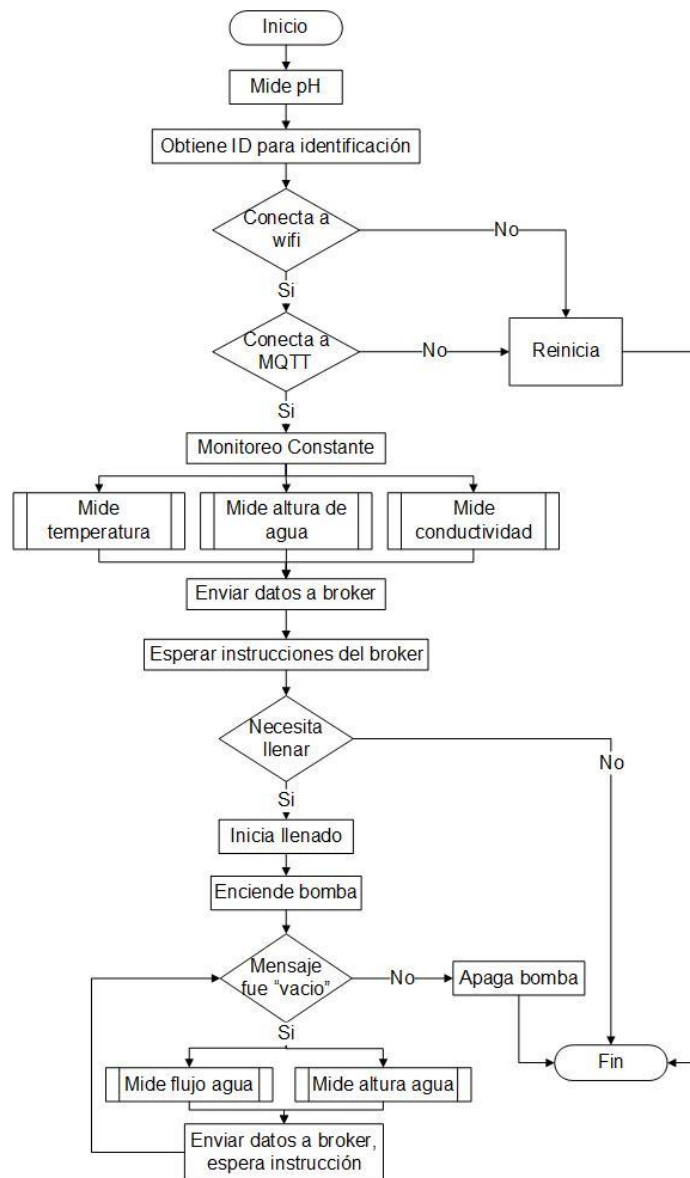
Figura 44. Caja de nodo de tanque, mediciones en mm



Fuente: elaboración propia, empleando Solidworks Student Edition.

Para realizar el control del tanque se realiza la programación por medio de la plataforma Arduino IDE, esta se puede encontrar en el área de apéndice. El diagrama de flujo define cómo se implementa el control del nodo.

Figura 45. Diagrama de flujo del nodo del tanque

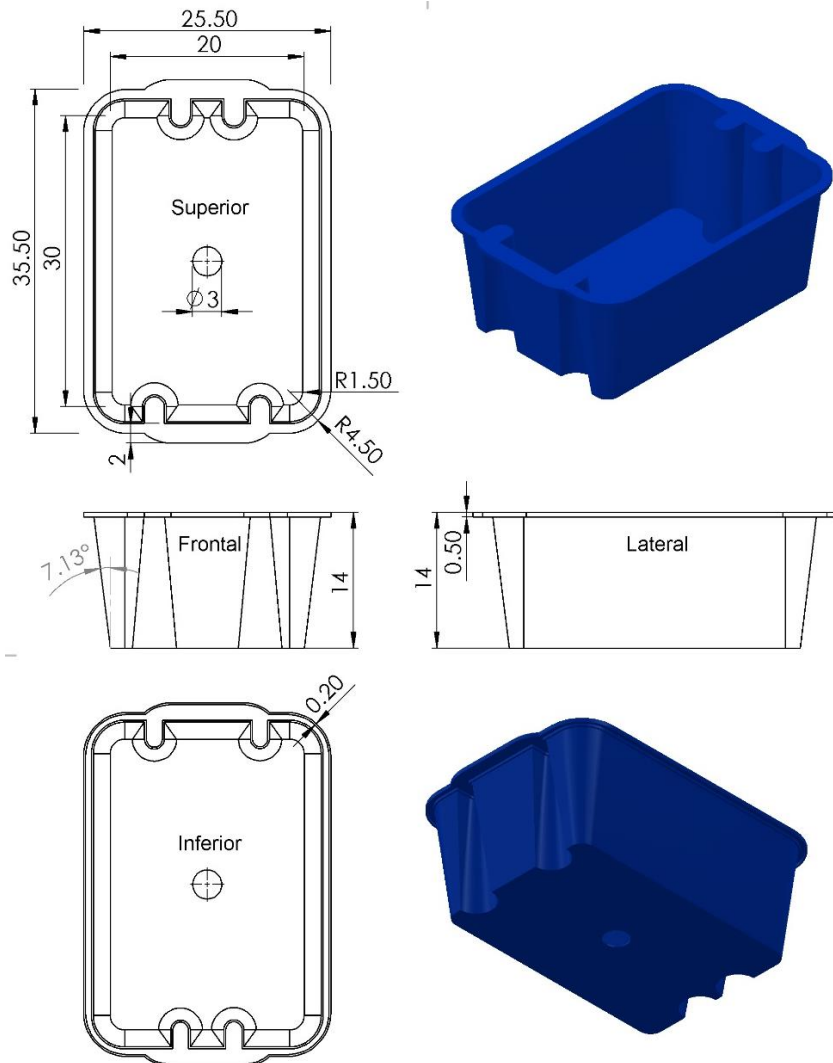


Fuente: elaboración propia, empleando Visio.

3.2. Cama de sembrado

Para la cama de sembrado se utiliza un contenedor de plástico con dimensiones: 20 cm de ancho, 30 cm de largo y 14 cm de alto, con un ángulo de salida de $7,13^\circ$, el cual tiene un volumen de 481,04 centímetros cúbicos.

Figura 46. Cama para sembrado, mediciones en cm



Fuente: elaboración propia, empleando Solidworks Student Edition.

Para realizar el proceso de sembrado se utiliza grava en lugar de tierra, con base en la guía de *Técnicas de acuaponía* (Candarle, 2014), dado que la función principal del sustrato es brindar una importante superficie de contacto para el alojamiento y colonización de las bacterias nitrificantes, además de brindar soporte a las plantas. Se llena la cama de sembrado a un 70 % del volumen con grava, lo cual permite tener un patrón de medición para el manejo del agua que llenará este elemento.

Se utiliza un tubo PVC de 2 pulgadas de diámetro como medio de separación, de esta manera el nivel del agua dentro de la cama se puede medir sin ningún medio de distorsión.

Figura 47. **Propuesta de montaje del tubo PVC en la cama de sembrado**



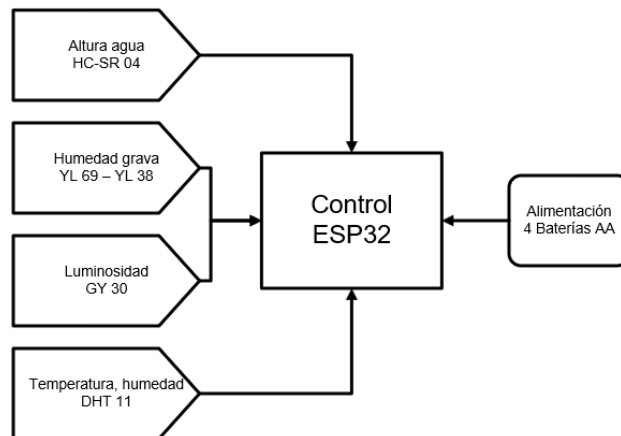
Fuente: elaboración propia, empleando Solidworks Student Edition.

3.2.1. Sensores de la cama de sembrado

Para el monitoreo de la cama de sembrado se realiza un módulo de sensores que lleva el control de las siguientes condiciones básicas:

- La humedad de la grava con el sensor YL - 69
- La altura de agua con el sensor HC-SR04
- Cantidad de luz que reciben las plantas con el sensor GY30
- Temperatura y humedad del ambiente con el sensor DHT11
- Alimentado por 4 baterías AA de 1,5 VDC, 2400 mAh cada una

Figura 48. Diagrama de bloques para el nodo de sensores para plantas

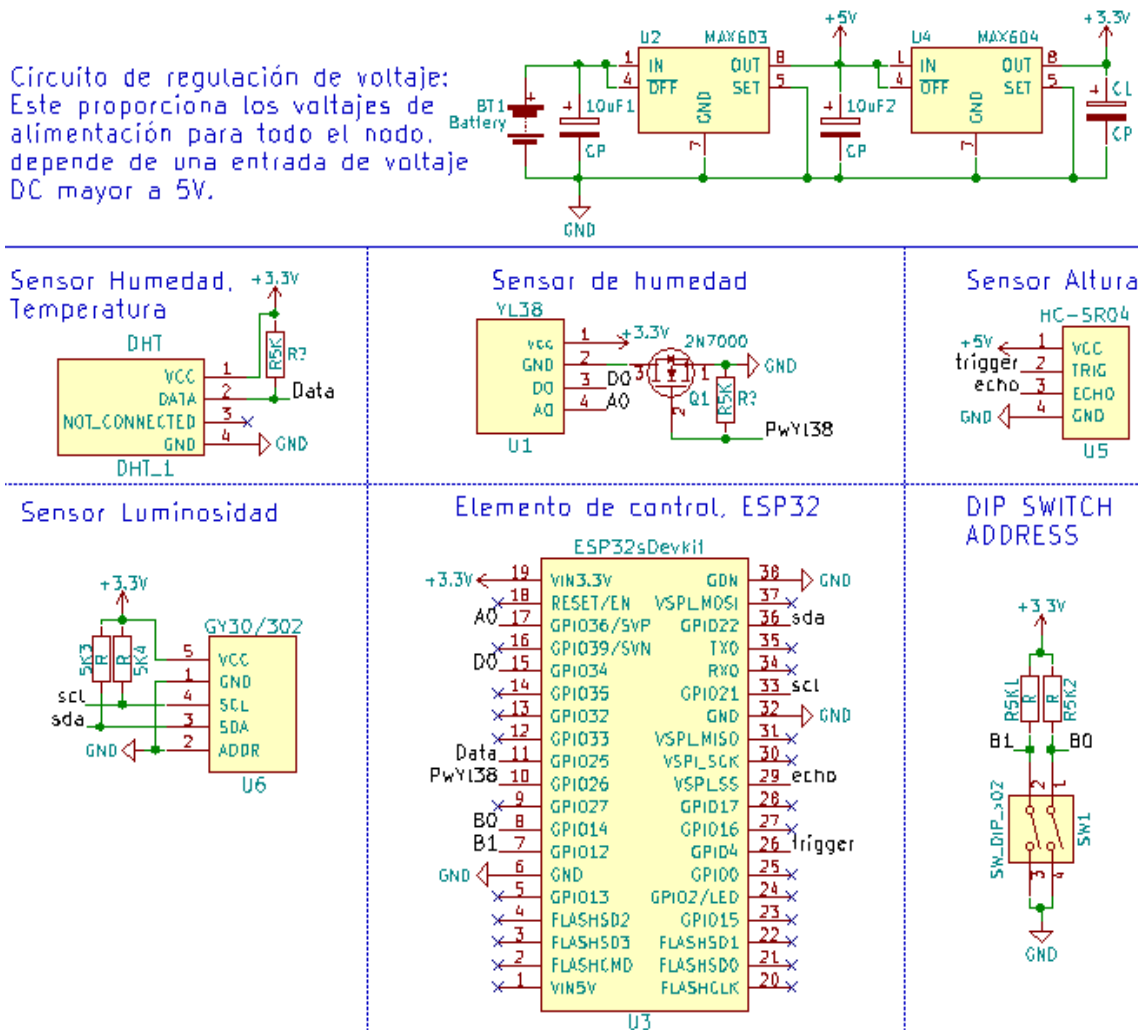


Fuente: elaboración propia, empleando Visio.

Se utiliza el microcontrolador ESP32 para el control de estos sensores, este se encarga de solicitar las mediciones de cada dispositivo, interpretarlas y procesarlas. Dado que este microcontrolador opera a 3,3VDC y algunos sensores operan a 5VDC, se implementa el mismo circuito de regulación de voltaje que se

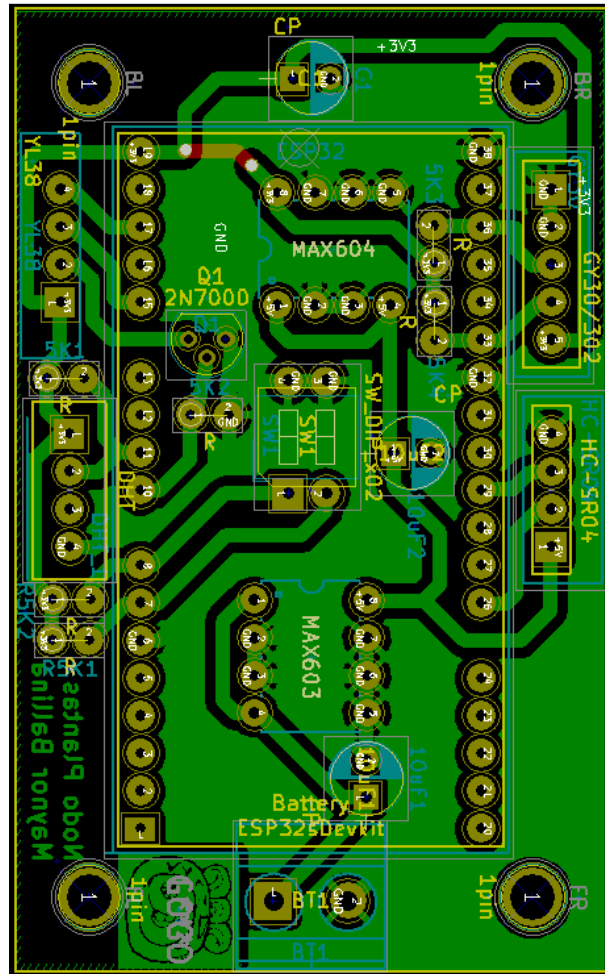
diseño para el nodo de sensores del tanque, además del mismo circuito para encendido y apagado para cada sensor. Se utilizó el programa KiCad para el diseño del esquema del circuito y su respectiva PCB.

Figura 49. Esquema del circuito para el nodo de sensores para plantas



Fuente: elaboración propia, empleando KiCad.

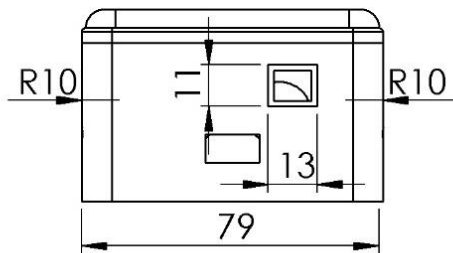
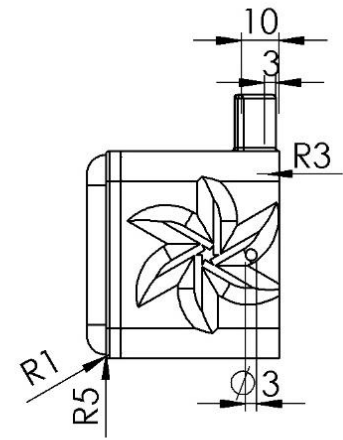
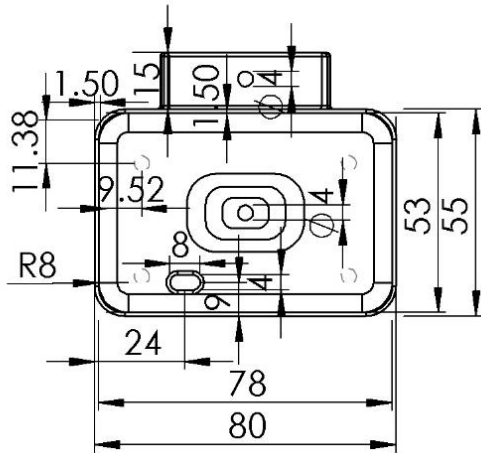
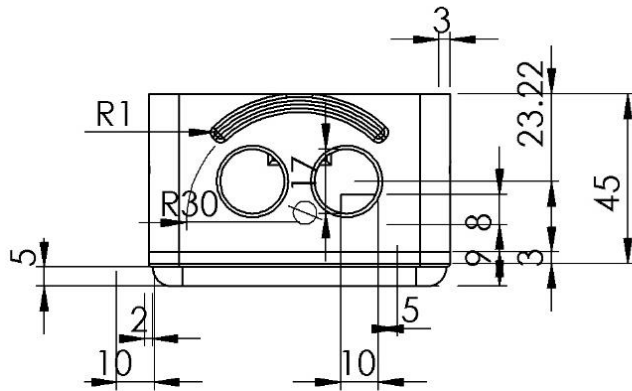
Figura 50. Esquema del PCB para el nodo de sensores para plantas



Fuente: elaboración propia, empleando KiCad.

Debido a condiciones de clima y que el nodo está cerca de agua, para que el PCB no se dañe y facilite su instalación en la cama de sembrado se diseña un contenedor, el cual se elaboró en ABS. Las mediciones están especificadas en la figura 51.

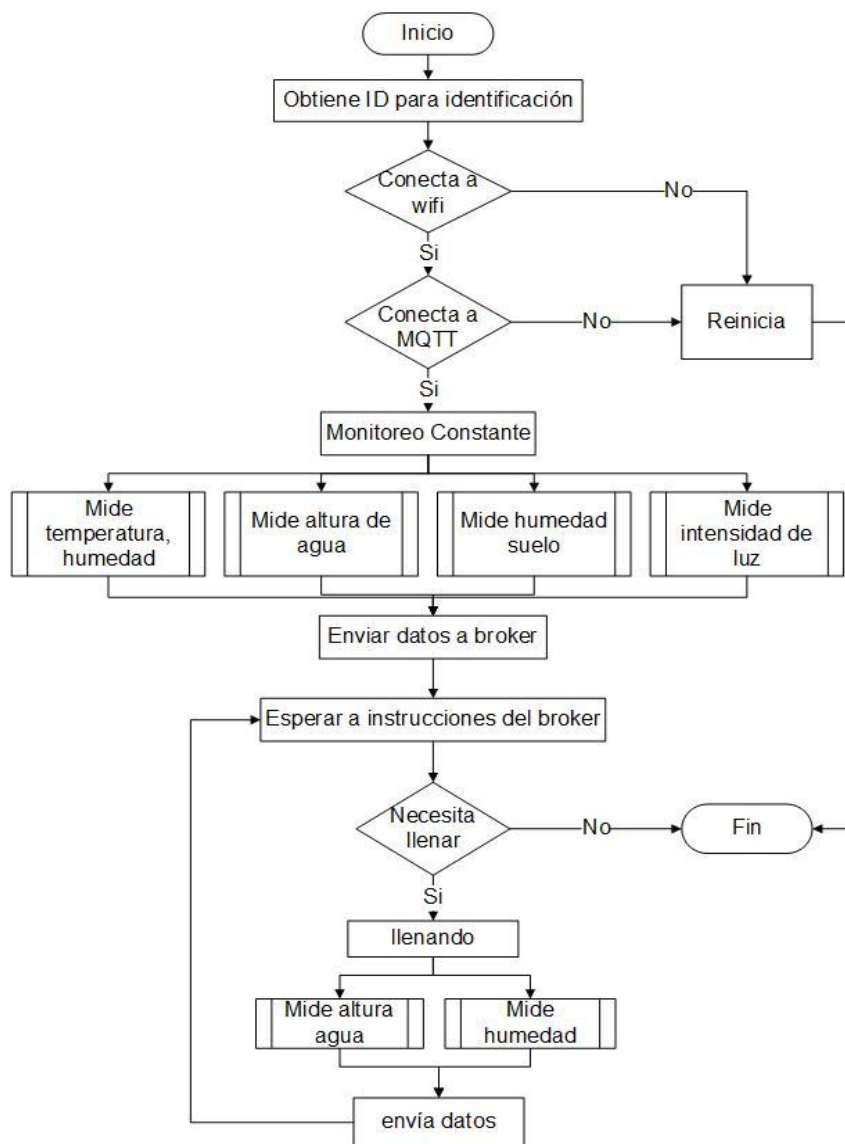
Figura 51. Caja de nodo de plantas, mediciones en mm



Fuente: elaboración propia, empleando Solidworks Student Edition.

Para realizar el control de los nodos de sembrado se realiza la programación por medio de la plataforma Arduino IDE, esta se puede encontrar en el área de apéndice. El diagrama de flujo define cómo se implementa el control del nodo.

Figura 52. **Diagrama de flujo del nodo de sembrado**

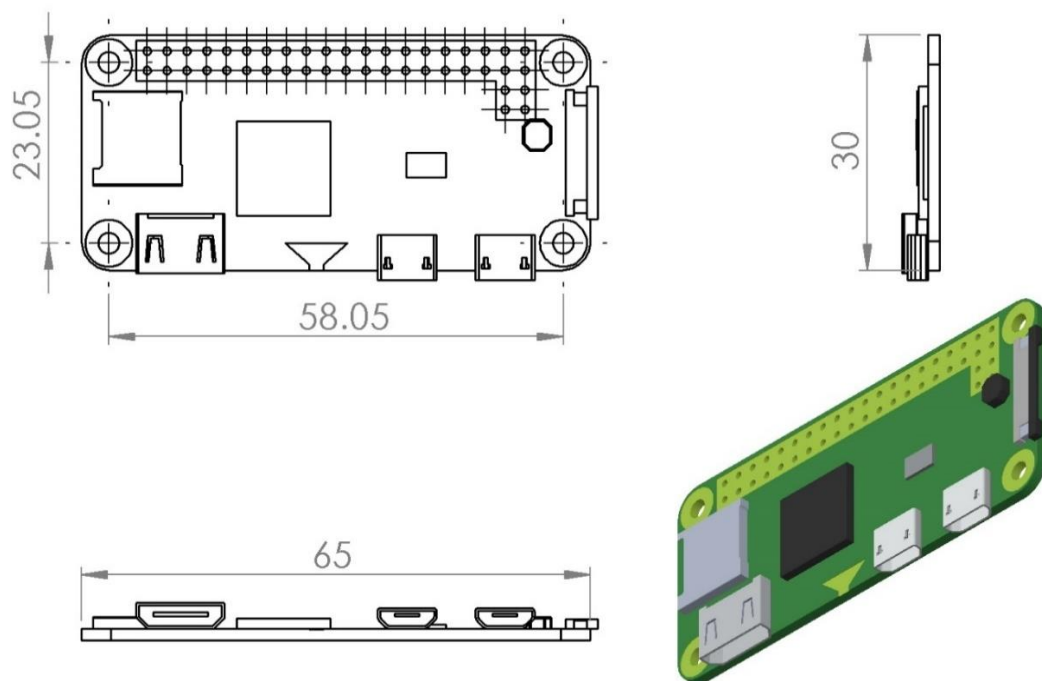


Fuente: elaboración propia, empleando Visio.

3.3. Módulo de control y almacenamiento

Para que los módulos de sensores estén comunicados entre sí y los datos obtenidos sean almacenados, se utiliza una Raspberry Pi Zero W, este es un computador de tamaño pequeño que contiene un sistema operativo Linux y tiene embebido un *chip* para conexión Wifi y Bluetooth, la implementación del *broker* MQTT se realiza por medio del *open source* software mosquitto, el cual implementa la versión 3,1 del protocolo MQTT.

Figura 53. Raspberry Pi Zero W

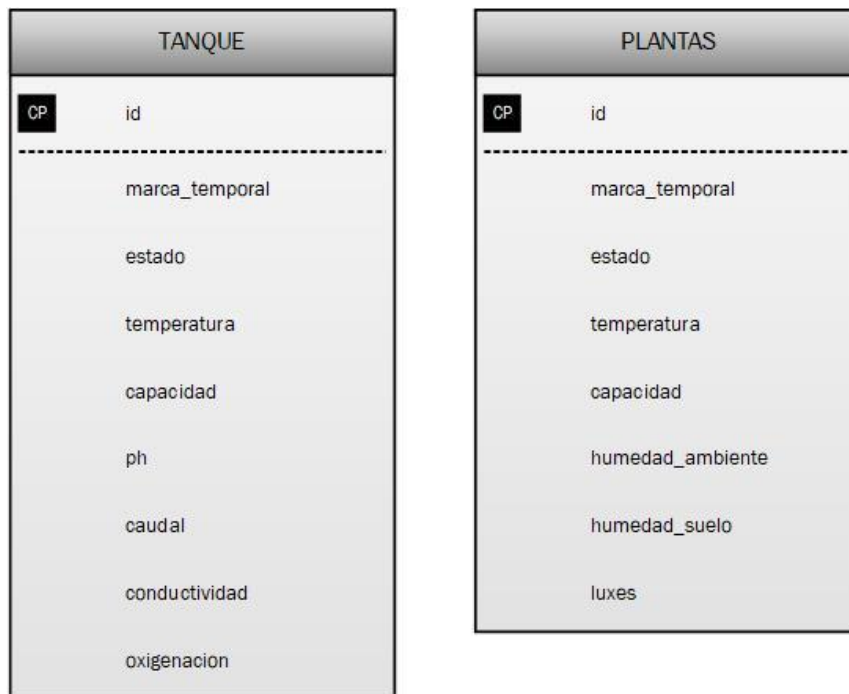


Fuente: elaboración propia, empleando Solidworks Student Edition.

Se almacena la información obtenida desde los módulos de sensores por medio de archivos de texto de forma local, almacenando cada dato en formato

de archivo Json. Además, se realiza una copia de esta información en la nube por medio de hojas de cálculo en Google Drive, la forma de almacenamiento está dada por el siguiente diagrama de entidades:

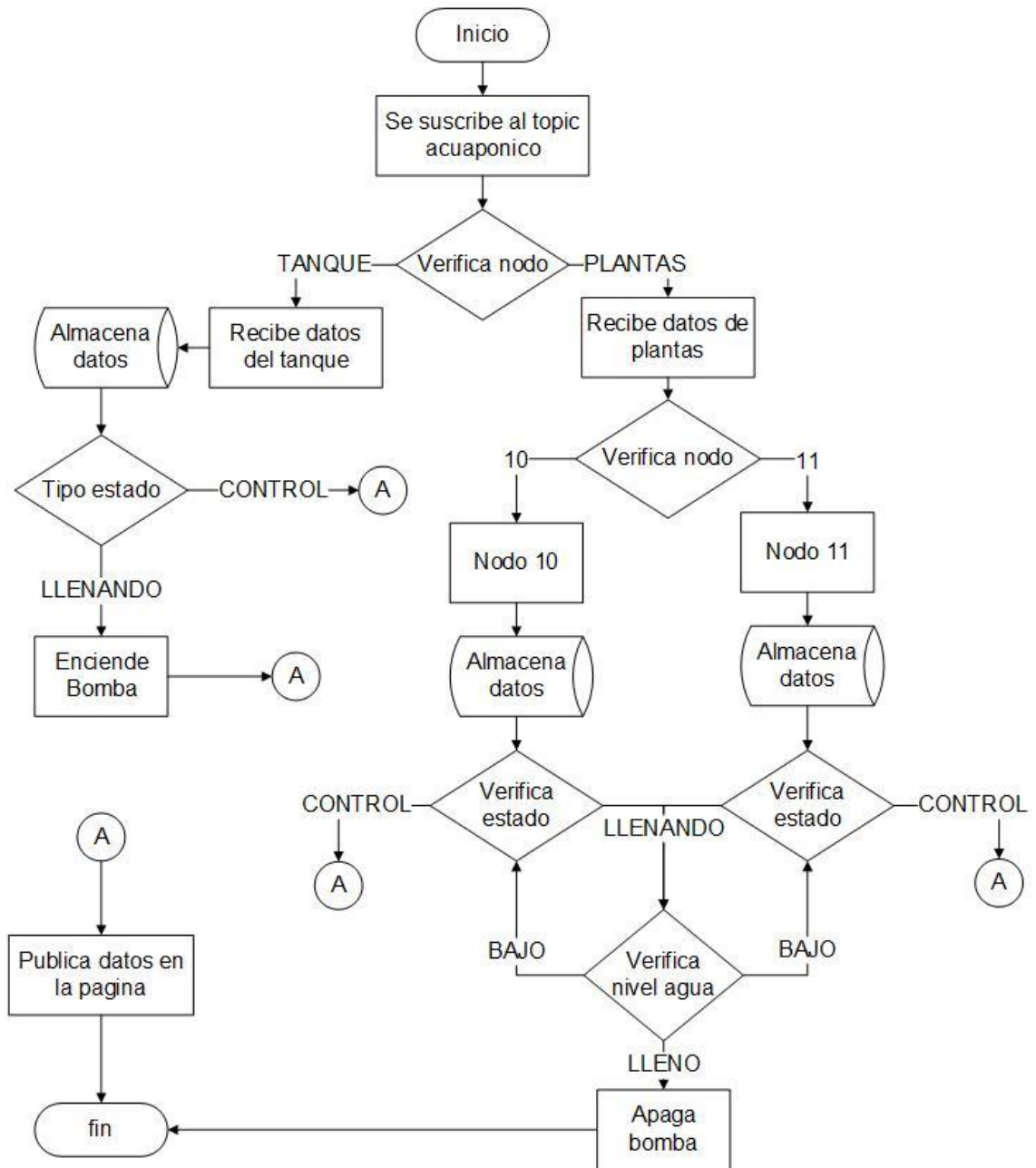
Figura 54. **Diagrama de entidades**



Fuente: elaboración propia, empleando Visio.

Con este dispositivo se implementa además del *broker* MQTT un servidor web, el cual despliega los datos actuales del sistema acuapónico. Para realizar el control de los nodos se realiza la programación en la plataforma Node-Red, esta programación es realizada en nodos propios del sistema, además de funciones programadas por medio de JavaScript, el código se puede encontrar en el área de apéndice en formato Json. El siguiente diagrama de flujo define cómo se implementa el control del sistema acuapónico:

Figura 55. Diagrama de flujo del nodo de control

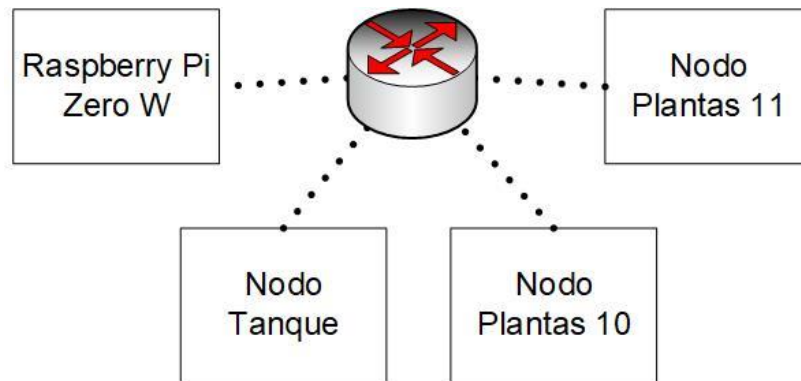


Fuente: elaboración propia, empleando Visio.

4. DISEÑO DEL SISTEMA DE COMUNICACIÓN

Para la comunicación entre cada nodo se utiliza el estándar IEEE 802.11 g/n (Wifi), por medio de un enrutador de hogar, para este caso se utiliza el enrutador Nexxt 01 V3, cada microprocesador contiene un circuito integrado que facilita la conexión entre ellos vía este protocolo de comunicación, se utiliza una topología de conexión de tipo estrella, el enrutador asigna una dirección IP a cada nodo con base en el protocolo DHCP, el único elemento que tiene una dirección IP estática es el *broker*.

Figura 56. Diagrama de conexión hacia el *router*

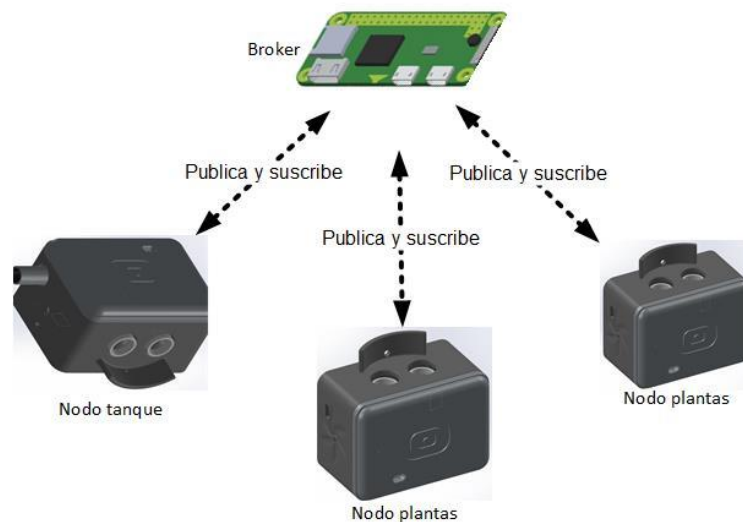


Fuente: elaboración propia, empleando Visio.

La transmisión de datos se realiza por medio del protocolo de comunicación MQTT, a pesar de estar basado en el protocolo TCP, este tiene una topología de comunicación diferente a la conexión física, se trabaja con base en una topología de tipo estrella para MQTT, se utiliza la Raspberry Pi Zero W como el *broker* y, como clientes, los módulos de sensores. Para una mejor identificación los

módulos que miden los datos de los tanques del sistema acuapónico se denominan “nodo tanque” y los módulos que realizan mediciones de las camas de sembrado donde se tienen las plantas se identifican como “nodo plantas”. La comunicación es enviada por medio del protocolo TCP utilizando el puerto 1883, el cual es un puerto donde se envían los mensajes sin encriptación, dado que es un prototipo dentro de una red local, no será necesario encriptar, por medio de MQTT cada dispositivo publica y se suscribe al *broker*, así como este publica y se suscribe a cada nodo.

Figura 57. Diagrama de conexión MQTT, de *broker* a clientes

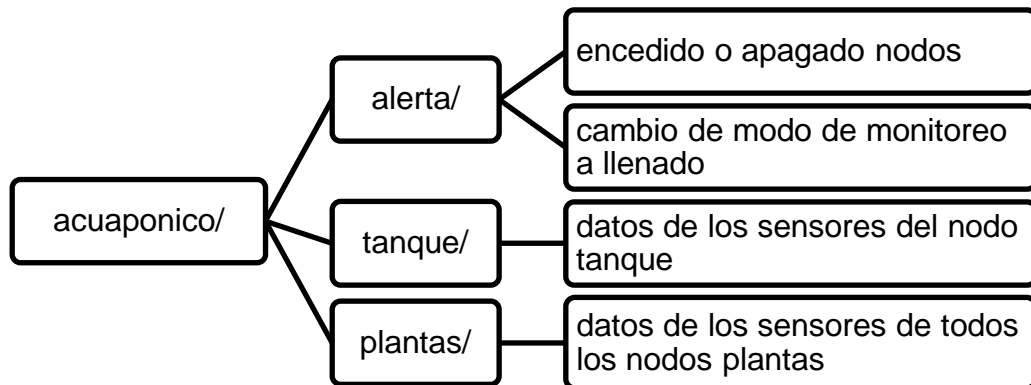


Fuente: elaboración propia, empleando Visio.

El protocolo de comunicación MQTT está basado en publicaciones y suscripciones a *topics*, los cuales definen y separan el tipo de mensajes con los cuales se controla el sistema.

Para el sistema acuapónico el *topic* principal es *acuaponico/*, el cual contiene toda la información necesaria del sistema, este se subdivide en tres *topics* más: */tanque/*, que contiene toda la información sobre el nodo tanque y sus mediciones realizadas; */plantas/*, que contiene toda la información sobre todos los nodos plantas y sus mediciones realizadas, y */alerta/*, que utiliza el *broker* para comunicarse con todos los nodos.

Figura 58. Diagrama de *topics* utilizados



Fuente: elaboración propia.

4.1. Mensaje MQTT

Para evitar el uso innecesario de varios *topics* y optimizar la comunicación, se implementa el envío de cada mensaje en formato Json, este formato permite una mejor estructura al mensaje y fácil interpretación, el contenido del mensaje es identificado por medio de la palabra *payload*, dentro de este se utiliza el identificador tipo para dos condiciones de trabajo, “llenando” y “control”. Todos los nodos envían los mensajes con una calidad de servicio 0, sin retención, el *broker* los envía con una calidad de servicio de 2 y sin retención.

Topic acuapónico/alerta: el *broker* se encarga de dar un aviso a todos los nodos, bajo un único mensaje dentro del *payload* y es formado por una de las palabras: “lleno”, “vacío” o “apagar”.

- El mensaje “lleno” se utiliza para finalizar el modo “llenando”, por lo cual el *broker* solo lo genera cuando recibe que los nodos están en dicho modo, con base en los datos obtenidos por los sensores el mensaje es emitido al cumplir una de las siguientes condiciones:
 - El nodo tanque no está conectado.
 - Ningún nodo plantas están conectados.
 - La altura del agua dentro del tanque es menor a 15 cm.
 - pH del agua es mayor a 7.
 - La cantidad de agua dentro de cualquier nodo plantas es mayor a 90 % de la capacidad del contenedor.

- El mensaje “vacío” se utiliza para iniciar o mantener el modo “llenando”, lo envía el *broker* con base en mediciones de todos los nodos, es emitido al cumplir una de las siguientes condiciones:
 - pH del agua es menor a 6.
 - El sensor de altura de agua del nodo funciona y el porcentaje de humedad de la grava menor al 10 %.
 - En el modo “llenando”, si la cantidad de agua dentro del contenedor es menor al 90 %.

- El mensaje “apagar” se envía al final de la recolección de datos del modo “monitoreo constante”, a excepción de que cumpla con las condiciones para enviar el mensaje “vacío”.

Figura 59. Mensaje en json del *broker* para todos los estados

```
{
  "_comment": "sale modo llenando",
  "topic": "acuaponico/alerta/",
  "payload": "lleno"
}

{
  "_comment": "entra o mantiene modo llenando",
  "topic": "acuaponico/alerta/",
  "payload": "vacio"
}

{
  "_comment": "apaga todos los nodos",
  "topic": "acuaponico/alerta/",
  "payload": "apagar"
}
```

Fuente: elaboración propia, empleando Atom.

Topic acuapónico/tanque: el nodo tanque se encarga de enviar datos medidos al *broker*, el *payload* contiene la identificación del nodo y el tipo del mensaje, definido como:

- Control constante: está identificado con el tipo “control”, bajo esta condición el *payload* contiene:
 - Estados: contiene los estados de cada sensor:
 - 1: funciona de forma correcta.
 - 0: no conectado o dañado.
 - Valores: este vector contiene los valores individuales de las mediciones realizadas.

Figura 60. Mensaje en json del control contante del tanque

```
{
  "topic": "acuaponico/tanque/",
  "payload": {
    "NodoID": "NTanqueXX",
    "tipo": "control",
    "estados": [
      "DS19B20",
      "HC-SR04",
      "YL38",
    ],
    "valores": [
      "temperatura agua °C",
      "altura agua cm",
      "conductividad agua S/cm",
      "pH",
      "rssi"
    ]
  },
  "qos": 0,
  "retain": false
}
```

Fuente: elaboración propia, empleando Atom.

- Llenando camas de sembrado: está identificado con el tipo "llenando", bajo esta condición el tanque, cada vez que recibe el mensaje "vacío" del *broker*, envía la altura del agua en el tanque y el caudal al que sale el agua del mismo.

Figura 61. Mensaje en json del llenado desde el tanque

```
{
  "topic": "acuaponico/tanque/",
  "payload": {
    "NodoID": "NTanqueXX",
    "tipo": "llenando",
    "nivel": "altura agua cm",
    "caudal": "Caudal l/minutos"
  },
  "qos": 0,
  "retain": false
}
```

Fuente: elaboración propia, empleando Atom.

Topic acuapónico/plantas: el nodo plantas trabaja bajo la misma estructura del nodo tanque, en el cual solo varía el tipo de datos medidos y la forma de identificación. Los nodos plantas se encargan de enviar datos medidos al *broker*, el *payload* contiene la identificación del nodo y el tipo del mensaje, este se regula bajo dos condiciones:

- Control constante: está identificado con el tipo "control", bajo esta condición cada nodo plantas envía:
 - Estados: contiene los estados de cada sensor:
 - 1: funciona de forma correcta.
 - 0: no conectado o dañado.
 - Valores, este vector contiene los valores individuales de las mediciones realizadas.

Figura 62. Mensaje en json del control contante del nodo plantas

```
{
  "topic":"acuaponico/plantas/",
  "payload":{
    "NodoID":"NPlantasXX",
    "tipo":"control",
    "estados":[
      "DHT11",
      "HC-SR04",
      "YL38",
      "BH1750"
    ],
    "valores":[
      "humedad ambiente %",
      "temperatura ambiente °C",
      "altura agua %",
      "humedad grava %",
      "luminosidad lux"
      "rssi"
    ]
  },
  "qos":0,
  "retain":false,
}
```

Fuente: elaboración propia, empleando Atom.

- Llenando camas de sembrado: está identificado con el tipo "llenando", bajo esta condición cada nodo plantas, al recibir el mensaje "vacío" del *broker*, envía el porcentaje de altura del agua dependiendo de la capacidad máxima del contenedor y el porcentaje de humedad del sustrato.

Figura 63. **Mensaje en json del llenado del nodo plantas**

```
{
  "topic": "acuaponico/plantas/",
  "payload": {
    "NodoID": "NPlantasXX",
    "tipo": "llenando",
    "nivel": "altura %",
    "humed": "humedad grava %"
  },
  "qos": 0,
  "retain": false,
}
```

Fuente: elaboración propia, empleando Atom.

5. RECOLECCIÓN DE DATOS Y CONTROL DE LOS NODOS

El control del sistema acuapónico depende de las mediciones realizadas por cada nodo, las decisiones tomadas por el *broker* son dependientes del conjunto de todos los datos del sistema, este determina el funcionamiento de cada nodo además de los cambios necesarios, siempre que el sistema necesite la atención necesaria genera mensajes de alerta con la descripción de los elementos a tomar en cuenta y los envía por medio de correo electrónico.

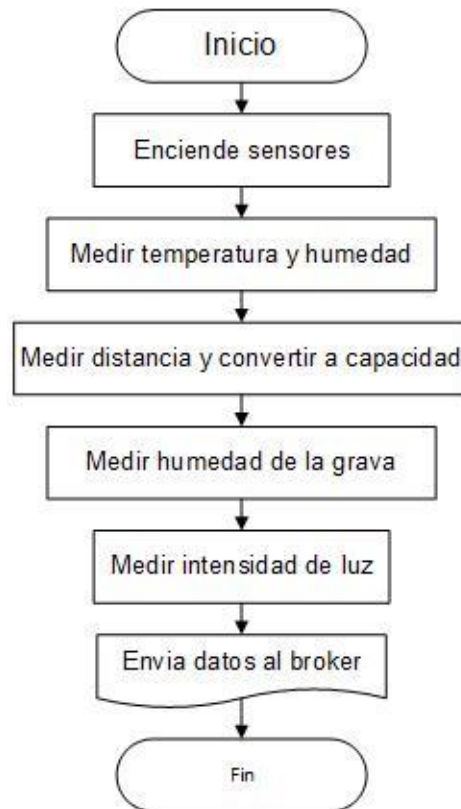
5.1. Nodo plantas

Inicia realizando un monitoreo constante de la cama de sembrado y sus condiciones, envía el dato al *broker* y espera el *topic* de alerta, en el momento en que el *broker* envía el mensaje de alerta sistema “vacio” este cambia al modo llenando, de lo contrario si el *broker* envía el mensaje “apagar”, descansa hasta el momento de realizar la siguiente medición esperando 30 minutos.

5.1.1. Monitoreo constante

El nodo obtiene directamente los valores de cada sensor, los que necesitan una interpretación diferente son transformados al valor necesario, el siguiente diagrama muestra el orden en el cual se realizan las mediciones seguido de la explicación de cada proceso:

Figura 64. **Diagrama de flujo de monitoreo constante**



Fuente: elaboración propia, empleando Visio.

- Temperatura ambiente: el sensor la obtiene en grados Celsius (°C), se realizan 3 mediciones y el resultado final es el promedio de las tres.
- Humedad del ambiente: el sensor obtiene el porcentaje de humedad relativa (%HR), se realizan 3 mediciones y el resultado final es el promedio de las tres.
- La medición de la altura del agua depende de la distancia en centímetros entre el sensor hasta el contacto con el agua, se realizan 5 mediciones y

el promedio de estas es la distancia entre el sensor y el agua. Dado que el contenedor tiene una altura máxima de 14 cm y el sensor estará en esta posición, se mide el valor máximo de agua deseado para el contenedor con respecto a la altura de la grava, esto da una distancia de 10 cm para el 100 % de su capacidad. Para el dato de referencia de altura del agua en el sistema acuapónico se toma el porcentaje con respecto al valor máximo de agua necesario, calculado con la ecuación 5.1.

$$\% \text{ capacidad} = (14 - \text{distancia}) * 10 \quad (5.1)$$

- La humedad del suelo depende de un voltaje analógico y la lectura de este con el microcontrolador. Se toma de referencia los valores experimentales:
 - Grava seca; valor analógico igual a 4095
 - Grava inundada; valor analógico igual a 977

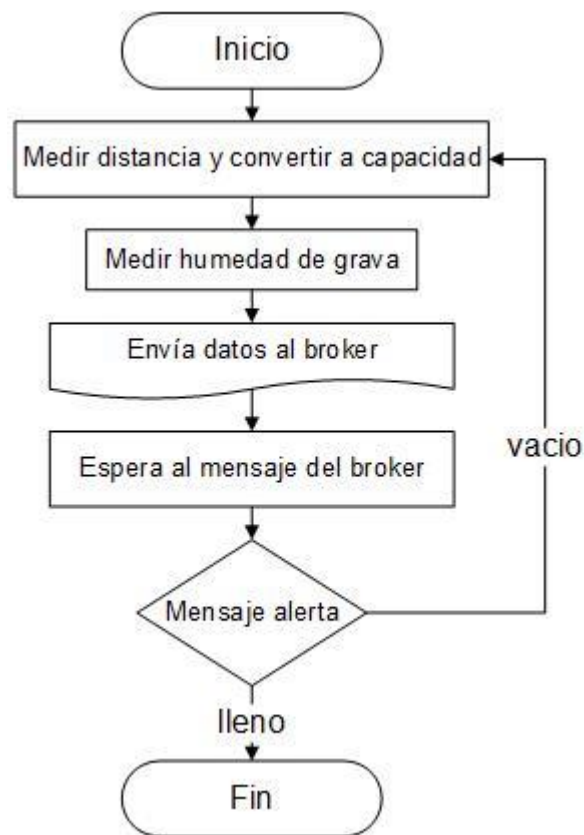
Para mejorar la exactitud de la medición se realizan 15 mediciones y se promedia el valor de las mediciones, utilizando la función *map* se puede hacer una conversión entre los valores medidos hacia los que serán enviados al *broker*, reordenando los rangos de la siguiente forma:

- Analógico 4095 = 0 %H
 - Analógico 977 = 100 %H
- Luminosidad: el sensor entrega la medición en lux. Se realizan 5 mediciones, el valor final es el promedio de estas mediciones.

5.1.2. Llenando

Para este modo el sensor realiza la medición de la capacidad del contenedor y la humedad de la grava, descrito en la sección anterior, enviando esta información cada vez reciben la alerta “vacio” y esperando que el *broker* cambie el mensaje de alerta hacia nivel “lleno”, al llegar este mensaje el nodo vuelve a descansar por 30 minutos hasta la siguiente medición necesaria.

Figura 65. Diagrama de flujo del modo llenando



Fuente: elaboración propia, empleando Visio.

5.2. Nodo tanque

Lo primero que realiza el nodo es la medición del pH del agua, esta medición se realiza primero dado que según la hoja de especificaciones del controlador ESP32¹⁹. El GPIO15 ADC está conectado al módulo de Wifi por medio de EMAC_TXD3, el cual contiene parte de la dirección MAC del dispositivo, que al estar encendido no permite realizar mediciones analógicas.

- La medición del pH se realiza bajo los parámetros establecidos por el fabricante en su hoja de especificaciones, el cual provee una ecuación que depende del voltaje medido, el voltaje se obtiene por medio de la ecuación 5.2, donde la variable de medición es el valor analógico que provee el microcontrolador, con un valor máximo de 4905 y el valor máximo de voltaje que es 3,3, debido a que el valor de voltaje que se obtiene del sensor es de 0 a 3,3 V y la ecuación 5.4 depende del voltaje variable de 3,3 a 0 V, entonces se realiza un ajuste del voltaje por medio de la ecuación 5.3.

$$voltaje_{medido} = \frac{pH_{analógico}}{4095} * 3300 \quad (5.2)$$

$$voltaje_{correcto} = 3300 - voltaje_{medido} \quad (5.3)$$

$$pH_{no\ calibrado} = (voltaje - 1500)/3,0 \quad (5.4)$$

A continuación, ingresa al modo monitoreo constante del tanque, envía el dato al *broker* y espera el *topic* de alerta; en el momento en que el *broker* envía

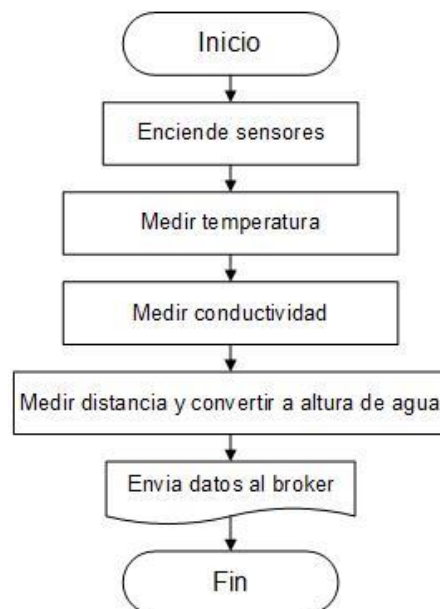
¹⁹ ESPRESSIF. *ESP32, Technical Reference Manual*. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Consulta: 09 de junio de 2018.

el mensaje de alerta “vacio” este cambia al modo llenando, de lo contrario al recibir “apagar” descansa por 30 minutos.

5.2.1. Monitoreo constante

El nodo obtiene directamente los valores de cada sensor, los que necesitan una interpretación diferente son transformados al valor necesario, el siguiente diagrama de flujo muestra el proceso de medición seguido de la descripción de cada proceso:

Figura 66. Diagrama de flujo de monitoreo constante



Fuente: elaboración propia, empleando Visio.

- Temperatura del agua: el sensor la obtiene en grados Celsius (°C).
- La medición de la altura del agua depende de la distancia en centímetros entre el sensor hasta el contacto con el agua, se realizan 5 mediciones de

esta distancia y se obtiene el promedio de estas mediciones. El tanque tiene una altura máxima de 30 cm y el sensor estará en esta posición, se determina el valor de la altura máxima del agua de 25 cm para el 100 % de su capacidad. Se obtiene la altura del agua en el tanque por la siguiente ecuación:

$$altura = (30 - distancia) \quad (5.5)$$

Esta medición se realiza para generar una alerta en el caso de tener menos de 15 cm de altura del agua en el tanque.

- La conductividad del agua depende de un voltaje analógico y la lectura de este con el microcontrolador. Se realizan 10 mediciones del dato analógico y se promedia este valor. Para obtener el valor de voltaje con respecto al dato analógico, se utiliza la ecuación 5.6, la conductividad se obtiene aplicando la ecuación 5.7, esta se describe en el área del apéndice.

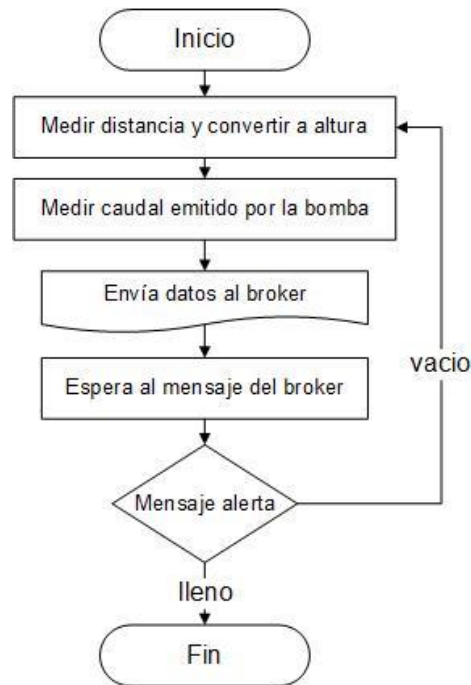
$$voltaje_{medido} = \frac{Valor_{analógico}}{4095} * 3,3 \quad (5.2)$$

$$Conductividad = \left(\frac{3,3 - voltaje_{medido}}{voltaje_{medido} * R1} \right) * 104,167 \quad (5.3)$$

5.2.2. Llenando

Este modo se integra de dos elementos importantes, la medición de caudal y la operación de la bomba de agua, se activa por medio del mensaje de alerta “vacío” del *broker* y se desactiva por medio del mensaje de “lleno”, reenviando esta información al *broker*, siempre que reciba el mensaje “vacío”.

Figura 67. Diagrama de flujo de llenado



Fuente: elaboración propia, empleando Visio.

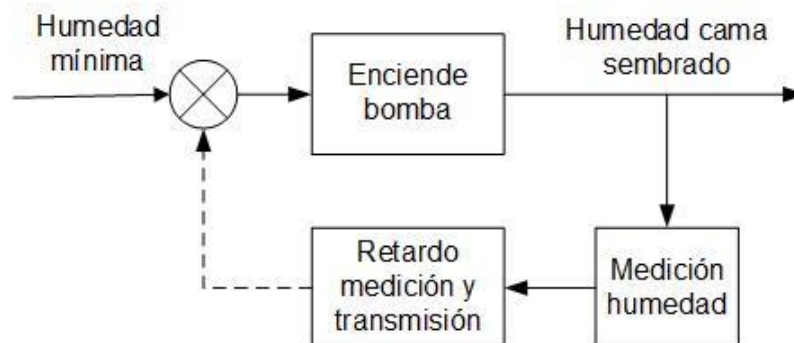
- El caudal se obtiene por medio de los parámetros establecidos por el fabricante y la ecuación 2.6, como resultado de esto el caudal está dado en Litros por minuto (l/m). El caudal determina la condición de la bomba de agua, si la medición de caudal se reduce más de un 50 % del caudal inicial, el filtro de la bomba de agua necesita limpieza.
- Mide la altura del agua tal como se describe en la sección anterior.
- La activación de la bomba se realiza por medio de un relé de estado sólido, el cual permite ser controlado por el microcontrolador y el consumo de batería sea bajo.

5.3. Broker

Este nodo es el elemento de control central. Obtiene valores de cada nodo de sensores y toma decisiones a largo tiempo con base en dos parámetros básicos:

- Control de humedad de las plantas: debido a que la capacidad de retención de agua de la piedra pómez es del 18 %²⁰, es necesario controlar el valor de humedad en las camas de sembrado, este está determinado por el siguiente diagrama de control:

Figura 68. Diagrama de control de humedad



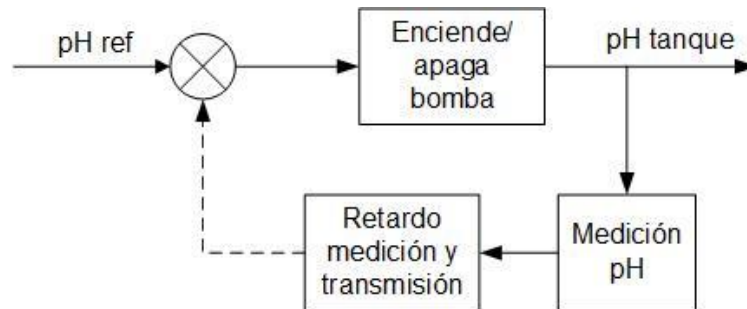
Fuente: Elaboración propia empleando Visio.

- Control de pH del tanque: debido a la importancia del control del pH en el sistema acuapónico, se debe mantener un rango entre 6 y 7,²¹ con base en estos parámetros el pH será controlado por el siguiente diagrama de control:

²⁰ BARBARO, Lorena; et al. *Sustrato para techos verdes sustentables*. p. 10.

²¹ SOMERVILLE, Christopher; et al. *Small-scale aquaponic food production – Integrated fish and plant farming*. p. 170.

Figura 69. Diagrama de control de pH



Fuente: elaboración propia, empleando Visio.

- pH ácido: el valor es menor a 6, el proceso de nitrificación decrece, dificultando el acceso de las plantas a nutrientes. La forma natural en la cual el pH aumenta de forma moderada, incrementando la oxigenación del agua²², la forma en la que se puede aumentar la oxigenación del agua es por reaeración²³, esto se cumple con circulación de agua del sistema.
- pH inicia a ser alcalino: el valor es mayor a 7, se empieza a notar el incremento en amoníaco (NH_3) dentro del agua, al momento en el que llega a un pH de 8 se tiene un 50 % de amoníaco en el agua, siendo este altamente tóxico para los peces²⁴, por lo cual no se realizará circulación de agua, así de forma natural el pH desciende en pequeñas variaciones.

Con base en el control esperado por el sistema acuapónico, el *broker* implementa dos entornos, al igual que cada nodo.

²² SOMERVILLE, op. cit., p. 24

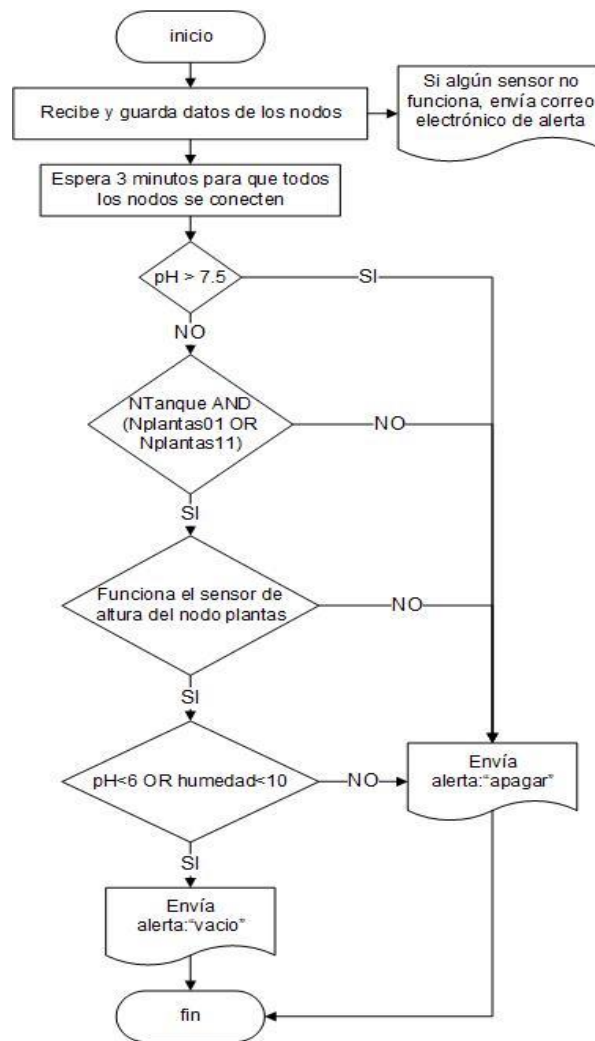
²³ Water Boards, op. cit., p. 3

²⁴ *Ibidem*.

5.3.1. Monitoreo constante

Al recibir los valores de los nodos, los compara con los valores de referencia y con base en las condiciones se establece el algoritmo descrito en el diagrama de flujo, que implementa el control del sistema acuapónico.

Figura 70. Diagrama de flujo, control de monitoreo constante



Fuente: elaboración propia, empleando Visio.

Al recibir el primer mensaje de los nodos, el *broker* espera por 3 minutos que todos los nodos se conecten, mientras espera si un sensor en los nodos tiene algún tipo de error, se envía un correo electrónico con el tipo de alerta que se generó y la identificación del nodo que envió dicha alerta. Este sistema de control es el que activa el modo “llenando” para todos los elementos, al cumplirse las siguientes condiciones:

- pH menor a 7
- El nodo tanque y algún nodo plantas debe estar conectado
- Los sensores de altura deben estar en correcto funcionamiento
- Humedad de la grava en la cama de plantas menor al 10 %

En caso contrario envía la alerta “apagar” a todos los nodos conectados, para que los controladores ESP32 entren dentro del modo *deep sleep*, por un período de media hora.

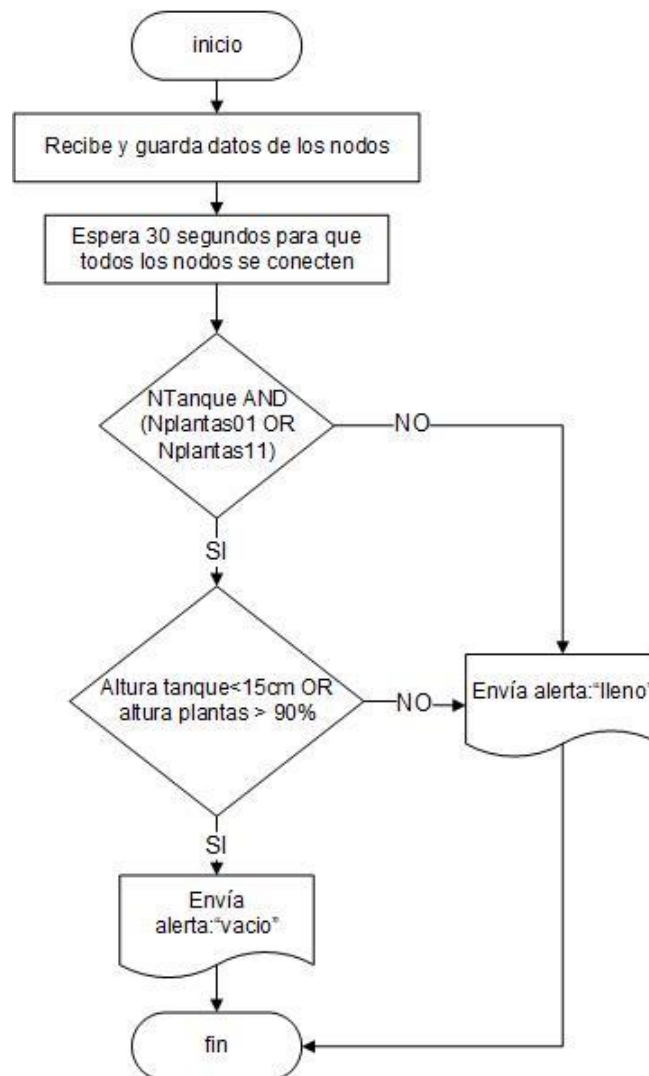
5.3.2. Llenando

Al recibir el primer mensaje de los nodos, el *broker* espera por 30 segundos que todos los nodos se conecten. Este sistema de control es el que mantiene activo el modo “llenando” para todos los elementos, al cumplirse las siguientes condiciones:

- El nodo tanque y algún nodo plantas debe estar conectado.
- La altura del agua del contenedor de plantas es menor al 90 % de su capacidad.
- La altura del agua del tanque es mayor a 15 cm de altura.

En caso contrario envía el mensaje “lleno” a todos los nodos conectados, estos salen del modo de llenado y los controladores ESP32 entran dentro del modo *deep sleep*, por un período de media hora.

Figura 71. Diagrama de flujo, control de modo llenando



Fuente: elaboración propia, empleando Visio.

6. PROTOTIPO DEL SISTEMA ACUAPÓNICO

Con base en el sistema propuesto, se implementa el prototipo de un sistema acuapónico para interiores, la calibración y toma de datos se documenta de forma gráfica. Los materiales utilizados para el prototipo se listan a continuación:

Tabla IV. **Lista de materiales varios**

Cantidad	Descripción
1	Pecera de Vidrio (30*50*22) cm
2	Caja de plástico (35,5*25,5*14) cm
1 m	Tubo PVC de 4"
1 m	Tubo PVC de 2" ø
1 m	Tubo PVC de 1" ø
1 m	Tubo CPVC de ½" ø
1 m	Manguera de ½" ø
2	Codo CPVC de ½" ø
1	T CPVC de ½" ø
2	Codo PVC de 1" ø
2	Reductor PVC de 1" a ½" ø
2	Adaptador Rosca ¾" a tubo ½"
1	Tubo pegamento PVC, Duraman, 25 gr.
1	Bote pegamento de madera
20	Clavos de 1 ½"
2 m	Tabla de pino 6 cm ancho por 2 cm grosor
1	Costal de piedra pómez para jardinería.
2	Plantas de albahaca
2	Plantas de hierbabuena
5	Peces (tilapia)
1	Bomba para acuario, Kintons KT101
1	Bote pegamento de madera
1	Rollo cinta aislante
3	Contenedor impreso en ABS

Fuente: elaboración propia.

Tabla V. **Lista de materiales electrónicos**

Cantidad	Descripción
1	Placa de cobre y fibra de vidrio (20*20) cm
2	Estañó para soldar
3	Microcontrolador ESP32S Dev Board
3	MAX603
3	MAX604
2	Contenedor para 4 baterías AA
1	Contenedor para 6 baterías AA
6	Capacitor electrolítico 10uF
3	Capacitores de tantalio 1 nF
4	Terminales de dos conectores
3	DIP Switch de cuatro pines.
3	Resistencias 10 K Ω
15	Resistencias 5,1 K Ω
4	Transistores Mosfet 2N7000
7	Tira de 20 pines IDE hembra
3	Tira de 20 pines IDE macho
3	YL38 / YL69
3	HC-SR04
2	DTH11
2	GY302
1	DS18B20
1	pH meter SKU: SEN0161
1	Cable de corriente tipo 8 OEM
1	G3MB relé de estado sólido.
1	FS300A
1	Cable UPT CAT 6
1	Raspberry Pi Zero W
1	Fuente de alimentación Micro USB 120 VAC a 5VDC

Fuente: elaboración propia.

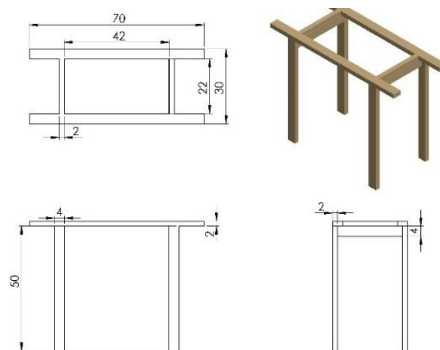
6.1. Contenedores e inmobiliario

A continuación, se ilustra por medio de fotografías, el proceso del armado de la parte física del prototipo.

6.1.1. Mesa de soporte

Esta es una mesa realizada en pino y pintada para evitar que se dañe al estar en contacto con agua.

Figura 72. Plano de la mesa en cm



Fuente: elaboración propia, empleando Solidworks Student Edition.

Figura 73. Mesa de soporte



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

6.1.2. Tanque

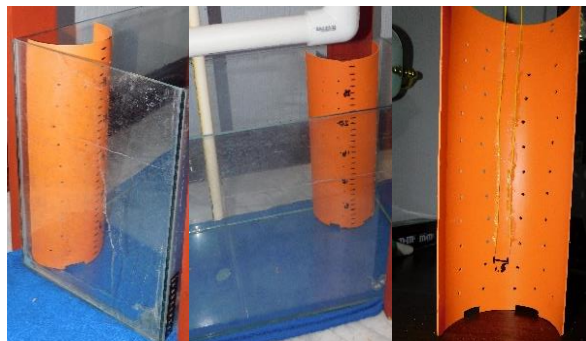
Con base en las especificaciones de la figura 38, se utiliza una pecera de vidrio con las mismas dimensiones, la cual tiene un separador de PVC, descrito en la sección 3.1, para las puntas de prueba y evitar contacto directo con los peces.

Figura 74. **Tanque vacío**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

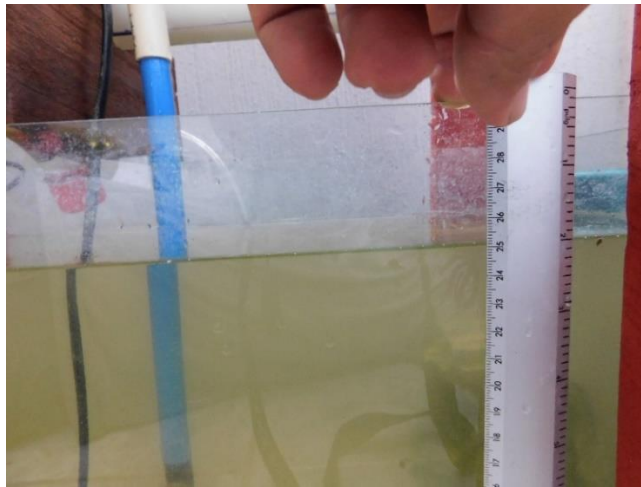
Figura 75. **Tanque con separador**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

El agua inicial del sistema estará a una altura máxima de 25 cm, siendo 15 cm la menor altura aceptable. La cantidad de peces utilizada es de 5 tilapias.

Figura 76. **Altura inicial del agua**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 77. **Tilapias para el sistema acuapónico**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

6.1.3. Camas de sembrado

Con base en las especificaciones brindadas por la sección 3.2, se utiliza el recipiente plástico descrito en la figura 46, al igual que el tubo de PVC para poder colocar el nodo plantas y realizar la medición de altura sin obstrucciones, al centro de este se realiza un agujero de 1" de diámetro para el drenaje, acá va situado el reductor PVC, este va a un tubo PVC que conecta ambos nodos y el drenaje, como se muestra en la figura 81, el tubo tiene 3 puntos de salida, para que el agua regrese por gravedad al tanque de peces. Se utilizan dos camas de sembrado, una para sembrar *Ocimum basilicum* "albahaca" y otra para *Mentha spicata* "hierbabuena", por lo cual para poder diferenciar se utilizan dos colores diferentes de contenedor, rojo y azul.

Figura 78. **Contenedor vacío**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 79. **Separador y tanque con separador**



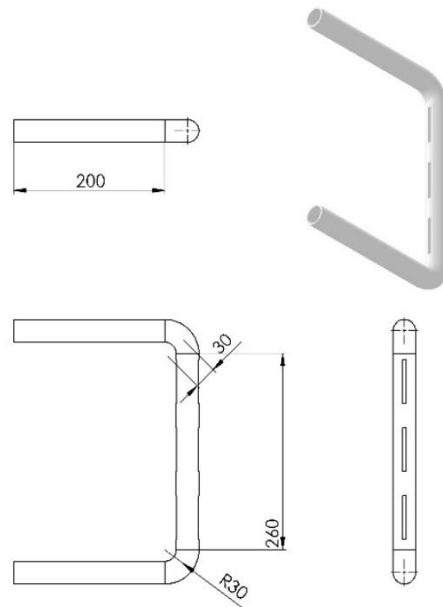
Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 80. **Dos contenedores instalados en la mesa de soporte**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 81. **Plano del drenaje en mm**



Fuente: elaboración propia, empleando Solidworks Student Edition.

Figura 82. **Contenedor con reductor y drenaje instalado**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 83. **Contenedores con piedra pómez**

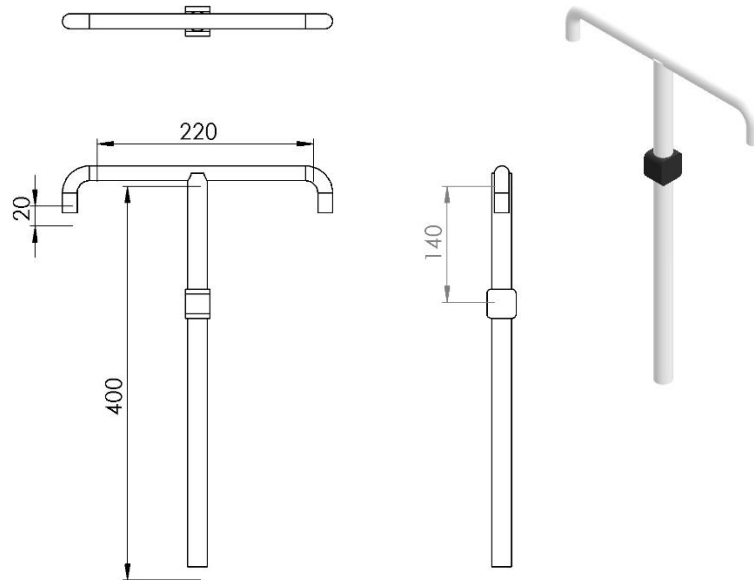


Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

6.1.4. Bomba de agua

Para la instalación de la bomba de agua se diseña un sistema de distribución de agua para las dos camas de plantas con tubo CPVC de $\frac{1}{2}$ de pulgada, al medio de esta instalación estará conectado el sensor de caudal FS300A, debido a que la altura a la que se encuentran las plantas no excede un metro, esto cumple con las especificaciones de la bomba de agua especificadas en la sección 3.1.1., la parte que se conecta a la bomba está hecha por medio de una manguera, para que la misma sea fácil de remover al momento en que se le deba dar mantenimiento.

Figura 84. **Plano de tubería de distribución en mm**



Fuente: elaboración propia, empleando Solidworks Student Edition.

Figura 85. **Tubería instalada con sensor de caudal**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 86. **Tubería y bomba instalada**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

6.2. Módulos de sensores

Con base en lo propuesto en el capítulo 3, los módulos fueron diseñados en el programa KiCad y el empaquetado de los mismos en SolidWorks. La tarjeta de circuitos es de fabricación casera y los contenedores fueron impresos en una impresora 3D modelo Prusa i3, los módulos están diseñados para una fácil instalación.

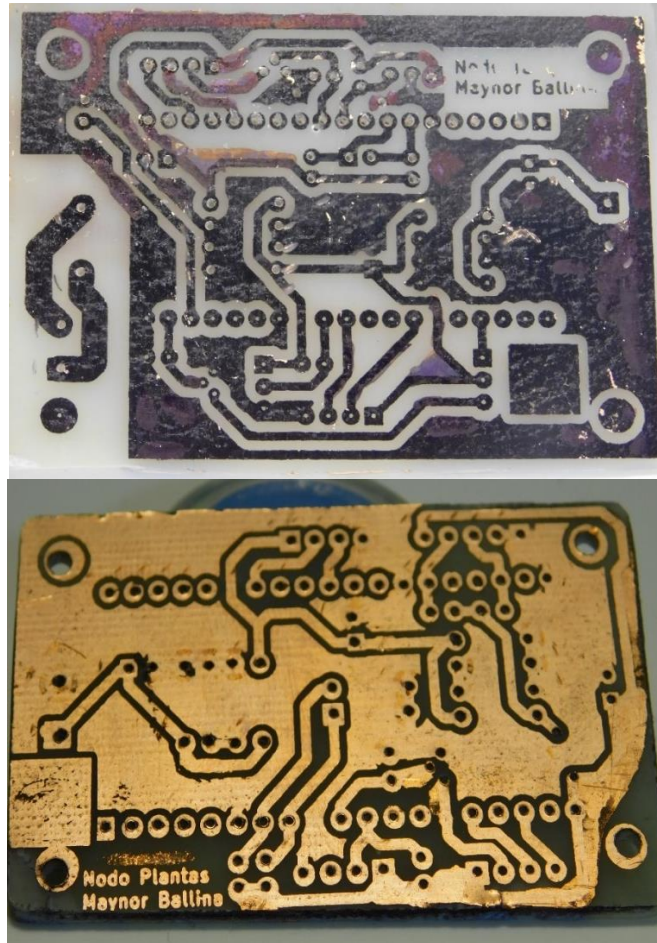
Para el tanque las puntas de medición de pH, temperatura y conductividad se instalaron dentro del tubo de 4" para que los peces no tengan contacto con ellas.

Figura 87. **Contenedores impresos**



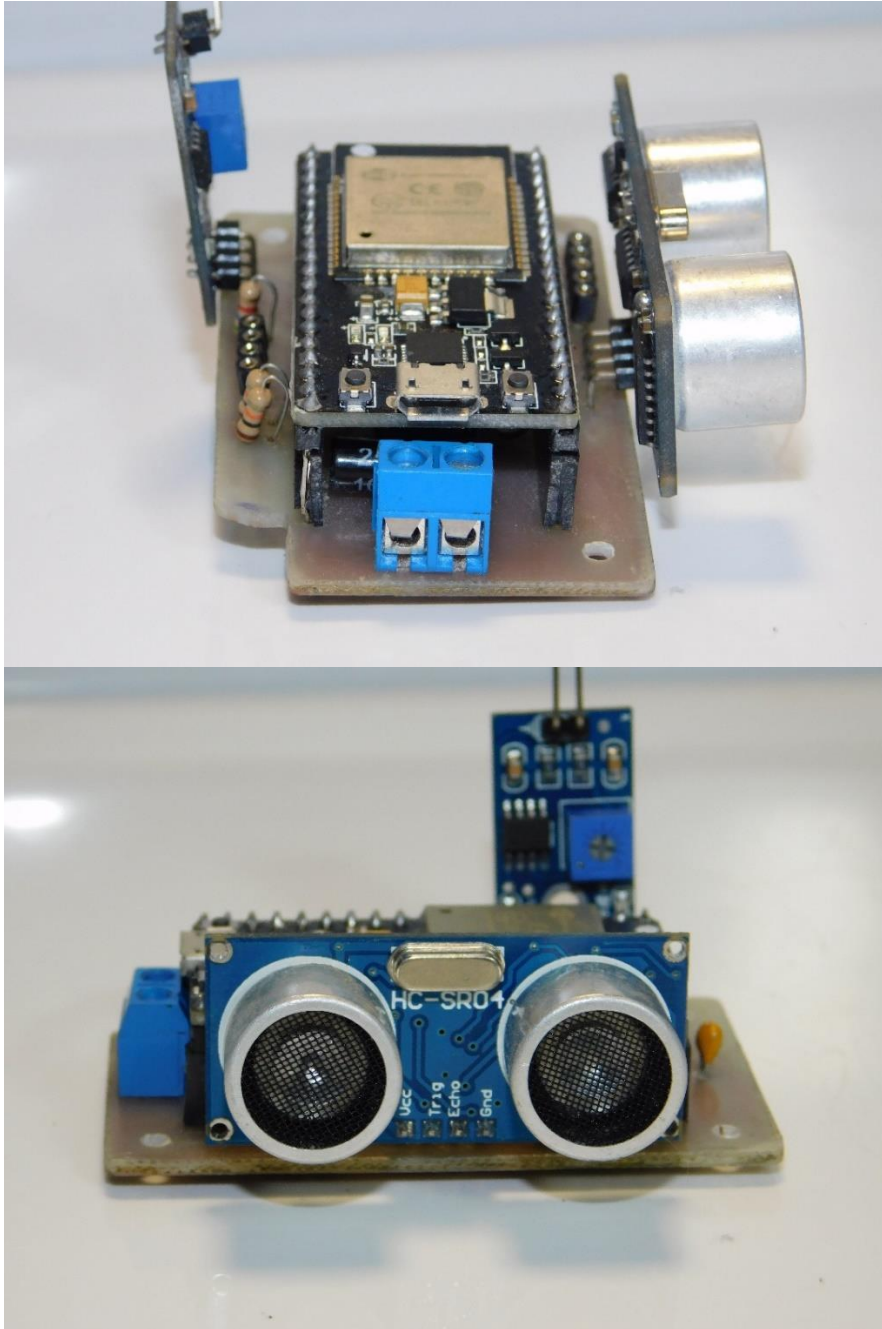
Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 88. **Placa de circuito impreso de tanque y plantas**



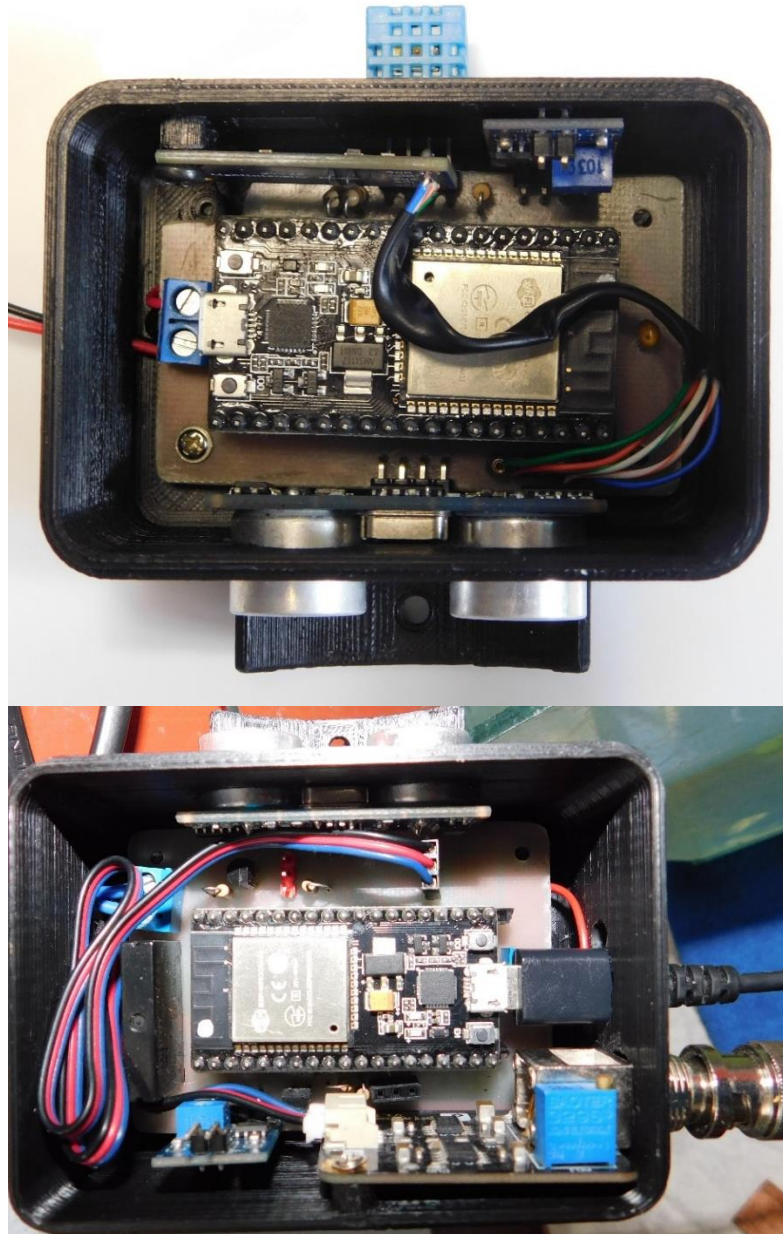
Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 89. Placas con componentes



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 90. **Instalación de las placas de los módulos**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 91. **Módulo plantas terminado e instalado**



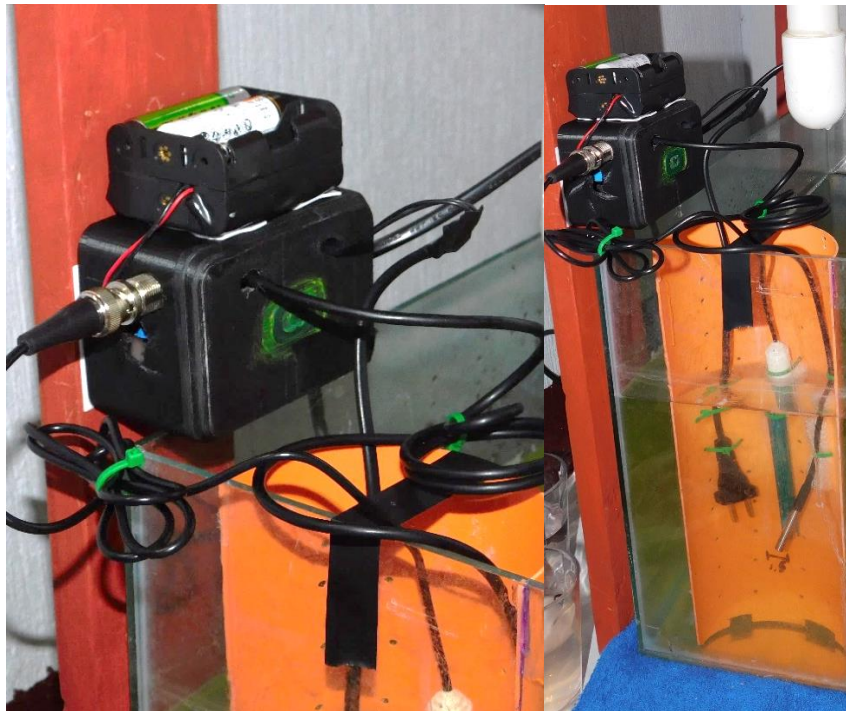
Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 92. **Tubo con puntas de prueba**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 93. **Módulo tanque terminado e instalado**



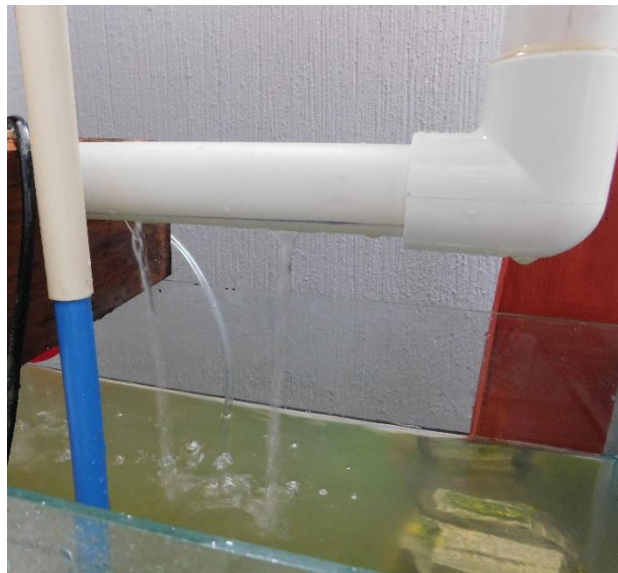
Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 94. **Salida del agua por medio de los tubos**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 95. **Caída del agua por medio del drenaje**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 96. **Sistema acuapónico completo**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

6.3. Notificaciones y monitoreo

Para facilitar el monitoreo del sistema acuapónico, por medio de la plataforma Node Red, la cual se encuentra en la Raspberry Pi Zero W, se diseña una página web que es accesible a nivel local y que muestra los valores y estados del sistema en tiempo real.

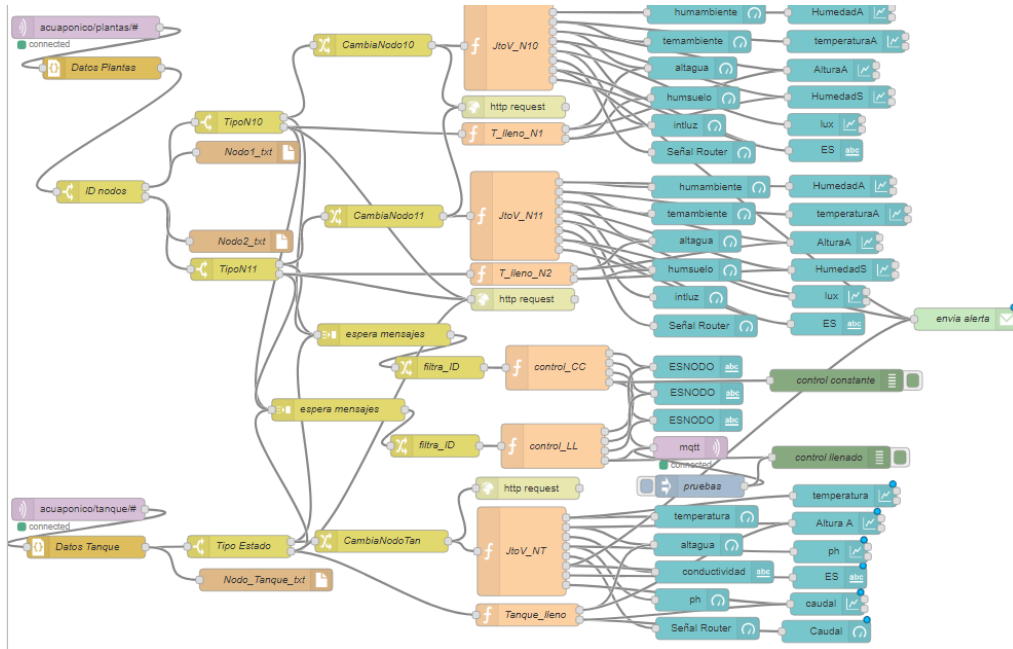
Por medio de Google Drive se puede ingresar a la base de datos realizada en hojas de cálculo de Google para revisar el comportamiento que ha tenido el sistema y, por último, por medio de correo electrónico desde la cuenta mbacuaponic@gmail.com, el sistema envía a los usuarios especificados las alertas para las cuales el sistema necesita intervención del usuario.

Figura 97. **Raspberry Pi Zero W, instalada**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 98. Flujo de Node Red



Fuente: elaboración propia, empleando Node Red.

Figura 99. Página web de monitoreo



Fuente: elaboración propia, empleando Node Red.

Figura 100. Base de datos actualizable en tiempo real

1	Marca temporal	Id	Estado	Humedad Ambiente	Temperatura Ambiente	Capacidad	Humedad Suelo	Lux
2	23/8/2018 8:41:45	NPlantas10	control	48	29	0	0	0
3	23/8/2018 8:50:05	NPlantas11	control	36	23	0	0	0
4	23/8/2018 9:01:59	NPlantas10	control	0	0	82	96	0
5	23/8/2018 9:02:14	NPlantas11	control	32	24	106	29	0
6	23/8/2018 9:32:00	NPlantas10	control	37	25	99	82	1002
7	23/8/2018 9:32:16	NPlantas11	control	30	24	43	32	1110
8	23/8/2018 10:02:02	NPlantas10	control	37	26	38	78	885
9	23/8/2018 10:02:20	NPlantas11	control	31	24	92	31	1013
10	23/8/2018 10:32:04	NPlantas10	control	0	0	50	45	474
11	23/8/2018 10:32:23	NPlantas11	control	30	24	94	34	430
12	23/8/2018 12:06:22	NPlantas10	control	0	0	70	40	550
13	23/8/2018 12:06:36	NPlantas11	control	29	27	94	16	570
14	23/8/2018 12:36:20	NPlantas10	control	36	28	74	36	1249
15	23/8/2018 12:36:38	NPlantas11	control	25	27	81	45	1564
16	23/8/2018 13:06:43	NPlantas11	control	25	27	81	40	2454
17	23/8/2018 13:36:25	NPlantas10	control	35	29	51	32	1223
18	23/8/2018 13:36:46	NPlantas11	control	25	28	86	39	1398
19	23/8/2018 14:06:50	NPlantas11	control	25	28	95	39	908
20	23/8/2018 14:36:28	NPlantas10	control	35	29	23	30	833
21	23/8/2018 14:36:54	NPlantas11	control	29	28	91	38	1067
22	23/8/2018 14:06:58	NPlantas11	control	25	28	91	39	1303

Fuente: elaboración propia, empleando Google Spreadsheet.

Figura 101. Correo enviado por el sistema

ALERTA Nodo Plantas 11! Papelera x

mbacuaponic@gmail.com mar., 18 sept. 10:24 (hace 1 día)

para yo

inglés > español Traducir mensaje Desactivar para: inglés x

HC-SR04
Estos sensores no estan funcionando

mbacuaponic@gmail.com mar., 18 sept. 11:48 (hace 1 día)

para yo

inglés > español Traducir mensaje Desactivar para: inglés x

DHT11
HC-SR04
YL38
BH1750

Estos sensores no estan funcionando

Fuente: elaboración propia, empleando Gmail.

6.4. Calibración y pruebas

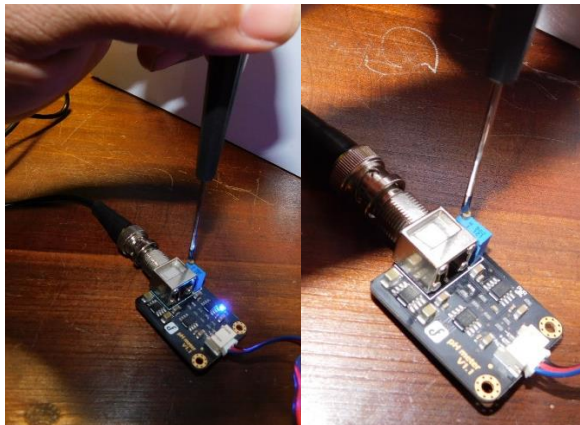
El sensor de pH requiere una calibración previa para que los valores medidos por este sean confiables. Para realizar esta calibración se utilizaron dos soluciones que ya tienen un pH definido:

- pH 4 identificada con la letra P.
- pH 7 identificada con la letra D.
- Agua desmineralizada identificada con la letra L, la funcionalidad de esta es limpiar la punta antes de cada medición,

Al realizar varias mediciones de las soluciones anteriores se comparó estas con los parámetros dados por las láminas de prueba EasyStrips, descritos en la gráfica que se encuentra en la sección 2.6 (véase la figura 32).

La calibración del dispositivo se realiza por medio del potenciómetro de calibración de ganancia, perteneciente al módulo de pH.

Figura 102. **Calibración por medio del módulo pH**



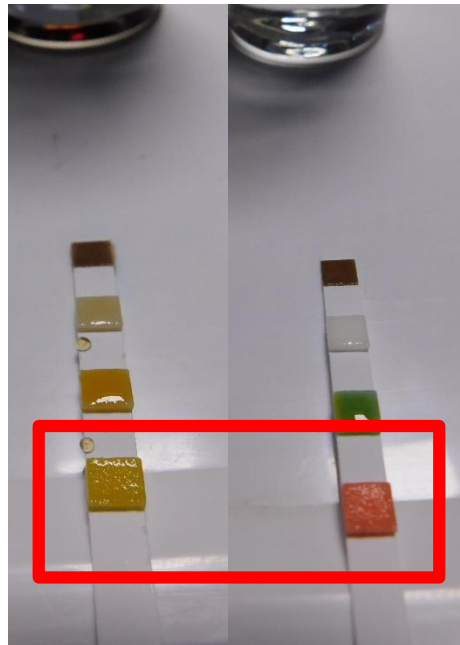
Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 103. **Soluciones de pH para calibración**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

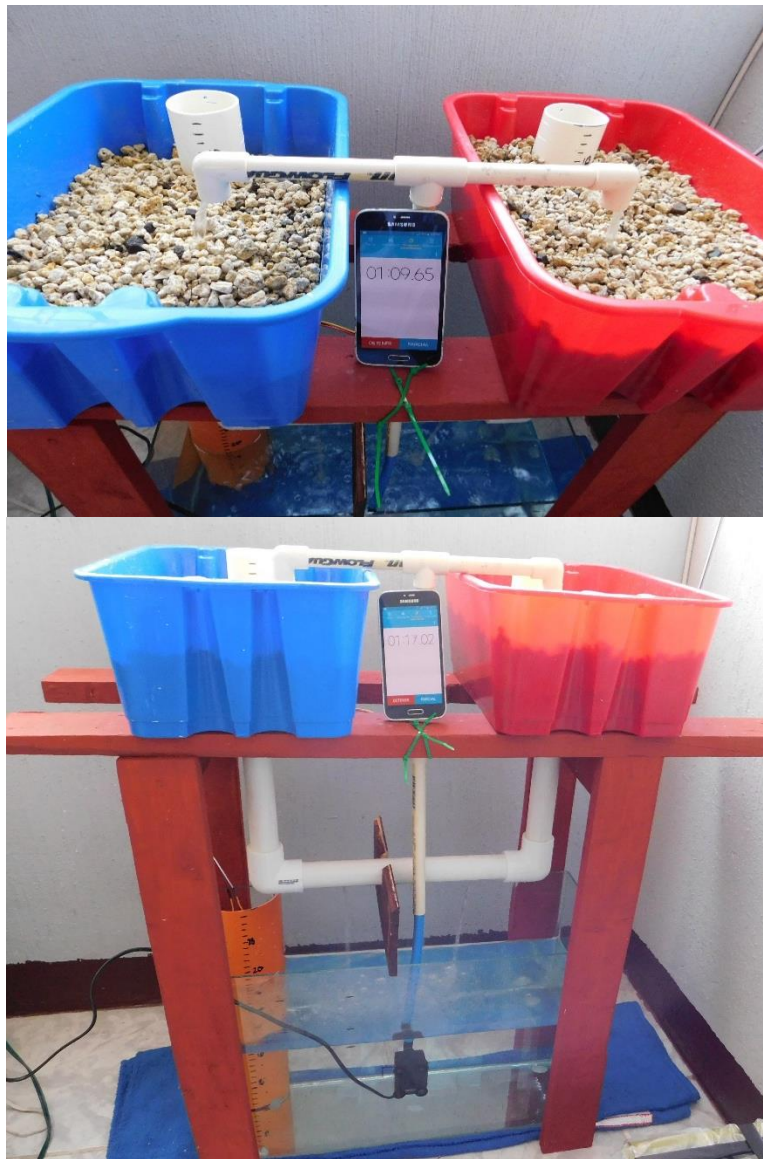
Figura 104. **Resultado de las láminas de prueba pH 4 y pH 7, respectivamente**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

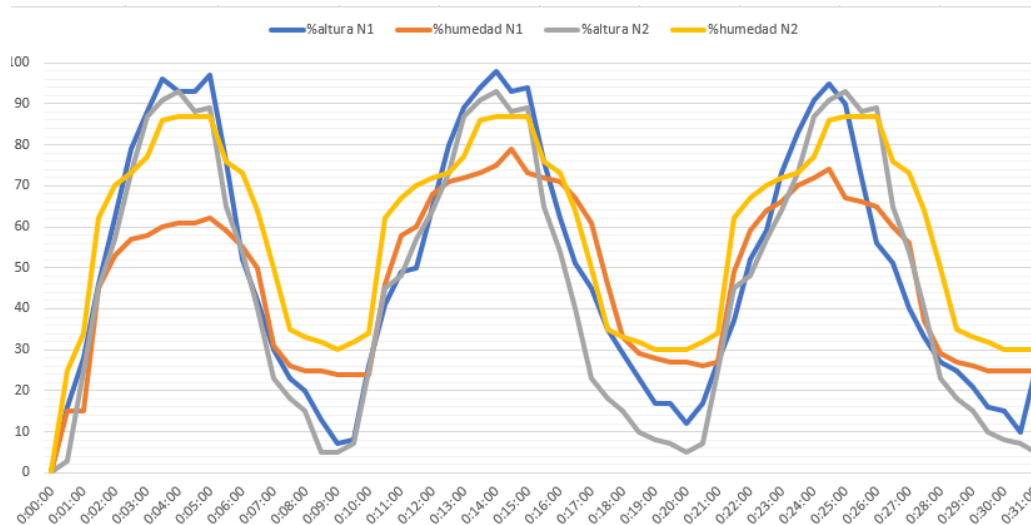
Se realiza 3 corridas de llenado y vaciado de los contenedores para graficar la reacción de los nodos con base en la altura del agua y la humedad de cada nodo.

Figura 105. **Medición de tiempos en el sistema**



Fuente: elaboración propia, empleando cámara Nikon Coolpix B500.

Figura 106. **Gráfica de reacción de nodos, llenado y vaciado**

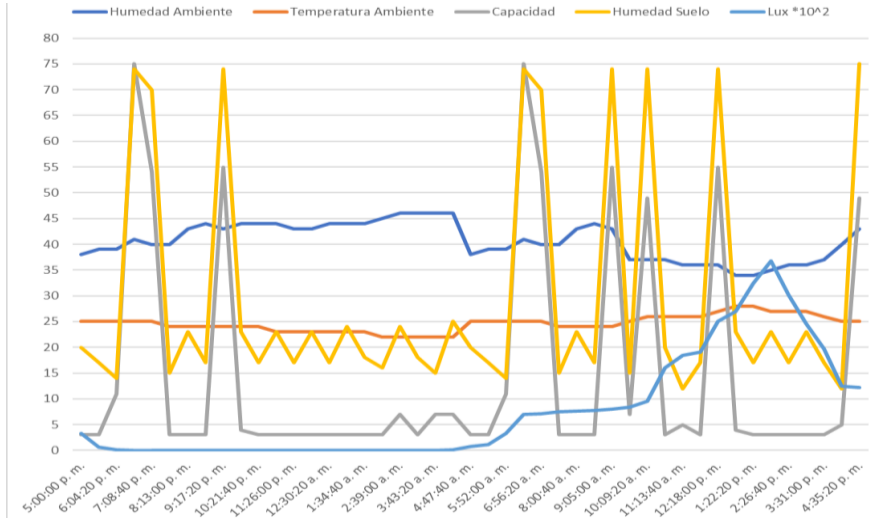


Fuente: elaboración propia.

Se grafica el comportamiento del sistema acuapónico con la información recopilada por cada nodo en la base de datos de Google Drive. Las gráficas están divididas por nodo, conectados por un período de 24 horas y sus parámetros:

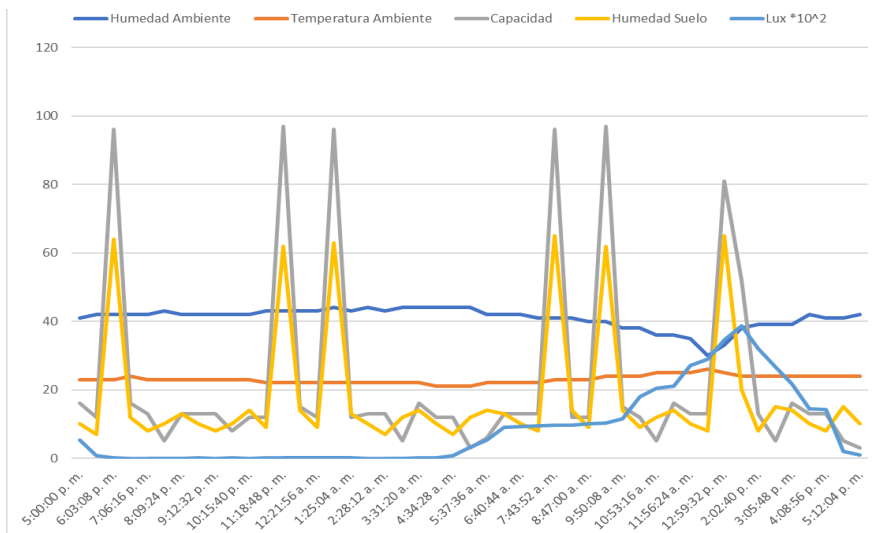
- Nodo plantas
 - Temperatura ambiente
 - Humedad ambiente
 - Luminosidad
 - Agua dentro del contenedor
 - Humedad de grava
- Nodo tanque
 - Temperatura del agua
 - pH
 - Conductividad
 - Altura del agua

Figura 107. **Nodo plantas 10**



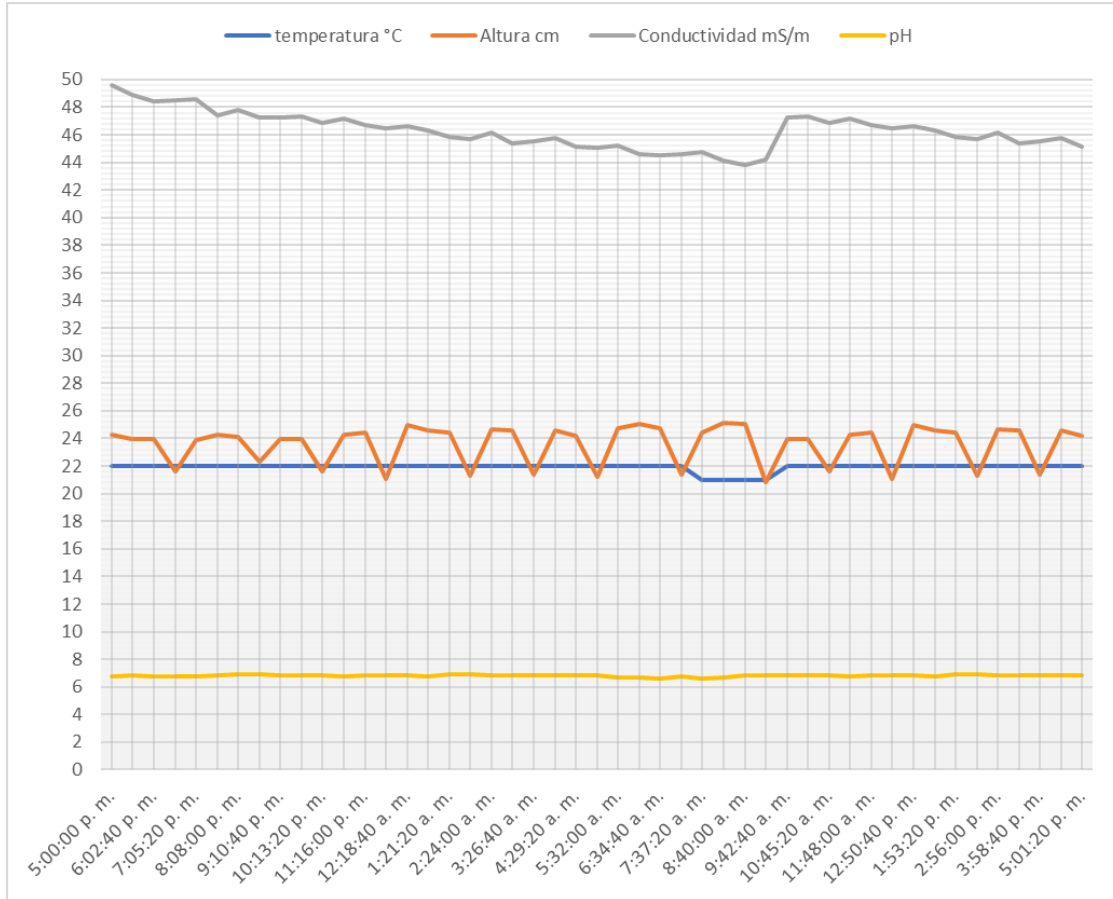
Fuente: elaboración propia.

Figura 108. **Nodo plantas 11**



Fuente: elaboración propia.

Figura 109. **Nodo tanque 11**



Fuente: elaboración propia.

CONCLUSIONES

1. La red de sensores inalámbrica que utiliza un sistema acuapónico depende de un nodo central, el cual controla los nodos que contengan sensores y actuadores.
2. El microcontrolador ESP32S contiene los elementos necesarios para implementar un módulo IoT, optimizando el espacio y el consumo de energía.
3. El conjunto de sensores utilizados cumple las condiciones necesarias para implementar en un sistema acuapónico, además de ser de bajo costo.
4. El protocolo MQTT facilita la comunicación entre los nodos sin necesidad de saturar el ancho de banda de la red Wifi, siendo la comunicación entre cada nodo asíncrona, el tiempo de desconexión de cada nodo se puede controlar por medio del *broker*, además de asegurar que cada dato sea entregado a su destino.
5. El sensor de pH debe estar calibrado antes de cada primer uso, con base en la hoja de especificaciones del fabricante, este tendrá que calibrarse de forma periódica para garantizar confiabilidad en las mediciones obtenidas.
6. Dado que el sistema es modular, la alimentación por medio de baterías es necesaria para su funcionamiento, debido a que el voltaje mínimo de algunos sensores es de 4,5 V, la alimentación mínima es de 5 V, el equivalente a 4 baterías recargables de 1,2 V en serie.

RECOMENDACIONES

1. La implementación de la red de sensores no necesita de un *router* específico, esta se puede implementar en cualquier dispositivo capaz de crear una red local inalámbrica que cumpla con el estándar IEEE 802.11 g/n.
2. La configuración del *broker* MQTT utilizando mosquitto se puede implementar en cualquier dispositivo que contenga el sistema operativo Windows, MacOS, y diversas distribuciones que contengan el *kernel* Linux y BSD.
3. En el caso de utilizar el sistema para mayor cantidad de nodos de plantas y tanques, el máximo número de nodos es de 8, 4 tanques y 4 camas de plantas.
4. Dependiendo del *router* que se utilice en el sistema, este debe estar distribuido de manera que la distancia entre los nodos y el *router* sea la menor posible, para evitar pérdidas de comunicación entre este y los nodos.
5. La distancia mínima de separación entre cada nodo debe ser una longitud de onda a frecuencia de 2,4 GHz (12,5 cm).

6. Verificar que, al instalar cada módulo, la distancia de separación entre el módulo y la altura máxima del agua sea mayor a 2 cm.
7. El sistema utiliza el puerto 1883 debido a que este está dentro de una red local segura, de lo contrario se aconseja cambiar al puerto 8883 para que la comunicación sea encriptada.
8. Habilitar TLS en el cliente y servidor si se desea que la comunicación MQTT sea encriptada.

BIBLIOGRAFÍA

1. AGARWAL, Tarun. *Wireless sensor networks and their applications*. [en línea]. <<https://www.elprocus.com/introduction-to-wireless-sensor-networks-types-and-applications/>>. [Consulta: 29 de mayo de 2018].
2. BARBARO, Lorena; et al. *Sustrato para techos verdes sustentables*. [en línea]. <https://inta.gob.ar/sites/default/files/sustrato_para_techos_verde_sustentables_extensivos.pdf>. [Consulta: 16 de septiembre de 2018].
3. BOLTON, William. *Mecatrónica. Sistemas de control electrónico en la ingeniería mecánica y eléctrica. Un enfoque multidisciplinario*. 5a ed. Distrito Federal, México: Alfaomega Grupo Editor S. A., 2013. 648 p.
4. CANDARLE, Pablo. *Técnicas de Acuaponía*. Argentina: Centro Nacional de Desarrollo Acuícola (CENADAC), 2014. 47 p.
5. CANTÚ TREVIÑO, Thelma Guadalupe; et al. *Teoría de autómatas. Un enfoque práctico*. 1a ed. México: Pearson Educación, 2015. 168 p.
6. CHEN LOPEZ, Jose. *La influencia de la luz en el crecimiento del cultivo*. [en línea]. <<https://www.pthorticulture.com/es/centro-de-formacion/la-influencia-de-la-luz-en-el-crecimiento-del-cultivo/>>. [Consulta: 4 de junio de 2018].

7. CREUS, Antonio. *Instrumentación industrial*. 8a ed. México: Alfaomega Grupo Editor S.A., 2010. 792 p.
8. Cytr0n Technologies. *Product User's Manual HC-SR04 Ultrasonic Sensor*. [en línea]. <<https://www.electroschematics.com/8902/hc-sr04-datasheet/>>. [Consulta: 4 de junio de 2018].
9. Dallas Semiconductor. *DS18B20 Programmable Resolution 1-Wire Digital Thermometer*. [en línea]. <<https://cdn.sparkfun.com/datasheets/Sensors/Temp/DS18B20.pdf>>. [Consulta: 5 de junio de 2018].
10. Espressif Systems. *ESP32 Datasheet, Version 2.2*. [en línea]. <https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf>. [Consulta: junio 2018].
11. Espressif Systems. *Esp32 Overview*. [en línea]. <<https://www.espressif.com/en/products/hardware/esp32/overview>>. [Consulta: 9 de junio de 2018].
12. Exposed Acuaponics. *Acuaponics Exposed*. [en línea]. <<https://aquaponicsexposed.com/ph-levels-in-aquaponics/>>. [Consulta: 16 de septiembre de 2018].
13. FAO Organización de las Naciones Unidas para la Alimentación y la Agricultura. *Siete reglas básicas que hay que seguir en la acuaponía*. [en línea]. <<http://www.fao.org/zhc/detail-events/es/c/325888/>>. [Consulta: 5 de junio de 2018].

14. HERNANDEZ, Irene. *Sensor de humedad del suelo: módulos YL-69 y YL-38*. [en línea]. <<https://myrobotself.com/2016/07/23/sensor-de-humedad-del-suelo-modulos-yl-69-y-yl-38/>>. [Consulta: 4 de junio de 2018].
15. MASSER, Michael; et al. *Recirculating aquaculture tank production systems, management of recirculating systems*. Southern Regional Aquaculture Center(452). [en línea]. <<http://aqua.ucdavis.edu/DatabaseRoot/pdf/452RFS.PDF>>. [Consulta: septiembre de 2017].
16. MikroElectronica. *UART - Serial communication*. [en línea]. <<https://www.mikroe.com/blog/uart-serial-communication>>. [Consulta: 29 de mayo de 2018].
17. Mouser Electronics. *DHT11 humidity and temperature sensor*. [en línea]. <<https://www.mouser.com/ds/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>>. [Consulta: 1 de junio de 2018].
18. MQTT Community. *MQTT.ORG*. [en línea]. <<http://www.mqtt.org>>. [Consulta: 26 de junio de 2018].
19. National Instruments. *¿Qué es una Red de Sensores Inalámbricos?* [en línea]. <<http://www.ni.com/white-paper/7142/es/>>. [Consulta: 3 de febrero de 2018].
20. OASIS. *MQTT Version 3.1.1*. [en línea]. <<http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>>. [Consulta: 30 de mayo de 2018].

21. PEERY, JoAnn. *¿Cómo influye la humedad en la calidad de los cultivos?* [en línea]. <<https://www.pthorticulture.com/es/centro-de-formacion/como-influye-la-humedad-en-la-calidad-de-los-cultivos>>. [Consulta: 1 de junio de 2018].
22. PÉREZ, Marco Antonio; et al.. *Fundamentos de robótica y mecatrónica con Matlab y Simulink*. 1a ed. México: Alfaomega Grupo Editor S.A., 2014. 682 p.
23. Rohm Semiconductors. *Digital 16bit Serial Output Type Ambient Light Sensor IC BH1750FVI*. [en línea]. <<http://www.mouser.com/ds/2/348/bh1750fvi-e-186247.pdf>>. [Consulta: 11 de junio de 2018].
24. ROUSE, Margaret. *MQTT (MQ Telemetry Transport)*. [en línea]. <<https://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport>>. [Consulta: 26 de junio de 2018].
25. SOMERVILLE, Christopher; et al. *Small-scale aquaponic food production – Integrated fish and plant farming*. Food and Agriculture Organization of the United Nations, Fisheries and Aquaculture. Roma: FAO. [en línea]. <<https://fao.org/publications>>. [Consulta: 7 de julio de 2018].
26. Sparkfun. *CdS Photoconductive cells GL5528*. [en línea]. <<https://cdn.sparkfun.com/datasheets/Sensors/LightImaging/SEN-09088.pdf>>. [Consulta: 4 de junio de 2018].
27. Water Boards California State Water Resources Control Board. *Folleto informativo amoniaco*. [en línea]. <<https://www.waterboards.ca.gov/>>

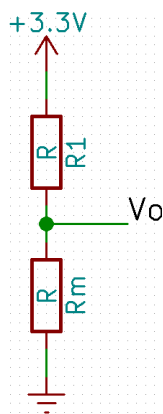
water_issues/programs/swamp/docs/cwt/guidance/3310sp.pdf>.
[Consulta: 17 de septiembre de 2018].

28. _____. *Folleto Informativo, Oxígeno Disuelto (OD)*. [en línea].
<https://www.waterboards.ca.gov/water_issues/programs/swamp/docs/cwt/guidance/3110sp.pdf>. [Consulta: 5 de junio de 2018].

APÉNDICES

Apéndice 1. Cálculo de conductividad

Con base en un simple divisor de voltaje donde R_m es la resistencia obtenida por medio de la punta de medición.



Con base en la segunda ley de Kirchoff se puede obtener el divisor de voltaje para V_o :

$$V_o = \frac{R_m}{R_1 + R_m} * V_{in}$$

Según Ohm la resistencia se define por medio de la resistividad del material, la longitud y el área del mismo, la resistividad es igual al inverso de la conductividad, despejando para R_m y substituyendo los valores se obtiene:

$$R_2 = \frac{V_o * R_1}{V_{in} - V_o} \rightarrow \frac{l}{\sigma * A} = \frac{V_o * R_1}{V_{in} - V_o} \rightarrow \sigma = \frac{l * (V_{in} - V_o)}{A * (V_o * R_1)}$$

Teniendo los valores para cada elemento de la siguiente forma:

- $R_1 = 5.1K\Omega$
- $V_{in} = 3.3V$
- $A = 0.006 \text{ m} * 0.016 \text{ m} = 0.000096 \text{ m}^2$
- $l = 0.01 \text{ m}$
- V_o valor medido por el microcontrolador.

Obteniendo la ecuación para la conductividad:

$$\sigma = \left(\frac{3.3 - V_o}{5100 * V_o} \right) * 104.167$$

Fuente: elaboración propia.

Apéndice 2. Código en json para el nodo central, *broker*

```
[{"id":"8d8938ab.619df8","type":"mqttin","z":"47ecbde4c4a764","name":"","topic":"acuaponico/plantas/#","qos":"1","broker":"d1a35940.03ea38","x":105,"y":40,"wires":[["a5164221.d4926"]]}, {"id":"a5164221.d4926","type":"json","z":"47ecbde4c4a764","name":"Datos Plantas","property":"payload","action":"obj","pretty":false,"x":127,"y":95,"wires":[["f752b385.29348"]]}, {"id":"525380d6.511e4","type":"ui_gauge","z":"47ecbde4c4a764","name":"temambiente","group":"c5d665b8.e6c208","order":3,"width":3,"height":3,"gtype":"gauge","title":"Temperatura","label":"°C","format":"{{value}}","min":0,"max":60,"colors":["#ff0000","#00ff00","#ff0000"],"seg1":16,"seg2":30,"x":934.5000190734863,"y":59.25000190734863,"wires":[]}, {"id":"38654e3f.1a4ff2","type":"ui_gauge","z":"47ecbde4c4a764","name":"altagua","group":"c5d665b8.e6c208","order":1,"width":4,"height":3,"gtype":"wave","title":"Altura Agua","label":"%","format":"{{value}}","min":0,"max":100,"colors":["#00b500","#e6e600","#ca3838"],"seg1":25,"seg2":"","x":925.5000267028809,"y":95.00000190734863,"wires":[]}, {"id":"9d6c13e2.2623e","type":"ui_gauge","z":"47ecbde4c4a764","name":"humusuelo","group":"c5d665b8.e6c208","order":4,"width":3,"height":3,"gtype":"gauge","title":"Humedad Suelo","label":"% HR","format":"{{value}}","min":0,"max":100,"colors":["#ff0000","#e6e600","#00ff00"],"seg1":20,"seg2":30,"x":929.2500267028809,"y":135.25000190734863,"wires":[]}, {"id":"9b521032.0d99d","type":"ui_gauge","z":"47ecbde4c4a764","name":"intluz","group":"c5d665b8.e6c208","order":5,"width":3,"height":3,"gtype":"gauge","title":"Intensidad Luminosa","label":"lux","format":"{{value}}","min":0,"max":5000,"colors":["#ff0000","#e6e600","#00ff00"],"seg1":400,"seg2":2000,"x":920.2500228881836,"y":172.25000286102295,"wires":[]}, {"id":"e372e94d.d0b328","type":"ui_chart","z":"47ecbde4c4a764","name":"temperaturaA","group":"82d6bf8e.19f7f","order":6,"width":6,"height":5,"label":"temperatura ambiente","chartType":"line","legend":false,"xformat":"HH:mm","interpolate":"step","nodata":"","dot":true,"ymin":0,"ymax":50,"removeOlder":1,"removeOlderPoints":"","removeOlderUnit":86400,"cutout":0,"useOneColor":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"x":1133.0000190734863,"y":59.00000190734863,"wires":[[],[]]}, {"id":"f752b385.29348","type":"switch","z":"47ecbde4c4a764","name":"ID nodos","property":"payload.ID","propertyType":"msg","rules":[{"t":"eq","v":"NPlantas10","vt":"str"}, {"t":"eq","v":"NPlantas11","vt":"str"}],"checkall":true,"repair":false,"outputs":2,"x":125.75003051757812,"y":262.00000381469727,"wires":[["7971a00c.f8669","cef2ba0f.948d18"],["9008eddb.4b981","d42f62a1.189bd"]]}, {"id":"6ceae97f.200248","type":"ui_text","z":"47ecbde4c4a764","group":"c5d665b8.e6c208","order":7,"width":4,"height":3,"name":"ES","label":"Estatus Sensores","format":"{{msg.payload}}","layout":"col-center","x":1105.5000038146973,"y":206.25,"wires":[]}, {"id":"7971a00c.f8669","type":"switch","z":"47ecbde4c4a764","name":"TipoN10","property":"payload.tipo","propertyType":"msg"}]
```

Continuación apéndice 2.

```
, "rules": [{"t": "eq", "v": "control", "vt": "str"}, {"t": "eq", "v": "llenando", "vt": "str"}], "checkall": "true", "repair": false, "outputs": 2, "x": 312.75000762939453, "y": 168.00000190734863, "wires": [{"id": "912fbc17.6809e", "type": "switch", "z": "47ecbde4a764", "name": "TipoN11", "property": "payload.tipo", "propertyType": "msg", "rules": [{"t": "eq", "v": "control", "vt": "str"}, {"t": "eq", "v": "llenando", "vt": "str"}], "checkall": "true", "repair": false, "outputs": 2, "x": 306.74997329711914, "y": 366.5000057220459, "wires": [{"id": "83c8115b.cfb0f", "type": "function", "z": "47ecbde4a764", "name": "T_lleno_N1", "func": "var nivel = (\\"payload\\":msg.payload.nivel);\nvar humed = (\\"payload\\":msg.payload.humed);\nflow.set(\\"ALL1\\", nivel);\nreturn [nivel,humed];", "outputs": 2, "noerr": 0, "x": 683.749927520752, "y": 183.75001525878906, "wires": [{"id": "38654e3f.1a4ff2", "type": "mqtt out", "z": "47ecbde4a764", "name": "", "topic": "", "qos": 2, "retain": "", "broker": "d1a35940.03ea38", "x": 922.7501354217529, "y": 607.5000696182251, "wires": [{"id": "94416246.8b62c", "type": "mqtt in", "z": "47ecbde4a764", "name": "", "topic": "acuapónico/tanque/#", "qos": 1, "broker": "d1a35940.03ea38", "x": 95, "y": 690.750129699707, "wires": [{"id": "e1cdc1cf.09c8a", "type": "json", "z": "47ecbde4a764", "name": "Datos Tanque", "property": "payload", "action": "obj", "pretty": false, "x": 105.75, "y": 741.7501306533813, "wires": [{"id": "8e38c31d.0f706", "type": "switch", "z": "47ecbde4a764", "name": "Tipo Estado", "property": "payload.tipo", "propertyType": "msg", "rules": [{"t": "eq", "v": "control", "vt": "str"}, {"t": "eq", "v": "llenando", "vt": "str"}], "checkall": "true", "repair": false, "outputs": 2, "x": 312.49998664855957, "y": 741.0001430511475, "wires": [{"id": "83c8115b.cfb0f", "type": "ui_text", "z": "47ecbde4a764", "group": "62c99496.b8760c", "order": 7, "width": 4, "height": 3, "name": "ES", "label": "Estatus Sensores:", "format": "{{msg.payload}}", "layout": "col-center", "x": 111.0000343322754, "y": 782.6429500579834, "wires": [{"id": "ad2cc8a6.8dd7f8", "type": "ui_gauge", "z": "47ecbde4a764", "name": "temperatura", "group": "62c99496.b8760c", "order": 2, "width": 3, "height": 3, "gtype": "gage", "title": "temperatura", "label": "°C", "format": "{{value}}", "min": 0, "max": 60, "colors": ["#ff0000", "#00ff00", "#ff0000"], "seg1": "16", "seg2": "32", "x": 944.2499961853027, "y": 700.5001239776611, "wires": [{"id": "98ed5e9a.041b8", "type": "ui_gauge", "z": "47ecbde4a764", "name": "altagua", "group": "62c99496.b8760c", "order": 1, "width": 4, "height": 3, "gtype": "wave", "title": "Altura Agua", "label": "cm", "format": "{{value}}", "min": 0, "max": 27, "colors": ["#00b500", "#e6e600", "#ca3838"], "seg1": "25", "seg2": "", "x": 933.2499961853027, "y": 738.5001239776611, "wires": [{"id": "2d005215.2c407e", "type": "ui_gauge", "z": "47ecbde4a764", "name": "ph", "group": "62c99496.b8760c", "orde
```

Continuación apéndice 2.

```
r":3,"width":3,"height":3,"gtype":"gage","title":"pH","label":"","format":"{{value}}","min":0,"max":15,"colors":["#ff0000","#00ff00","#ff0000"],"seg1":"6","seg2":"8","x":924.7499961853027,"y":813.0001449584961,"wires":[{}],{"id":"567befd.7bfea1","type":"httprequest","z":"47ecbde.c4a764","name":"","method":"GET","ret":"txt","url":"https://docs.google.com/forms/d/e/1FAIpQLSfVltEiv_SGSujlpMPu-8-52JamSaIJ98311HvanNy-3-64Gw/formResponse?entry.1490523329={{payload.ID}}&entry.2135298951={{payload.tipo}}&entry.1671924424={{payload.temperatura}}&entry.1034451498={{payload.altura}}&entry.1169345962={{payload.conductividad}}&entry.1332604270={{payload.ph}}&entry.1652115617=0","tls":"","x":702.4999923706055,"y":662.2501430511475,"wires":[{}],{"id":"309b23f5.7e188c","type":"file","z":"47ecbde.c4a764","name":"Nodo_Tanque_txt","filename":"/home/pi/Documents/Acuaponia/Tanque.txt","appendNewline":true,"createDir":false,"overwriteFile":"false","x":348.82141494750977,"y":786.000129699707,"wires":[{}],{"id":"57446118.cb08","type":"ui_gauge","z":"47ecbde.c4a764","name":"Caudal","group":"62c99496.b8760c","order":4,"width":3,"height":3,"gtype":"gage","title":"Caudal","label":"L/m","format":"{{value}}","min":0,"max":5.5,"colors":["#ff0000","#e6e600","#00ff00"],"seg1":"2","seg2":"3","x":1106.2499923706055,"y":855.6430130004883,"wires":[{}],{"id":"5c6e3ffd.6545","type":"ui_chart","z":"47ecbde.c4a764","name":"temperatura","group":"4476d787.235918","order":2,"width":8,"height":5,"label":"temperatura","chartType":"line","legend":"false","xformat":"HH:mm","interpolate":"step","nodata":"","dot":true,"ymin":0,"ymax":50,"removeOlder":5,"removeOlderPoints":"","removeOlderUnit":60,"cutout":0,"useOneColor":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"x":1131.25,"y":672.5001430511475,"wires":[[{}],{}],{"id":"a773215e.4e0e7","type":"ui_chart","z":"47ecbde.c4a764","name":"HumedadS","group":"82d6bf8e.19f7f","order":8,"width":6,"height":5,"label":"Humedad Suelo","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpolate":"linear","nodata":"","dot":false,"ymin":"","ymax":"","removeOlder":1,"removeOlderPoints":"","removeOlderUnit":86400,"cutout":0,"useOneColor":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"x":1124.2500190734863,"y":134.00000286102295,"wires":[[{}],{}],{"id":"52dc1346.1fa68c","type":"ui_chart","z":"47ecbde.c4a764","name":"lux","group":"82d6bf8e.19f7f","order":9,"width":6,"height":5,"label":"Luminosidad","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpolate":"linear","nodata":"","dot":false,"ymin":"","ymax":"","removeOlder":1,"removeOlderPoints":"","removeOlderUnit":86400,"cutout":0,"useOneColor":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"x":1104.0000171661377,"y":171.25000286102295,"wires":[[{}],{}],{"id":"afaaf214.1d372","type":"http request","z":"47ecbde.c4a764","name":"","method":
```

Continuación apéndice 2.

```
"GET", "ret": "txt", "url": "https://docs.google.com/forms/d/e/1FAIpQLSf1Wue5njQkX-339W5AG85y_U7LbmiafnGbfMC4ZZJ9SQLNnw/formResponse?entry.1919654248={{payload.ID}}&entry.1336959631={{payload.tipo}}&entry.1807322180={{payload.humedadA}}&entry.16656599={{payload.temperaturaA}}&entry.1036599364={{payload.capacidad}}&entry.1480680478={{payload.humedadS}}&entry.119943915={{payload.lux}}", "tls": "", "x": 683.7498970031738, "y": 147.5000400543213, "wires": [], {"id": "cef2ba0f.948d18", "type": "file", "z": "47ecbde.c4a764", "name": "Nodo1_txt", "fileName": "/home/pi/Documents/Acuaponia/Nodo1.txt", "appendNewline": true, "createDir": false, "overwriteFile": "false", "x": 324.2500343322754, "y": 208.75008010864258, "wires": [], {"id": "6915b193.e9037", "type": "ui_gauge", "z": "47ecbde.c4a764", "name": "temambiente", "group": "97e686a9.dbf448", "order": 3, "width": 3, "height": 3, "gtype": "gage", "title": "Temperatura", "label": "°C", "format": "{{value}}", "min": 0, "max": 60, "colors": ["#ff0000", "#00ff00", "#ff0000"], "seg1": "16", "seg2": "30", "x": 940.2118110656738, "y": 291.5000801086426, "wires": [], {"id": "bfd8205e.68616", "type": "ui_gauge", "z": "47ecbde.c4a764", "name": "altagua", "group": "97e686a9.dbf448", "order": 1, "width": 4, "height": 3, "gtype": "wave", "title": "Altura Agua", "label": "%", "format": "{{value}}", "min": 0, "max": 100, "colors": ["#00b500", "#e6e600", "#ca3838"], "seg1": "25", "seg2": "", "x": 931.2118110656738, "y": 328.5000801086426, "wires": [], {"id": "294277b.4bcd088", "type": "ui_gauge", "z": "47ecbde.c4a764", "name": "humusuelo", "group": "97e686a9.dbf448", "order": 4, "width": 3, "height": 3, "gtype": "gage", "title": "Humedad Suelo", "label": "% HR", "format": "{{value}}", "min": 0, "max": 100, "colors": ["#ff0000", "#e6e600", "#00ff00"], "seg1": "25", "seg2": "50", "x": 931.2118110656738, "y": 367.5000801086426, "wires": [], {"id": "8e44078c.7d22a8", "type": "ui_gauge", "z": "47ecbde.c4a764", "name": "intluz", "group": "97e686a9.dbf448", "order": 5, "width": 3, "height": 3, "gtype": "gage", "title": "Intensidad Luminosa", "label": "lux", "format": "{{value}}", "min": 0, "max": 5000, "colors": ["#ff0000", "#e6e600", "#00ff00"], "seg1": "400", "seg2": "2000", "x": 922.2118110656738, "y": 404.5000801086426, "wires": [], {"id": "9e320252.17c1d", "type": "ui_chart", "z": "47ecbde.c4a764", "name": "temperaturaA", "group": "b5640d0a.cd87d", "order": 6, "width": 6, "height": 5, "label": "temperatura ambiente", "chartType": "line", "legend": "false", "xformat": "HH:mm", "interpolate": "linear", "nodata": "", "dot": false, "ymin": "", "ymax": "", "removeOlder": "1", "removeOlderPoints": "", "removeOlderUnit": "86400", "cutout": 0, "useOneColor": false, "colors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a", "#d62728", "#ff9896", "#9467bd", "#c5b0d5"], "useOldStyle": false, "x": 1136.2118110656738, "y": 292.5000801086426, "wires": [], {"id": "bb57e99.88be5", "type": "ui_gauge", "z": "47ecbde.c4a764", "name": "humambiente", "group": "97e686a9.dbf448", "order": 2, "width": 3, "height": 3, "gtype": "gage", "title": "Humedad Ambiente", "label": "% HR", "format": "{{value}}", "min": 0, "max": 100, "colors": ["#ff0000", "#00ff00", "#ffff00"], "seg1": "30", "seg2": "50", "x": 950.2118110656738, "y": 255.50008010864258, "wires": [], {"id": "4da41a94.
```

Continuación apéndice 2.

cbc854", "type": "ui_text", "z": "47ecbde4a764", "group": "97e686a9.dbf448", "order": 7, "width": "4", "height": "3", "name": "ES", "label": "Estatus Sensores:", "format": "{{msg.payload}}", "layout": "col-center", "x": 1108.2118034362793, "y": 440.5000705718994, "wires": [], {"id": "f2f311a6.7a7e4", "type": "ui_chart", "z": "47ecbde4a764", "name": "HumedadA", "group": "b5640d0a.cd87d", "order": 7, "width": "6", "height": "5", "label": "Humedad ambiente", "chartType": "line", "legend": "false", "xformat": "HH:mm:ss", "interpolate": "linear", "nodata": "", "dot": false, "ymin": "", "ymax": "", "removeOlder": "1", "removeOlderPoints": "", "removeOlderUnit": "86400", "cutout": 0, "useOneColor": false, "colors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a", "#d62728", "#ff9896", "#9467bd", "#c5b0d5"], "useOldStyle": false, "x": 1126.2118110656738, "y": 254.50008010864258, "wires": [], {"id": "e5273955.8501d8", "type": "ui_chart", "z": "47ecbde4a764", "name": "HumedadS", "group": "b5640d0a.cd87d", "order": 8, "width": "6", "height": "5", "label": "Humedad Suelo", "chartType": "line", "legend": "false", "xformat": "HH:mm:ss", "interpolate": "linear", "nodata": "", "dot": false, "ymin": "", "ymax": "", "removeOlder": 1, "removeOlderPoints": "", "removeOlderUnit": "86400", "cutout": 0, "useOneColor": false, "colors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a", "#d62728", "#ff9896", "#9467bd", "#c5b0d5"], "useOldStyle": false, "x": 1129.2118110656738, "y": 367.5000801086426, "wires": [], {"id": "a62b0a90.cad678", "type": "ui_chart", "z": "47ecbde4a764", "name": "lux", "group": "b5640d0a.cd87d", "order": 9, "width": "6", "height": "5", "label": "Luminosidad", "chartType": "line", "legend": "false", "xformat": "HH:mm:ss", "interpolate": "linear", "nodata": "", "dot": false, "ymin": "", "ymax": "", "removeOlder": 1, "removeOlderPoints": "", "removeOlderUnit": "86400", "cutout": 0, "useOneColor": false, "colors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a", "#d62728", "#ff9896", "#9467bd", "#c5b0d5"], "useOldStyle": false, "x": 1110.2118110656738, "y": 403.5000801086426, "wires": [], {"id": "d42f62a1.189bd", "type": "file", "z": "47ecbde4a764", "name": "Nodo2_txt", "filename": "/home/pi/Documents/Acuaponia/Nodo2.txt", "appendNewline": true, "createDir": false, "overwriteFile": "false", "x": 315.2118148803711, "y": 328.75006675720215, "wires": [], {"id": "b2d95c71.3b777", "type": "ui_chart", "z": "47ecbde4a764", "name": "AlturaA", "group": "82d6bf8e.19f7f", "order": 0, "width": "6", "height": "5", "label": "Altura Agua", "chartType": "line", "legend": "false", "xformat": "auto", "interpolate": "linear", "nodata": "", "dot": false, "ymin": "", "ymax": "", "removeOlder": "1", "removeOlderPoints": "", "removeOlderUnit": "86400", "cutout": 0, "useOneColor": false, "colors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a", "#d62728", "#ff9896", "#9467bd", "#c5b0d5"], "useOldStyle": false, "x": 1113.2500171661377, "y": 98.00000190734863, "wires": [], {"id": "8d387218.90653", "type": "ui_chart", "z": "47ecbde4a764", "name": "AlturaA", "group": "b5640d0a.cd87d", "order": 0, "width": "6", "height": "5", "label": "Altura Agua", "chartType": "line", "legend": "false", "xformat": "HH:mm:ss", "interpolate": "linear", "nodata": "", "dot": false, "ymin": "", "ymax": "", "removeOlder": 1, "removeOlderPoints": "", "removeOlderUnit": "86400", "cutout": 0, "useOneCol

Continuación apéndice 2.

```
or":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"x":1117.7500038146973,"y":330.5000762939453,"wires":[[.[]]},{id:"675493f6.7a41cc","type":"ui_chart","z":"47ecbde4c4a764","name":"AlturaA","group":"4476d787.235918","order":7,"width":"8","height":"5","label":"AlturaAgua","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpolate":"bezier","nodata":"","dot":false,"ymin":"0","ymax":"100","removeOlder":"5","removeOlderPoints":"10","removeOlderUnit":"60","cutout":0,"useOneColor":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"x":1120.2499961853027,"y":709.5001239776611,"wires":[[.[]]},{id:"d36fb83c.65b118","type":"ui_chart","z":"47ecbde4c4a764","name":"caudal","group":"4476d787.235918","order":8,"width":"8","height":"5","label":"Caudal","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpolate":"linear","nodata":"","dot":false,"ymin":"","ymax":"","removeOlder":1,"removeOlderPoints":"","removeOlderUnit":"3600","cutout":0,"useOneColor":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"x":1107.642978668213,"y":818.535924911499,"wires":[[.[]]},{id:"e398ec0f.57ff2","type":"ui_chart","z":"47ecbde4c4a764","name":"ph","group":"4476d787.235918","order":0,"width":"8","height":"5","label":"pH","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpolate":"linear","nodata":"","dot":false,"ymin":"","ymax":"","removeOlder":1,"removeOlderPoints":"","removeOlderUnit":"3600","cutout":0,"useOneColor":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"x":1112.2168312072754,"y":747.9286937713623,"wires":[[.[]]},{id:"e83690f5.8d3ef","type":"ui_gauge","z":"47ecbde4c4a764","name":"","group":"c5d665b8.e6c208","order":6,"width":"3","height":"3","gtype":"gage","title":"Señal Router","label":"dBm","format":"{{value}}","min":"-100","max":"-30","colors":["#ff0000","#e6e600","#00ff00"],"seg1":"-88","seg2":"67","x":940.2500267028809,"y":208.75000381469727,"wires":[[.[]]},{id:"a0517896.4e3178","type":"ui_gauge","z":"47ecbde4c4a764","name":"","group":"97e686a9.dbf448","order":6,"width":"3","height":"3","gtype":"gage","title":"Señal Router","label":"dBm","format":"{{value}}","min":"-100","max":"-30","colors":["#ff0000","#e6e600","#00ff00"],"seg1":"-88","seg2":"-67","x":943.0000152587891,"y":443.00001335144043,"wires":[[.[]]},{id:"912fbc17.6809e","type":"change","z":"47ecbde4c4a764","name":"CambiaNodo10","rules":[{"t":"move","p":"payload.valores[0]","pt":"msg","to":"payload.humedadA","tot":"msg"}]},{t":"move","p":"payload.valores[0]","pt":"msg","to":"payload.temperaturaA","tot":"msg"}]},{t":"move","p":"payload.valores[0]","pt":"msg","to":"payload.capacidad","tot":"msg"}]},{t":"move","p":"payload.valores[0]","pt":"msg","to":"payload.humedadS","tot":"msg"}]},{t":"move","p":"payload.valores[0]","pt":"msg","to":"payload.lux","tot":"msg"}]},{t":"move","p":"payload.valores[0]","pt":"msg","to":"payload.r
```

Continuación apéndice 2.

```
ssi","tot":"msg"}},{t:"delete","p":"payload.valores","pt":"msg"},"action":"","property":"","from":"","to":"","reg":false,"x":493.000057220459,"y":64.00000667572021,"wires":[["40aca270.94d43c","afaaf214.1d372"]]},{"id":"40aca270.94d43c","type":"function","z":"47ecbde4c4a764","name":"JtoV_N10","func":"flow.set(\"SAAN1\",msg.payload.estados[1]);\nvar HA =({\"payload\":msg.payload.humedadA});\nvarTA =({\"payload\":msg.payload.temperaturaA});\nvar AA =({\"payload\":msg.payload.capacidad});\nflow.set(\"AAN1\",AA);\nvar HS =({\"payload\":msg.payload.humedadS});\nflow.set(\"HSN1\",HS);\nvar LA =({\"payload\":msg.payload.lux});\nvar RSSI =({\"payload\":msg.payload.rssi});\nvar Alerta =\"\";\nif (msg.payload.estados[0] ===0){\nAlerta += \"DHT11 \\\n\";\n}\nif (msg.payload.estados[1]===0){\nAlerta+=\"HC-SR04\\n\";\n}\nif (msg.payload.estados[2]===0){\nAlerta += \"YL38 \\\n\";\n}\nif (msg.payload.estados[3]===0){\nAlerta += \"BH1750 \\\n\";\n}\nif (Alerta !== \"\"){\nAlerta += \"Estos sensores no estan funcionando\";\n}\nmsg1={\npayload : Alerta,\ntopic : \"ALERTA Nodo Plantas 10\";\n}\nreturn [HA,TA,AA,HS,LA,RSSI,msg1,null];\n}\nAlerta=\"OK\";\nreturn[HA,TA,AA,HS,LA,RSSI,null,msg1];","outputs":8,"noerr":0,"x":676.7499656677246,"y":65,"wires":[["3f69145d.f5b68c","f291028f.72d28"],["525380d6.511e4","e372e94d.d0b328"],["38654e3f.1a4ff2","b2d95c71.3b777"],["9d6c13e2.2623e","a773215e.4e0e7"],["9b521032.0d99d","52dc1346.1fa68c"],["e83690f5.8d3ef"],["6ceae97f.200248","871a6471.5bc5d8"],["6ceae97f.200248"]]},{"id":"44ab3228.dd65cc","type":"function","z":"47ecbde4c4a764","name":"control_CC","func":"var NT =false;\nvar N1 =false;\nvar N2 =false;\nvar NTS =\"No Conecta\";\nvar N1S =\"No Conecta\";\nvar N2S =\"No Conecta\";\nvar altura_N1 = Flow.get(\"AAN1\");\nvar humedad_N1 = flow.get(\"HSN1\");\nvar altura_N2 = flow.get(\"AAN2\");\nvar humedad_N2 = Flow.get(\"HSN2\");\nvar altura_NT = flow.get(\"AANT\");\nvar ph_NT = flow.get(\"PH\");\nvar sensor_N1 = flow.get(\"SAAN1\");\nvar sensor_N2 = flow.get(\"SAAN2\");\nvar Alerta = \"apagar\";\nvar tiempo = new Date().toLocaleTimeString();\nfor (var i=0; i<3;i++){ if ( msg.payload.ID[i] === \"NTanque11\"){\nNT = true;\nNTS = \"conectado\"+tiempo;\n} else if (msg.payload.ID[i] === \"NPlantas10\"){\nN1 = true;\nN1S = \"conectado\"+tiempo;\n} else if (msg.payload.ID[i] === \"NPlantas11\"){\nN2 = true;\nN2S = \"conectado\"+tiempo;\n} }\nif (ph_NT.payload > 8){\nAlerta = \"apagar\";\n} else if (NT === true && (N1 === true || N2 === true)){\nif (sensor_N1 === 1 || sensor_N2 === 1){\nif (ph_NT.payload < 6 || humedad_N1.payload < 10 || humedad_N2.payload<10){\nAlerta=\"vacio\";\n} else{\nAlerta=\"apagar\";\n}\n}\n}\nreturn[msg1,msg2,msg3,msg4,msg5,msg6,msg7,msg8];","outputs":4,"noerr":0,"x":743.0000076293945,"y":499.50000762939453,"wires":[["573c2ca2.a909f4"],["b0b5c8cd.cda768"],["3e8dbe25.15aad2"],["76a78d05.27d454"],"f4
```

Continuación apéndice 2.

```
4c0c67.58e71"]],{"id":"83c8115b.cfb0f","type":"join","z":"47ecbde4a764","name":"esperamen
sajes","mode":"custom","build":"array","property":"payload.ID","propertyType":"msg","key":"topic"
,"joiner":"\n","joinerType":"str","accumulate":false,"timeout":"120","count":"","reduceRight":false,"
reduceExp":"","reducelnit":"","reducelnitType":"num","reduceFixup":"","x":508.36109924316406,"
y":455.05556297302246,"wires":[["9d5bbec1.a3d7c"]],{"id":"9d5bbec1.a3d7c","type":"change","
z":"47ecbde4a764","name":"filtra_ID","rules":[{"t":"delete","p":"payload.estados","pt":"msg"},{"t
":"delete","p":"payload.valores","pt":"msg"},{"t":"delete","p":"payload.tipo","pt":"msg"}],"action":"","
property":"","from":"","to":"","reg":false,"x":583.5000076293945,"y":499.25000762939453,"wires":
[[["44ab3228.dd65cc"]],{"id":"b0b5c8cd.cda768","type":"ui_text","z":"47ecbde4a764","group":
"c5d665b8.e6c208","order":7,"width":2,"height":3,"name":"ESNODO","label":"EstatusNodo","f
ormat":"{{msg.payload}}","layout":"colcenter","x":933.7500152587891,"y":534.2500076293945,"
wires":[]],{"id":"573c2ca2.a909f4","type":"ui_text","z":"47ecbde4a764","group":"62c99496.b87
60c","order":8,"width":2,"height":3,"name":"ESNODO","label":"EstatusNodo","format":"{{msg.
payload}}","layout":"colcenter","x":932.5000152587891,"y":498.50000762939453,"wires":[]],{"id":
"3e8dbe25.15aad2","type":"ui_text","z":"47ecbde4a764","group":"97e686a9.dbf448","order":8,
"width":2,"height":3,"name":"ESNODO","label":"EstatusNodo","format":"{{msg.payload}}","lay
out":"col-center","x":932.5000152587891,"y":570.5000095367432,"wires":[]],{"id":"9128db4e.38
5488","type":"ui_gauge","z":"47ecbde4a764","name":"","group":"62c99496.b8760c","order":5,"
width":3,"height":3,"gtype":"gage","title":"SeñalRouter","label":"dBm","format":"{{value}}","min
":"-100","max":"-30","colors":["#ff0000","#e6e600","#00ff00"],"seg1":"-88","seg2":"-67","x":945
.75,"y":851.5000858306885,"wires":[]],{"id":"d717834d.300a2","type":"change","z":"47ecbde4
a764","name":"CambiaNodo11","rules":[{"t":"move","p":"payload.valores[0]","pt":"msg","to":"pa
yload.humedadA","tot":"msg"},{"t":"move","p":"payload.valores[0]","pt":"msg","to":"payloa
d.temperaturaA","tot":"msg"},{"t":"move","p":"payload.valores[0]","pt":"msg","to":"payload.capacidad"
,"tot":"msg"},{"t":"move","p":"payload.valores[0]","pt":"msg","to":"payload.humedadS","tot":"msg"},{"t":
"move","p":"payload.valores[0]","pt":"msg","to":"payload.lux","tot":"msg"},{"t":"move","p":"payload.
valores[0]","pt":"msg","to":"payload.rssi","tot":"msg"},{"t":"delete","p":"payload.valores","pt":"msg"}
],"action":"","property":"","from":"","to":"","reg":false,"x":508.2500686645508,"y":295.0000057220
459,"wires":[["8d8f31fc.79b5d"],"afaaf214.1d372"]],{"id":"8d8f31fc.79b5d","type":"function","z":"4
7ecbde4a764","name":"JtoV_N11","func":"flow.set(\"SAAN2\",msg.payload.estados[1]);\nvar
HA ={\n\"payload\":msg.payload.humedadA};\nvar TA ={\n\"payload\": msg.payload.tempera
turaA};\nvar AA ={\n\"payload\":msg.payload.capacidad};\nflow.set(\"AAN2\",AA);\nvar HS ={\n\"pay
load\":msg.payload.humedadS};\nflow.set(\" HSN2\",HS);\nvar LA ={\n\"payload\":msg.payload
```

Continuación apéndice 2.

```
.lux} ;\nvar RSSI ={\n  payload\":msg.payload.rssi};\nvar Alerta ={\n  \"\n\nif (msg.payload.estados [0]===0) {\n  Alerta += \"DHT11\\n\\n\";\n}\nif (msg.payload.estados[1]===0){\n  Alerta += \"HC-SR04 \\n\\n\";\n}\nif (msg.payload.estados[2]===0){\n  Alerta += \"YL38\\n\\n\";\n}\nif (msg.pa yload.estados [3]===0){\n  Alerta += \"BH1750\\n\\n\";\n}\nif (Alerta !== \"\"){\n  Alerta += \"Estos sensores no estan funcionando\";\n}\nmsg1={\n  payload : Alerta,\n  topic : \"ALERTA Nodo Plantas 11!\\n\" }\n}; \nreturn [HA,TA,AA,HS ,LA,RSSI ,msg 1,null];\n}\n\nAlerta = \"OK\";\n\nreturn [HA,TA,AA,HS,LA,RSSI,null,msg={payload:Alerta}];\n\n,\"outputs\":8,\"noerr\":0,\"x\":685.00006103515 62,\"y\":294.49999713897705,\"wires\":[[\"bbb57e99.88be5\", \"f2f311a6.7a7e4\"],[\"6915b193.e9037\", \"9e320252.17c1d\"],[\"bfd8205e.68616\", \"8d387218.90653\"],[\"294277b.4bcd088\", \"e5273955.8501d 8\"],[\"8e44078c.7d22a8\", \"a62b0a90.cad678\"],[\"a0517896.4e3178\"],[\"4da41a94.cbc854\", \"871a64 71.5bc5d8\"],[\"4da41a94.cbc854\"]]],{\n  \"id\": \"e63d2a32.49ce48\", \"type\": \"change\", \"z\": \"47ecbde d.c4a7 64\", \"name\": \"CambiaNodoTan\", \"rules\": [{\n    \"t\": \"move\", \"p\": \"payload.valores[0]\", \"pt\": \"msg\", \"to\": \"payloa d.temperatura\", \"tot\": \"msg\" }, {\n    \"t\": \"move\", \"p\": \"payload.valores[0]\", \"pt\": \"msg\", \"to\": \"payload.altura\", \"t ot\": \"msg\" }, {\n    \"t\": \"move\", \"p\": \"payload.valores[0]\", \"pt\": \"msg\", \"to\": \"payload.conductividad\", \"tot\": \"msg\" }, {\n    \"t\": \"move\", \"p\": \"payload.valores[0]\", \"pt\": \"msg\", \"to\": \"payload.ph\", \"tot\": \"msg\" }, {\n    \"t\": \"move\", \"p\": \"pay load.valores[0]\", \"pt\": \"msg\", \"to\": \"payload.rssi\", \"tot\": \"msg\" }, {\n    \"t\": \"delete\", \"p\": \"payload.valores\", \"pt\": \" msg\" }], \"action\": \"\", \"property\": \"\", \"from\": \"\", \"to\": \"\", \"reg\": false, \"x\": 504.99999237060547, \"y\": 733.75002 0980835, \"wires\": [[\"12b26863.bfe248\", \"567befd.7bfea1\"]]], {\n  \"id\": \"12b26863.bfe248\", \"type\": \"functio n\", \"z\": \"47ecbde d.c4a764\", \"name\": \"JtoV_NT\", \"func\": \"var TA ={\n  payload\":msg.payload.temp eratura};\nvar AA ={\n  payload\":msg.payload.altur a};\n\nflow.set(\"AANT\",AA);\n\nvar CA ={\n  payload\":msg.payload.conductividad};\nvar PH ={\n  pa yload\":msg.pa yload.ph};\n\nflow. set(\"PH\",PH);\n\nvar RSSI ={\n  payload\":msg.payload.rssi};\n\nvar Alerta ={\n  \"\n\nif (msg.payload .estados[0]===0){\n  Alerta += \"DS19B20\\n\\n\";\n}\nif (msg.payloa d.estados[1]===0){\n  Alerta += \"HC-SR04\\n\\n\";\n}\nif (msg.payl oad.estados[2]===0){\n  Alerta += \"YL38\\n\\n\";\n}\nif (msg.payload.ph<0){\n  Alerta += \"pH SEN0161\\n\\n\";\n}\nif (Alerta !== \"\"){\n  Alerta += \"Estos sensores no estan funcionando\";\n}\n\nmsg1={\n  payload : Alerta,\n  topic : \"ALERTA Nodo Tanque!\\n\" }\n}; \n\nreturn [TA,AA,CA,PH,RSSI,msg1,null,0];\n}\n\nAlerta = \"OK\";\n\nreturn[TA,AA,CA,PH,RSSI,null,msg={payload:Alerta},0];\n\n,\"outputs\":8, \"noerr\":0,\"x\":694. 5040054321289,\"y\":747.4047698974609,\"wires\":[[\"ad2cc8a6.8dd7f8\", \"5c6e3ffd.6545\"],[\"98ed5e 9a.041b8\", \"675493f6.7a41cc\"],[\"7b60894f.f347a8\"],[\"2d005215.2c407e\", \"e398ec0f.57ff2\"],[\"912 8db4e.385488\"],[\"8b92882b.4d8128\", \"871a6471.5bc5d8\"],[\"8b92882b.4d8128\"],[\"d36fb83c.65b1 18\"]]], {\n  \"id\": \"a6d51cdc.9db79\", \"type\": \"debug\", \"z\": \"47ecbde d.c4a764\", \"name\": \"control llenado\", \"ac tive\": true, \"tosidebar\": true, \"console\": false, \"tostatus\": false, \"complete\": \"true\", \"x\": 1113.0000610351
```

Continuación apéndice 2.

```
562,"y":620.5000972747803,"wires":[]},{ "id":"46c98483.fc954c","type":"function","z":"47ecbde4a764","name":"T_lleno_N2","func":"var niv el ={\npayload\":msg.payload.nivel};\nvar humed ={\npayload\":msg.payload.humed};\nflow.set(\nAALL2\", nivel);\nreturn[nivel,humed];","outputs":2,"noerr":0,"x":695,"y":373,"wires":[[{"id":"bfd8205e.68616","8d387218.90653"},{"294277b.4bcd088","e5273955.8501d8"}]],{"id":"208884f0.c6c4bc","type":"join","z":"47ecbde4a764","name":"espera mensajes","mode":"custom","build":"array","property":"payload.ID","propertyType":"msg","key":"topic","joiner":"\n","joinerType":"str","accumulate":false,"timeout":"20","count":"","reduceRight":false,"reduceExp":"","reducelnit":"","reducelnitType":"num","reduceFixup":"","x":446.5555877685547,"y":556.3611278533936,"wires":[[{"e2062be2.4132e8"}]],{"id":"e2062be2.4132e8","type":"change","z":"47ecbde4a764","name":"filtra_ID","rules":[{"t":"delete","p":"payload.estados","pt":"msg"}, {"t":"delete","p":"payload.nivel","pt":"msg"}, {"t":"delete","p":"payload.tipo","pt":"msg"}, {"t":"delete","p":"payload.caudal","pt":"msg"}],"action":"","property":"","from":"","to":"","reg":false,"x":577.2500267028809,"y":605.0000267028809,"wires":[[{"632e2036.4eede"}]],{"id":"632e2036.4eede","type":"function","z":"47ecbde4a764","name":"control_LL","func":"var NT =false;\nvar N1 =false;\nvar N2 =false;\nvar NTS =\nNo Conecta\";\nvar N1S =\nNo Conecta\";\nvar N2S =\nNo Conecta\";\nvar altura_N1 = 0;\nvar altura_N2 = 0;\nvar altura_NT = 0;\nvar Alerta = \"\";\nvar tiempo = new Date().toLocaleTimeString();\nfor (var i=0; i<3;i++){ \n if (msg.payload.ID[i] === \nNTanque11\"){\n NT = true;\n NTS = \nconec tado\"+\ntiempo;\n altura_NT = flow.get(\nAALLT\");\n } else if (msg.payload.ID[i] === \nNPlantas10\"){\n N1 = true;\n N1S = \nconectado\"+\ntiempo;\n altura_N1 = flow.get(\nAALL1\");\n } else if (msg.payload.ID[i] === \nNPlantas11\"){\n N2 = true;\n N2S = \nconectado\"+\ntiempo;\n altura_N2 = Flow.get(\nAALL2\");\n } \n \nif (NT === true && (N1 === true || N2 === true)){ \n if ((altura_NT.payload < 15) || (altura_N1.payload > 90) || (altura_N2.payload > 90)){ \n Alerta = \nlleno\"; //llenoln } else {\n Alerta = \nvacio\";\n } \n } else {\n Alerta = \nlleno\"; //llenoln\nreturn [msg={payload:NTS},msg={payload:N1S},msg={payload:N2S},msg={topic:\nacuaponico/alerta\",payload:Alerta}];\n\n\n","outputs":4,"noerr":0,"x":736.7500267028809,"y":605.2500267028809,"wires":[[{"573c2ca2.a909f4"}, {"b0b5c8cd.cda768"}, {"3e8dbe25.15aad2"}, {"76a78d05.27d454"}, {"a6d51cdc.9db79"}]],{"id":"79cb1e9d.3af55","type":"function","z":"47ecbde4a764","name":"Tanque_lleno","func":"var nivel ={\npayload\":msg.payload.nivel};\nvar caudal ={\npayload\":msg.payload.caudal};\nflow.set(\nAALLT\", nivel);\nreturn[nivel,caudal];","outputs":2,"noerr":0,"x":702.7717132568359,"y":833.2909145355225,"wires":[[{"98ed5e9a.041b8","675493f6.7a41cc"}, {"57446118.cbb08","d36fb83c.65b118"}]],{"id":"f44c0c67.58e71","type":"debug","z":"47ecbde4a764","name":"control constante","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":"true","x":1120.1944656
```


Apéndice 3. Código Arduino del nodo de plantas

```
/*  
  Universidad de San Carlos de Guatemala, Facultad de Ingenieria  
  Escuela de Mecanica Electrica, Maynor Giovanni Ballina Escobar  
  Programacion para nodo cama de plantas  
  Microcontrolador: ESP32S Dev module  
  Sensores y modulos: DHT11; HC-SR04; YL38, YL69; GY-30, BH1750;  
  Alimentacion: 5V y 3.3V  
*/  
  
// Librerias para el proyecto  
#include <DHTesp.h>  
#include <HCSR04.h>  
#include <Wire.h>  
#include <BH1750.h>  
#include <WiFi.h>  
#include <PubSubClient.h>  
#include <ArduinoJson.h>  
  
//Definiendo elementos extra  
#define US_A_S 1000000  
#define TIEMPO_DORMIDO 1800 //en segundos 900 = 15 min  
#define TIEMPO_REBOOT 5 //reinicia dentro de 5 segundos  
  
//Definiendo pines  
#define LED 2  
#define SENSOR_TRIGGER 4  
#define SENSOR_ECHO 5  
#define DIP_1 12  
#define DIP_2 14  
#define I2C_SCL 21  
#define I2C_SDA 22  
#define DHT_DATO 25  
#define YL38_PODER 26  
#define YL38_DATO_DIGITAL 34  
#define YL38_DATO_ANALOGICO 36  
  
//datos guardados en RTC memory
```

Continuación apéndice 3.

```
RTC_DATA_ATTR int bootCount = 0;
// Constantes
const int PUERTO_MQTT = 1883;
const char* DATOS_PUBLICADOS;
const char* SSID_ROUTER = "SSIDROUTER";
const char* PASSWORD_MQTT = "PASSWORDMQTT";
const char* USUARIO_MQTT = "USUARIOMQTT";
const char* BROKER_IP = "IPBROKER";
const char* PASSWORD_ROUTER = "CLAVEROUTER";
// Declaraciones dependientes de librerias
DHTesp dht;
BH1750 BH1750_LIB;
HCSR04 HCSR04_DATO(SENSOR_TRIGGER, SENSOR_ECHO);
WiFiClient NPlantas;
PubSubClient client(NPlantas);
// Variables de medicion
int lux;
int capacidad;
int humedad_suelo;
int tiempo1, tiempo2;
int humedad_ambiente, temperatura_ambiente;
int estados_sensores[] = {0, 0, 0, 0}; //dht11, hcrs04, yl38, bh1750
long rssi;
char datos_recibidos[10];
String alerta = "";
String datos_en_json;
String mqtt_id = "NPlantas";
String MAIN_TOPIC = "acuaponico/plantas/";
//inicializa programa
void setup() { //porcion de pruebas
  Serial.begin(115200);
  delay(500);
  Serial.println();
```


Continuación apéndice 3.

```
//inicializar elementos
Wire.begin(I2C_SCL, I2C_SDA);
BH1750_LIB.begin();
// Declaracion de pines
dht.setup(DHT_DATO);
pinMode(DIP_1, INPUT);
pinMode(DIP_2, INPUT);
pinMode(YL38_DATO_DIGITAL, INPUT);
pinMode(YL38_DATO_ANALOGICO, INPUT);
pinMode(LED, OUTPUT);
pinMode(YL38_PODER, OUTPUT);
//comunicacion
obtener_id();
conecta_wifi();
conecta_mqtt();}
void loop() { // programa
digitalWrite(LED, HIGH);
monitoreo_constante();
alerta = "";
while (alerta == "") {client.loop();
  delay(50);}
Serial.println(alerta);
if (alerta == "vacio") {
  while (alerta == "vacio") {
    alerta = "";
llenando();
delay(50);
if (!client.connected()) {
  conecta_mqtt();}
client.subscribe("acuaponico/alerta/#");
while (alerta == "") {
  client.loop();
  delay(50);}}
```

Continuación apéndice 3.

```
} else if (alerta == "apagar") {
  digitalWrite(LED, LOW);
  Serial.println(millis() / 1000.0);
  a_dormir(TIEMPO_DORMIDO);
} else if (alerta == "lleno") {
  digitalWrite(LED, LOW);
  Serial.println(millis() / 1000.0);
  a_dormir(TIEMPO_DORMIDO); }}
//CONEXIONES
void conecta_wifi() { //a wifi
  rssi = 0;
  digitalWrite(LED, HIGH);
  delay(200);
  digitalWrite(LED, LOW);
  delay(200);
  digitalWrite(LED, HIGH);
  delay(200);
  digitalWrite(LED, LOW);
  int contar = 0;
  WiFi.begin(SSID_ROUTER, PASSWORD_ROUTER);
  while (WiFi.status() != WL_CONNECTED) { //verifica si conecta
    Serial.print(contar);
    //al no conectar despues de n intentos resetea el micro
    if (contar == 20) {
      Serial.println(" no conecta");
      a_dormir(TIEMPO_REBOOT);
    } else { //mientras no conecta lleva conteo de intentos
      Serial.println(" Conectando a wifi ...");
      contar = contar + 1;
      delay(500); }}
//prueba al conectar muestra conectado
Serial.println("Conexion establecida");
rssi = WiFi.RSSI();
```

Continuación apéndice 3.

```
Serial.print("RSSI ");
Serial.println(rssi);
mide_tiempo();}
//obtiene id del dipsw
void obtener_id() {
  mqtt_id += !digitalRead(DIP_1);
  mqtt_id += !digitalRead(DIP_2); }
//a mqtt
void conecta_mqtt() {
  digitalWrite(LED, HIGH);
  delay(200);
  digitalWrite(LED, LOW);
  delay(200);
  digitalWrite(LED, HIGH);
  delay(200);
  digitalWrite(LED, LOW);
  int _contar = 0;
  //establece quien es el servidor mqtt
  client.setServer(BROKER_IP, PUERTO_MQTT);
  client.setCallback(callback);
  while (!client.connected()) { //verifica coneccion mqtt
    Serial.println("Conectando a MQTT...");
    if (client.connect(mqtt_id.c_str(), USUARIO_MQTT, PASSWORD_MQTT )) {
      Serial.println("conectado");
    } else {
      Serial.print("broker no responde ...");
      Serial.println(client.state());
      digitalWrite(LED, HIGH);
      delay(300);
      digitalWrite(LED, LOW);
      delay(300);
      _contar = _contar + 1;
      Serial.println(_contar);
    }
  }
}
```

Continuación apéndice 3.

```
//al no conectar despues de n intentos reinicia el micro
if (_contar == 6) {
    a_dormir(TIEMPO_REBOOT); }}}
client.subscribe("acuaponico/alerta/#");
mide_tiempo();}
// FUNCIONES DE SENSORES
void dato_temperatura_humedad () {
    float _humedad_dato = 0.00, _humedad_suma = 0.00;
    float _temperatura_dato = 0.00, _temperatura_suma = 0.00;
    for (int i = 0; i < 3; i++) {
        delay(dht.getMinimumSamplingPeriod());
        _humedad_dato = dht.getHumidity();
        _temperatura_dato = dht.getTemperature();
        _humedad_suma = _humedad_suma + _humedad_dato;
        _temperatura_suma = _temperatura_suma + _temperatura_dato; }
    humedad_ambiente = _humedad_suma / 3;
    temperatura_ambiente = _temperatura_suma / 3;
    // Pruebas
    Serial.print(humedad_ambiente, 2);
    Serial.print(" %HR; ");
    Serial.print(temperatura_ambiente, 2);
    Serial.println(" °C");
    mide_tiempo();}
void dato_llenado_agua () {
    float _distancia_cm = 0.00, _distancia_suma = 0.00, _distancia_promedio = 0.00;
    for (int i = 0; i < 5; i++) {
        _distancia_cm = HCSR04_DATO.dist();
        _distancia_suma = _distancia_suma + _distancia_cm;
        delay(100); }
    _distancia_promedio = _distancia_suma / 5;
    capacidad = (14 - _distancia_promedio) * 10;
    //Pruebas
    Serial.print(_distancia_promedio);
```

Continuación apéndice 3.

```
Serial.print(" cm -> "); //cm
Serial.print(capacidad);
Serial.println(" %"); //cm
mide_tiempo();}
void dato_humedad_suelo () {
  int _humedad_medir = 0, _humedad_suma = 0, _humedad_analogo = 0;
  humedad_suelo = 0;
  for (int i = 0; i < 15; i++) {
    _humedad_medir = analogRead(YL38_DATO_ANALOGICO);
    _humedad_suma = _humedad_suma + _humedad_medir;
    delay(50); }
  _humedad_analogo = _humedad_suma / 15;
  humedad_suelo = map(_humedad_analogo, 4095, 977, 0, 100);
  //Pruebas
  Serial.print(_humedad_analogo);
  Serial.print(" val analogo -> ");
  Serial.print(humedad_suelo);
  Serial.println(" %H");}
void dato_intensidad_luz () {
  float _lux_medido = 0, _lux_suma = 0;
  for (int i = 0; i < 5; i++) {
    _lux_medido = BH1750_LIB.readLightLevel();
    _lux_suma = _lux_suma + _lux_medido;
    delay(100); }
  lux = _lux_suma / 5;
  //pruebas
  Serial.print(lux);
  Serial.println(" luxes");}
void verifica_funcionamiento() { //funcionamiento de cada sensor
  if (dht.getStatusString() == "OK") { //DHT11
    estados_sensores[0] = 1;
  } else {
    estados_sensores[0] = 0;
```

Continuación apéndice 3.

```
    humedad_ambiente = 0;
    temperatura_ambiente = 0; }
if (HCSR04_DATO.dist() > 3) { //HC-SR04
    estados_sensores[1] = 1;
} else {
    estados_sensores[1] = 0;
    capacidad = 0; }
if (digitalRead(YL38_DATO_DIGITAL) == 1) { //YL38
    estados_sensores[2] = 1;
} else {
    estados_sensores[2] = 0;
    humedad_suelo = 0; }
Wire.beginTransmission(35); //decimal de 0x23 direccion del bh1750
if (Wire.endTransmission() == 0) { //BH1750
    estados_sensores[3] = 1;
} else {
    estados_sensores[3] = 0;
    lux = 0; }}
//CONTROL
void monitoreo_constante() {
    //Despues de llenarse el la cama se monitorea contrantemene por cambios
    digitalWrite(YL38_PODER, HIGH);
    dato_temperatura_humedad();
    dato_llenado_agua();
    dato_humedad_suelo();
    dato_intensidad_luz();
    verifica_funcionamiento();
    delay(100);
    mqtt_control();
    digitalWrite(YL38_PODER, LOW);
    mide_tiempo();}
void llenando() {
    digitalWrite(YL38_PODER, HIGH);
```

Continuación apéndice 3.

```
dato_llenado_agua();
dato_humedad_suelo();
verifica_funcionamiento();
delay(200);
mqtt_llenado();
digitalWrite(YL38_PODER, LOW); }
//COMUNICACION Y COMPLEMENTOS
void mqtt_control() { //genero formato json y envio datos via mqtt
  StaticJsonBuffer<300> jB;//creando el objeto principal json
  JsonObject& root = jB.createObject();
  root["ID"] = mqtt_id;
  root["tipo"] = "control";
  JsonArray& D1 = root.createNestedArray("estados");
  D1.add(estados_sensores[0]);
  D1.add(estados_sensores[1]);
  D1.add(estados_sensores[2]);
  D1.add(estados_sensores[3]);
  JsonArray& D2 = root.createNestedArray("valores");
  D2.add(humedad_ambiente);
  D2.add(temperatura_ambiente);
  D2.add(capacidad);
  D2.add(humedad_suelo);
  D2.add(lux);
  D2.add(rssi);
  //convierto json a string
  root.printTo(datos_en_json);
  DATOS_PUBLICADOS = datos_en_json.c_str();
  jB.clear();
  client.publish(MAIN_TOPIC.c_str(), DATOS_PUBLICADOS);
  //pruebas
  Serial.println(datos_en_json);
  delay(200);
  datos_en_json = "";
```

Continuación apéndice 3.

```
DATOS_PUBLICADOS = "";}
void mqtt_llenado() {
    StaticJsonBuffer<200> jB;
    JsonObject& root = jB.createObject();
    root["ID"] = mqtt_id;
    root["tipo"] = "llenando";
    root["nivel"] = capacidad;
    root["humed"] = humedad_suelo;
    root.printTo(datos_en_json);
    DATOS_PUBLICADOS = datos_en_json.c_str();
    client.publish(MAIN_TOPIC.c_str(), DATOS_PUBLICADOS);
    Serial.println(datos_en_json);
    jB.clear();
    datos_en_json = "";
    DATOS_PUBLICADOS = "";
    delay(200); }
void callback(char* topic, byte* payload, unsigned int length) {
    alerta = "";
    Serial.print("Message arrived in topic: ");
    Serial.println(topic);
    Serial.print("Message:");
    for (int i = 0; i < length; i++) {
        datos_recibidos[i] = (char)payload[i];
        Serial.print((char)payload[i]); }
    Serial.println();
    Serial.println("-----");
    alerta = String(datos_recibidos); }
void a_dormir(int tiempo) {
    if (tiempo == 5) {
        ++bootCount;
    } else {
        bootCount = 0; }
    if (bootCount == 8) {
```


Continuación apéndice 3.

```
Serial.println(" espera media hora ");
bootCount = 0;
tiempo = TIEMPO_DORMIDO; }
//dormira el microcontrolador por el tiempo estipulado
Serial.println("a dormir");
esp_sleep_enable_timer_wakeup(tiempo * US_A_S);
esp_deep_sleep_start();}
void mide_tiempo() {
tiempo2 = millis() - tiempo1;
tiempo1 = millis();
Serial.print(tiempo2 / 1000.0);
Serial.println(" segundos");}
```

Fuente: elaboración propia.

Apéndice 4. Código Arduino del nodo de tanque

```
/*
Universidad de San Carlos de Guatemala, Facultad de Ingenieria
Escuela de Mecanica Electrica, Maynor Giovanni Ballina Escobar
Programacion para nodo cama de plantas
Microcontrolador: ESP32S Dev module
Sensores y modulos: DS19B20, HC-SR04, YL38, YL69, SEN0161, FS300A
Alimentacion: 3.3V y 5V
*/
// Librerias para el proyecto
#include <HCSR04.h>
#include <Wire.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <WiFi.h>
#include <PubSubClient.h>
```

Continuación apéndice 4.

```
#include <ArduinoJson.h>
//Definiendo constantes
#define US_A_S 1000000
#define TIEMPO_DORMIDO 1800 //en segundos 900 = 15 min
#define TIEMPO_REBOOT 5 //reinicia dentro de 5 segundos
//Definiendo pines
#define LED 2
#define SENSOR_TRIGGER 4
#define SENSOR_ECHO 5
#define DIP_1 12
#define DIP_2 14
#define PH_DATO 15
#define FS300_DATO 18
#define BOMBA_PODER 22
#define PH_PODER 23
#define YL38_PODER 25
#define DS19_DATO 26
#define YL38_DATO_DIGITAL 34
#define YL38_DATO_ANALOGO 36
//datos guardados en RTC memory
RTC_DATA_ATTR int bootCount = 0;
// Constantes
const int PUERTO_MQTT = 1883;
const float FACTOR_K_CAUDAL = 5.5;
const float VOLTAJE_CALIBRACION_PH4 = 2032.44;
const float VOLTAJE_CALIBRACION_PH7 = 1500.0;
const char* DATOS_PUBLICADOS;
const char* SSID_ROUTER = "SSIDROUTER";
const char* PASSWORD_MQTT = "PASSWORDMQTT";
const char* USUARIO_MQTT = "USUARIOMQTT";
const char* BROKER_IP = "IPBROKER";
const char* PASSWORD_ROUTER = "CLAVEROUTER";
// Declaraciones dependientes de librerias
```

Continuación apéndice 4.

```
OneWire oneWire(DS19_DATO);
DallasTemperature TEMPERATURA(&oneWire);
HCSR04 HCSR04_DATO(SENSOR_TRIGGER, SENSOR_ECHO);
WiFiClient NTanque;
PubSubClient client(NTanque);
// Variables de medicion
int frecuencia;
int tiempo1, tiempo2;
int estados_sensores[] = {0, 0, 0}; //DS19B20, hcrs04, yl38
float altura_agua;
int temperatura_agua;
float valor_ph, valor_voltaje;
float conductividad_medicion;
float frecuencia_caudal, caudal_L_m, caudal_L_h;
long rssi;
char datos_recibidos[10];
String alerta = "";
String datos_en_json;
String mqtt_id = "NTanque";
String MAIN_TOPIC = "acuaponico/tanque/";
volatile int conteo_pulsos;
//inicializa programa
void setup() {
  //porcion de pruebas
  Serial.begin(115200);
  delay(500);
  Serial.println();
  //inicializar elementos
  TEMPERATURA.begin();
  // Declaracion de pines
  pinMode(DIP_1, INPUT);
  pinMode(DIP_2, INPUT);
  pinMode(FS300_DATO, INPUT);
```

Continuación apéndice 4.

```
pinMode(YL38_DATO_ANALOGO, INPUT);
pinMode(YL38_DATO_DIGITAL, INPUT);
pinMode(LED, OUTPUT);
pinMode(PH_PODER, OUTPUT);
pinMode(YL38_PODER, OUTPUT);
pinMode(BOMBA_PODER, OUTPUT);
attachInterrupt(digitalPinToInterrupt(FS300_DATO), cuenta_pulsos, RISING);
//comunicacion
dato_ph();
conecta_wifi(); //Inicializa wifi
obtener_id(); //obtiene id para mqtt
conecta_mqtt(); //inicia mqtt}
//programa principal
void loop() {
  // programa
  digitalWrite(LED, HIGH);
  monitoreo_constante();
  while (alerta == "") {
    client.loop();
    delay(50); }
  Serial.println(alerta);
  if (alerta == "vacio") {
    digitalWrite(BOMBA_PODER, HIGH);
    while (alerta == "vacio") {
      alerta = "";
      llenando();
      delay(50);
      if (!client.connected()) {
        conecta_mqtt();}
      client.subscribe("acuaponico/alerta/#");
      while (alerta == "") {
        client.loop();
        delay(50); }}
```

Continuación apéndice 4.

```
} else if (alerta == "apagar") {
  digitalWrite(LED, LOW);
  Serial.println(millis() / 1000.0);
  a_dormir(TIEMPO_DORMIDO);
} else if (alerta == "lleno") {
  digitalWrite(LED, LOW);
  digitalWrite(BOMBA_PODER, LOW);
  Serial.println(millis() / 1000.0);
  a_dormir(TIEMPO_DORMIDO); }}
//CONEXIONES
void conecta_wifi() { //a wifi
  rssi = 0;
  digitalWrite(LED, HIGH);
  delay(200);
  digitalWrite(LED, LOW);
  delay(200);
  digitalWrite(LED, HIGH);
  delay(200);
  digitalWrite(LED, LOW);
  int _contar = 0;
  WiFi.begin(SSID_ROUTER, PASSWORD_ROUTER);
  while (WiFi.status() != WL_CONNECTED) { //verifica si conecta
    Serial.print(_contar);
    //al no conectar despues de n intentos resetea el micro
    if (_contar == 20) {
      Serial.println(" no conecta");
      a_dormir(TIEMPO_REBOOT);
    } else { //mientras no conecta lleva conteo de intentos
      Serial.println(" Conectando a wifi ...");
      _contar = _contar + 1;
      delay(500); } }
//prueba al conectar muestra conectado
Serial.println("Conexion establecida");
```

Continuación apéndice 4.

```
    rssi = WiFi.RSSI();
Serial.print("RSSI ");
    Serial.println(rssi);
    mide_tiempo();}
//obtiene id del dipsw
void obtener_id() {
    mqtt_id += !digitalRead(DIP_1);
    mqtt_id += !digitalRead(DIP_2); }
//a mqtt
void conecta_mqtt() {
    digitalWrite(LED, HIGH);
    delay(200);
    digitalWrite(LED, LOW);
    delay(200);
    digitalWrite(LED, HIGH);
    delay(200);
    digitalWrite(LED, LOW);
    int _contar = 0;
    //establecer quien es el servidor mqtt
    client.setServer(BROKER_IP, PUERTO_MQTT);
    client.setCallback(callback);
    while (!client.connected()) { //verifica coneccion mqtt
        Serial.println("Conectando a MQTT...");
        if (client.connect(mqtt_id.c_str(), USUARIO_MQTT, PASSWORD_MQTT )) {
            Serial.println("conectado");
        } else {
            Serial.print("broker no responde ...");
            Serial.println(client.state());
            digitalWrite(LED, HIGH);
            delay(300);
            digitalWrite(LED, LOW);
            delay(300);
            _contar = _contar + 1;
```

Continuación apéndice 4.

```
Serial.println(_contar);
//al no conectar despues de n intentos reinicia el micro
if (_contar == 6) {
    a_dormir(TIEMPO_REBOOT); }}}
client.subscribe("acuaponico/alerta/#");
mide_tiempo();}
// FUNCIONES DE SENSORES
void dato_temperatura () {
    Serial.print(TEMPERATURA.getDeviceCount(), DEC);
    Serial.print(" sensor; ");
    TEMPERATURA.requestTemperatures();
    temperatura_agua = TEMPERATURA.getTempCByIndex(0);
    Serial.print(temperatura_agua);
    Serial.println(" °C");}
void dato_altura_agua () {
    float _distancia_cm = 0.00, _distancia_suma = 0.00, _distancia_promedio = 0.00;
    for (int i = 0; i < 5; i++) {
        _distancia_cm = HCSR04_DATO.dist();
        _distancia_suma = _distancia_suma + _distancia_cm;
        delay(100); }
    _distancia_promedio = _distancia_suma / 5;
    altura_agua = 30 - _distancia_promedio;
    //Pruebas
    Serial.print(_distancia_promedio);
    Serial.print(" cm; "); //cm
    Serial.print(altura_agua);
    Serial.println(" cm"); //cm
    mide_tiempo();}
void dato_conductividad () {
    int _conductividad_medir = 0, _conductividad_suma = 0, _conductividad_analogo = 0;
    float _voltaje_medido = 0;
    conductividad_medicion = 0;
    for (int i = 0; i < 10; i++) {
```

Continuación apéndice 4.

```
_conductividad_medir = analogRead(YL38_DATO_ANALOGO);
_conductividad_suma = _conductividad_suma + _conductividad_medir;
delay(50); }
_conductividad_analogo = _conductividad_suma / 10;
_voltaje_medido = _conductividad_analogo * (3.3 / 4095);
conductividad_medicion = ((3.3 - _voltaje_medido) / (_voltaje_medido * 5100)) * 104.2;
//Pruebas
Serial.print(_voltaje_medido);
Serial.print(" V, ");
Serial.print(conductividad_medicion, 3);
Serial.println(" conductividad");}
void dato_caudal () {
frecuencia_caudal = obtiene_frecuencia();
caudal_L_m = frecuencia_caudal * FACTOR_K_CAUDAL;
caudal_L_h = caudal_L_m * 60;
//pruebas
Serial.print ("FrecuenciaPulsos: ");
Serial.print (frecuencia_caudal, 0);
Serial.print ("Hz\tCaudal: ");
Serial.print (caudal_L_m, 2);
Serial.print (" L/m\t");
Serial.print (caudal_L_h, 2);
Serial.println ("L/h");}
void dato_ph() {
digitalWrite(PH_PODER, HIGH);
delay(500);
valor_voltaje = 0.0;
float _voltaje_sum = 0.0, _voltaje_calc = 0.0, _voltaje_negado = 0.0;
for (int i = 0; i < 5; i++) {
delay(500);
_voltaje_calc = analogRead(PH_DATO) / 4095.0 * 3300.0;
_voltaje_sum = _voltaje_sum + _voltaje_calc; }
digitalWrite(PH_PODER, LOW);
```


Continuación apéndice 4.

```
_voltaje_negado = _voltaje_sum / 5.0;
valor_voltaje = 3300.0 - _voltaje_negado;
calibracion_ph(valor_voltaje);
Serial.print(_voltaje_negado);
Serial.print(" voltaje negado; ");
Serial.print(valor_voltaje);
Serial.print(" voltaje; ");
Serial.print(valor_ph);
Serial.println(" pH");
Serial.print(analogRead(PH_DATO));
Serial.println(" pH");}

void verifica_funcionamiento() { //verificando el funcionamiento de cada elemento
  if (TEMPERATURA.getDeviceCount() == 1) { //DS19B20
    estados_sensores[0] = 1;
  } else {
    estados_sensores[0] = 0;
    temperatura_agua = 0; }
  if (HCSR04_DATO.dist() > 3) { //HC-SR04
    estados_sensores[1] = 1;
  } else {
    estados_sensores[1] = 0;
    altura_agua = 0; }
  if (digitalRead(YL38_DATO_DIGITAL) == 0) { //YL38
    estados_sensores[2] = 1;
  } else {
    estados_sensores[2] = 0;
    conductividad_medicion = 0; }}

//CONTROL
void llenando() {
  dato_altura_agua();
  dato_caudal();
  verifica_funcionamiento();
  delay(200);
```

Continuación apéndice 4.

```
mqtt_llenado();}
void monitoreo_constante() {
  digitalWrite(YL38_PODER, HIGH);
  dato_temperatura();
  dato_conductividad();
  verifica_funcionamiento();
  digitalWrite(YL38_PODER, LOW);
  dato_altura_agua();
  delay(500);
  mqtt_control();
  mide_tiempo();}
//COMUNICACION Y COMPLEMENTOS
void mqtt_control() { //genero formato json y envio datos via mqtt
  StaticJsonBuffer<350> jB;//creando el objeto principal json
  JsonObject& root = jB.createObject();
  root["ID"] = mqtt_id;
  root["tipo"] = "control";
  JsonArray& D1 = root.createNestedArray("estados");
  D1.add(estados_sensores[0]);
  D1.add(estados_sensores[1]);
  D1.add(estados_sensores[2]);
  JsonArray& D2 = root.createNestedArray("valores");
  D2.add(temperatura_agua);
  D2.add(altura_agua);
  D2.add(conductividad_medicion);
  D2.add(valor_ph);
  D2.add(rssi);
  //convierto json a string
  root.printTo(datos_en_json);
  DATOS_PUBLICADOS = datos_en_json.c_str();
  jB.clear();
  client.publish(MAIN_TOPIC.c_str(), DATOS_PUBLICADOS);
  //pruebas
```

Continuación apéndice 4.

```
Serial.println(datos_en_json);
delay(200);
datos_en_json = "";
DATOS_PUBLICADOS = "";
Serial.println("TERMINE");}
void mqtt_llenado() {
    StaticJsonBuffer<200> jB;
    JsonObject& root = jB.createObject();
    root["ID"] = mqtt_id;
    root["tipo"] = "llenando";
    root["nivel"] = altura_agua;
    root["caudal"] = caudal_L_m;
    root.printTo(datos_en_json);
    DATOS_PUBLICADOS = datos_en_json.c_str();
    client.publish(MAIN_TOPIC.c_str(), DATOS_PUBLICADOS);
    Serial.println(datos_en_json);
    jB.clear();
    datos_en_json = "";
    DATOS_PUBLICADOS = "";
    delay(200); }
void callback(char* topic, byte* payload, unsigned int length) {
    Serial.print("Message arrived in topic: ");
    Serial.println(topic);
    Serial.print("Message:");
    for (int i = 0; i < length; i++) {
        datos_recibidos[i] = (char)payload[i];
        Serial.print((char)payload[i]); }
    Serial.println();
    Serial.println("-----");
    alerta = String(datos_recibidos); }
void a_dormir(int tiempo) {
    if (tiempo == 5) {
        ++bootCount;
```

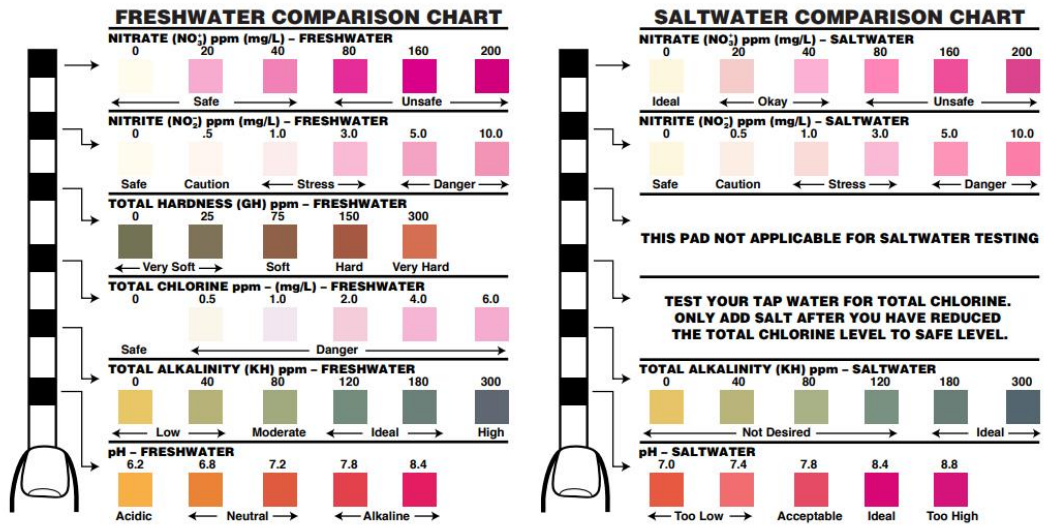
Continuación apéndice 4.

```
} else {
  bootCount = 0; }
if (bootCount == 8) {
  Serial.println(" espera media hora ");
  bootCount = 0;
  tiempo = TIEMPO_DORMIDO; }
//dormira el microcontrolador por el tiempo estipulado
Serial.println("a dormir");
esp_sleep_enable_timer_wakeup(tiempo * US_A_S);
esp_deep_sleep_start();}
void cuenta_pulsos() {
  conteo_pulsos++;}
int obtiene_frecuencia() {
  conteo_pulsos = 0;
  interrupts();
  delay(1000);
  noInterrupts();
  frecuencia = conteo_pulsos;
  return frecuencia;}
float calibracion_ph(float voltaje) { //en base a los dos puntos determinados por el fabricante
  float _pendiente_calibracion = (7.0 - 4.0) / ((VOLTAJE_CALIBRACION_PH7 - 1500.0) / 3.0 -
(VOLTAJE_CALIBRACION_PH4 - 1500.0) / 3.0);
  float _punto_intercepcion = 7.0 - _pendiente_calibracion * (VOLTAJE_CALIBRACION_PH7 -
1500.0) / 3.0;
  valor_ph = _pendiente_calibracion * (voltaje - 1500.0) / 3.0 + _punto_intercepcion;}
void mide_tiempo() {
  tiempo2 = millis() - tiempo1;
  tiempo1 = millis();
  Serial.print(tiempo2 / 1000.0);}
```

Fuente: elaboración propia.

ANEXOS

Apéndice 4. Lámina de colores de Tetra Easy Strips



Fuente: BIGOS. *TETRA easystrips 6 in 1*. <https://www.losbigos.cl/bienestar/2173-tetra-easystrips-6-en-1.html>. Consulta: 10 de septiembre de 2018

