



POSGRADO EN INGENIERÍA AGRÍCOLA Y USO INTEGRAL DEL AGUA

**SISTEMA DE CONTROL Y MONITOREO DE PH MEDIANTE
INTERNET DE LAS COSAS EN ACUAPONIA**

TESIS DE GRADO

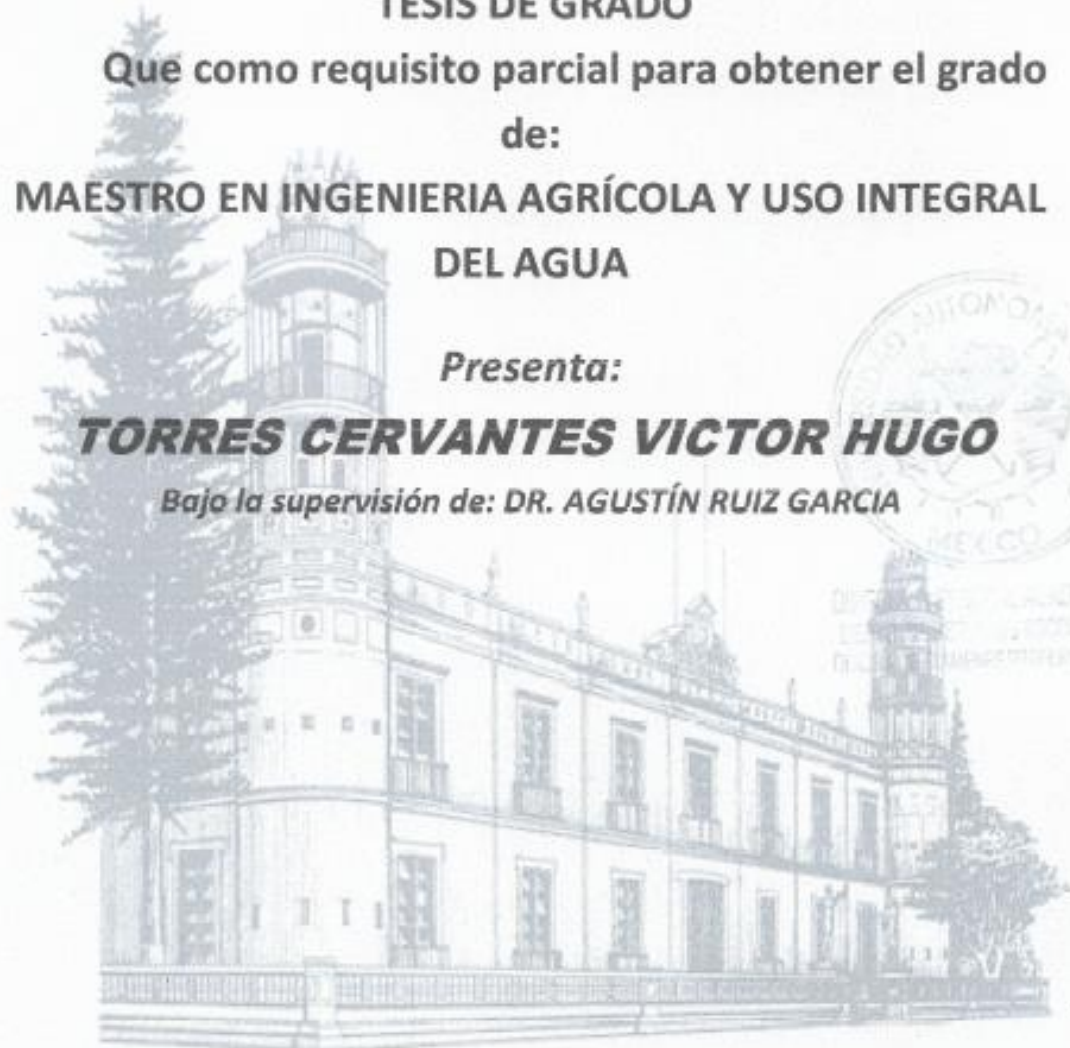
Que como requisito parcial para obtener el grado
de:

**MAESTRO EN INGENIERIA AGRÍCOLA Y USO INTEGRAL
DEL AGUA**

Presenta:

TORRES CERVANTES VICTOR HUGO

Bajo la supervisión de: DR. AGUSTÍN RUIZ GARCIA



Chapingo, Estado de México, Noviembre de 2020.



**SISTEMA DE CONTROL Y MONITOREO DE PH MEDIANTE INTERNET DE
LAS COSAS EN ACUAPONIA**

Tesis realizada por **TORRES CERVANTES VICTOR HUGO** bajo la supervisión del Comité Asesor indicado, aprobada por el mismo y aceptada como requisito parcial para obtener el grado de:

MAESTRO EN INGENIERÍA AGRÍCOLA Y USO INTEGRAL DEL AGUA

DIRECTOR:



DR. AGUSTÍN RUIZ GARCÍA

ASESOR:



DR. JOEL PINEDA PINEDA

ASESOR:



M.I. JUAN CARLOS OLGÚN ROJAS

Contenido

LISTA DE CUADROS.....	v
LISTA DE FIGURAS	v
DATOS BIOGRÁFICOS	xi
GENERAL ABSTRACT	xiii
1. INTRODUCCIÓN GENERAL	1
1.1. Objetivo general.....	5
1.3. Estructura de la tesis	6
1.4. Literatura citada.....	7
2. REVISIÓN DE LITERATURA	9
2.1. Introducción	9
2.2. Acuaponía.....	10
2.2.1. Tipo de sistemas acuapónicos.....	11
2.2.2. Parámetros de un sistema acuapónico	12
2.2.3. pH en un sistema acuapónico.....	13
2.3. Internet de las cosas.....	15
2.3.1. Internet de las cosas en la agricultura	16
2.4. Conclusiones.....	18
2.5. Literatura citada	19
3. DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO Y CONTROL DE PH CON TECNOLOGÍA DE INTERNET DE LAS COSAS PARA ACUAPONIA	21
3.1. Introducción	22
3.2. Materiales y Métodos	23
3.2.1. Estación de monitoreo.....	23
3.2.2. Gateway.....	24
3.2.3. Raspberry Pi.....	27
3.2.4. Sensor de pH Omega	28
3.2.5. Convertidor Analógico a Digital ADC ADS1115.....	31
3.2.6. Módulo de relevadores para bombas.....	33
3.2.7. Nube de Datos Firebase	34

3.2.8.	Estructura de la nube de datos Firebase mediante Python	35
3.2.9.	Creación archivos csv mediante la librería Pandas.....	36
3.2.10.	Envío de reportes de monitoreo mediante Gmail.....	41
3.2.11.	Envío de datos de lectura del sensor de pH a Firebase.....	46
3.2.12.	Control de bombas a través de la nube de datos Firebase	48
3.2.13.	Aplicación móvil en Android Studio	50
3.2.14.	HelloCharts	51
	Sensor de temperatura DS18B20 y ultrasónico HC-SR04.....	51
3.3.	Resultados y discusión	52
3.3.1.	Comunicación general del sistema IoT.....	52
3.3.2.	Lectura del sensor de pH	53
3.3.3.	Visualización de datos del Gateway en la estación de Monitoreo ..	54
3.3.4.	Estructura de la nube de datos Firebase.....	56
3.3.5.	Obtención de datos de la estación de monitoreo	57
3.3.6.	Creación y envío de Reportes de Monitoreo (csv) mediante Gmail	58
3.3.7.	Aplicación móvil Android	61
3.4.	Conclusiones	65
3.5.	Literatura citada	67
4.	ANEXOS	69
	ANEXO 1. Costo del sistema IoT	69
	ANEXO 2. Librerías Python	72
	ANEXO 3. Librerías Android Studio.....	73
	ANEXO 4. GPIO de la placa Raspberry utilizados en el Software.	75

LISTA DE CUADROS

Cuadro 1. Resultados de criterios búsqueda en bases de datos científicas sobre internet de las cosas (Pérez et al., 2019).....	18
Cuadro 2. Costos de elementos del Gateway.....	69
Cuadro 3. Costo de la nube de datos por un año.	71
Cuadro 4. Librerías utilizadas en el módulo Python del sistema IoT.....	72
Cuadro 5. Librerías utilizadas para la aplicación móvil en Android Studio.....	73
Cuadro 6. GPIO utilizados en el sistema IOT.	75

LISTA DE FIGURAS

Figura 1. Sistemas acuapónicos según el tipo de cama de crecimiento del cultivo (Wongkiewa, Hub, Chandranc, Woo & Kumar, 2017).....	12
Figura 2. Proceso de nitrificación en un sistema acuapónico (fuente propia)... ..	14
Figura 3. Cama de crecimiento y estanque del sistema acuapónico.	24
Figura 4. Esquema de funcionamiento de los hilos del código de programación.	26
Figura 5. Tarjeta Raspberry Pi 4 model B.....	28
Figura 6. Electrodo Omega PHE-1304.	29
Figura 7. Representación del funcionamiento del bulbo de un electrodo de pH.	30
Figura 8. Módulo PH-4502C.	30
Figura 9. Convertidor analógico a Digital ADS115.....	31
Figura 10. Conexión de SDA y SCL de Raspberry y ADS115.	32

Figura 11. Esquema de adaptación de protocolo de comunicación I2C por medio de las líneas SDA y SCL.....	32
Figura 12. Módulo de 4 relevadores	33
Figura 13. Bomba de diafragma de 12 volts	33
Figura 14. Ejemplo de estructura de base de datos en Firebase creada mediante Python.....	36
Figura 15. Método de hilo para la eliminación de archivos al arranque del programa.	37
Figura 16. Flujograma del hilo de Python para la creación y escritura del archivo "Reporte_diario.csv".....	39
Figura 17. Flujograma del hilo de Python para la creación y escritura del archivo "Base_de_datos.csv".....	40
Figura 18. Configuración Gmail para permitir el acceso de aplicaciones.....	41
Figura 19. Diagrama de flujo del método de hilo Python para enviar mensaje de prueba a través de Gmail.....	42
Figura 20. Diagrama de flujo del método de hilo Python para el envío automático vía Gmail del reporte de monitoreo diario en formato csv	43
Figura 21. Método de hilo de solicitud de reporte de monitoreo diario por Firebase.....	45
Figura 22. Subproceso de envío de reporte de monitoreo diario solicitado por el método de hilo.	46
Figura 23. Método de hilo de lectura y envío de dato de pH a Firebase.....	47
Figura 24. Método de hilo para la activación de bombas por Firebase.....	49
Figura 25. Subproceso para activar bomba por el tiempo determinado por el usuario.	50
Figura 26. Logo Android Studio.	51
Figura 27. Sensores DS18B20 y HC-SR04	52

Figura 28. Estructura general de comunicación del sistema IoT.....	53
Figura 29. Conexión de sensor de pH con la placa Raspberry.....	54
Figura 30 Estado de arranque exitoso del módulo Python del Gateway.....	55
Figura 31. Arranque con error de software en el módulo Python del Gateway.	55
Figura 32. Estructura en formato JSON de la nube de datos Firebase.....	56
Figura 33. Fallo en hilo en módulo python.	58
Figura 34. Código para enviar reporte de monitoreo mediante Gmail automáticamente.	59
Figura 35. Recepción de Reporte Diario mediante Gmail en celular.	59
Figura 36. Notificación en el Gateway de preparación, envío mediante Gmail, eliminación y comienzo de nuevo de reporte diario.	60
Figura 37. Notificación de eliminación de reportes al ejecutar el programa IoT.	61
Figura 38. Recepción de mensaje de prueba para comprobar la conexión del Gateway con Gmail.....	61
Figura 39. Interfaz gráfica de la aplicación móvil.	62
Figura 40. Obtención de reportes de monitoreo a través de Gmail mediante la aplicación móvil.....	62
Figura 41. Ejemplo de la configuración de la escala del eje “Y” en la grafica de aplicación móvil.....	63
Figura 42. Encendido de bomba mediante la aplicación Android.	64
Figura 43. Interfaz de Gateway con el mensaje de encendido/ Apago de Bombas.....	64
Figura 44. Cambio de tiempo de encendido de bombas desde la aplicación Android.	64
Figura 45. Prueba de encendido de Bombas mediante android, Firebase y python.	65

Figura 46. Importación de librerías en el módulo Python.....	73
Figura 47. Importación de librerías en el IDE de Android Studio.	75

DEDICATORIAS

Dedico este trabajo especialmente a mi madre:

Flor Cervantes Sánchez

Por su apoyo incondicional y por nunca dejar de creer en mí.

A mi padre Hugo Torres Román.

A mis hermanos Guadalupe, Vanessa, Geovany y Daniela. Esperando que este trabajo sea una inspiración para seguir adelante y continuar sus estudios.

AGRADECIMIENTOS

Agradezco a la Universidad Autónoma Chapingo por el apoyo brindado durante toda esta etapa como alumno de maestría.

Al Posgrado de Ingeniería Agrícola y Uso Integral del Agua por brindar todas las bases y herramientas necesarias para el desarrollo educativo a través de su programa.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo económico que brinda a estudiantes como yo para poder realizar posgrados de calidad.

Al Dr. Agustín Ruiz García por su distinguida dirección, apoyo y consejos para la realización y culminación de este proyecto

Al Dr. Joel Pineda Pineda y al M.I. Juan Carlos Olguín Rojas por su gran asesoramiento y observaciones realizadas en este trabajo.

DATOS BIOGRÁFICOS



Datos personales

Nombre: Víctor Hugo Torres Cervantes.
Fecha de nacimiento: 21 de junio de 1994.
Lugar de nacimiento: Ciudad de México.
CURP: TOCV940621HDFRRC05
Profesión: Ingeniero Mecánico Agrícola.
Cedula Profesional: 11434351

Desarrollo académico

Bachillerato: Preparatoria Agrícola, Universidad Autónoma Chapingo (2009 -2012).

Licenciatura: Ingeniería Mecánica Agrícola, Universidad Autónoma Chapingo (2012 - 2016).

RESUMEN GENERAL

SISTEMA DE CONTROL Y MONITOREO DE PH MEDIANTE INTERNET DE LAS COSAS EN ACUAPONIA¹

La acuaponía es un sistema de producción en el que se busca una interacción simbiótica entre sus elementos biológicos (plantas, peces y bacterias) controlando parámetros físicos y químicos. El monitoreo diario de estos parámetros brinda una solución para mantener una buena calidad de agua en el sistema. Dentro de estos parámetros está el pH del agua, que controla y regula el metabolismo de los peces, la actividad nitrificante de las bacterias y la solubilidad de nutrientes. La implementación de tecnología basada en Internet de las Cosas (IoT) permite al usuario monitorear parámetros de manera remota debido a la comunicación que ofrece entre el usuario y los objetos (sensores, actuadores). Este trabajo tuvo como objetivo desarrollar un sistema de monitoreo y control del pH de un sistema acuapónico utilizando tecnología IoT, desarrollando una aplicación móvil que permita al usuario acceder al monitoreo en tiempo real, obtener el historial de mediciones durante el día o un ciclo de tiempo determinado y que de esta manera se pueda efectuar la acción de actuadores para corregir el parámetro y llevarlo a un rango de trabajo óptimo para el sistema. Se desarrolló una intercomunicación vía WIFI del sistema IoT vinculando una placa Raspberry, con lenguaje de programación Python, a una nube de datos creada por google llamada Firebase, y esta a su vez con la plataforma Android Studio para el desarrollo de una aplicación móvil. Los resultados de la intercomunicación permiten al usuario visualizar mediante la aplicación móvil la lectura de los sensores en tiempo real, acceder a archivos csv con las lecturas tomadas por el sensor durante el día o en un lapso de tiempo determinado, activar actuadores durante un tiempo determinado en la misma aplicación para corregir el parámetro monitoreado.

Palabras clave: Acuaponía, Internet de las cosas, Python, Firebase, Raspberry, Android.

¹ Tesis de Maestría en Ingeniería Agrícola y Uso Integral del Agua

Autor: Torres-Cervantes Víctor Hugo

Director de Tesis: Ruiz-García Agustín

GENERAL ABSTRACT

PH CONTROL AND MONITORING SYSTEM THROUGH THE INTERNET OF THINGS IN ACUAPONIA²

Aquaponics is a production system in which a symbiotic interaction between its biological elements (plants, fish and bacteria) is sought by controlling physical and chemical parameters. The daily monitoring of these parameters provides us with a solution to maintain good water quality in the system. It is found that, within these parameters, pH of water controls and regulates the metabolism of fish, the nitrifying activity of bacteria and governs the solubility of nutrients. The implementation of Internet- based technology of Things (IoT) allows the user to monitor parameters remotely due to the communication it offers between the user and the objects (sensors, actuators). This work aims to develop a system for monitoring and controlling the pH of an aquaponic system using IoT, developing a mobile application that allows the user to access real-time monitoring, obtain the measurement history during the day or a determined cycle of time and in this way the action of actuators can be carried out to correct the parameter and bring it to an optimal working range for the system. A WIFI intercommunication of the IoT system was developed by linking a Raspberry board, with Python programming language, to a data cloud created by google called Firebase, and this in turn with the Android Studio platform for the development of a mobile application. The results of the intercom allow the user to visualize through the mobile application the reading of the sensors in real time, access csv files with the readings taken by the sensor during the day or within a specific period of time, activate actuators for a given time in the same application to correct the monitored parameter.

Keywords: Aquaponics, Internet of Things, Python, Firebase, Raspberry, Android.

² Thesis, Universidad Autónoma Chapingo

Autor: Torres-Cervantes Víctor Hugo

Advisor: Ruiz-García Agustín

1. INTRODUCCIÓN GENERAL

Actualmente son muchas las adversidades a las que se enfrenta la agricultura a nivel mundial: la sobrepoblación, el cambio climático, la contaminación y escases de agua en ciertas regiones del planeta destacan entre los factores más importantes que limitan la producción de alimentos. La necesidad de producir masivamente alimentos de excelente calidad para cubrir la demanda de la población, en espacios cada vez más reducidos ha llevado al sector agropecuario a la investigación y desarrollo de tecnologías que permitan la intensificación de producción, como la acuaponía, un sistema sustentable y redituable en los que se aprovecha de manera sustentable el agua para la producción de peces y plantas en un mismo espacio (Ronzón, Hernández & Pérez, 2012).

La acuaponía es un sistema de producción de alimentos en el que se combina la acuicultura con la hidroponía (Ronzón et al., 2012). Este sistema se compone por tres elementos esenciales que interactúan entre sí: plantas, peces y bacterias. Los desechos metabólicos producidos por los peces son convertidos en nutrientes asimilables por las plantas gracias a las bacterias (Caló, 2011).

En México los aztecas cultivaban maíz, zapallo y otras plantas en un sistema conocido como “chinampas”, los cuales consistían en pequeñas islas artificiales. Los canales que rodeaban estas se usaban para criar peces, los cuales producían fertilizantes naturales mediante sus desechos metabólicos para las plantas. Esta es una práctica de acuaponía de forma inicial en México (Cutíño, 2018).

Actualmente la acuaponía a nivel nacional se ha establecido como uno de los sistemas de agricultura protegida, brindando beneficios económicos y una fuente de alimentos con alto valor nutricional (Rodríguez, Rubio, García, Montoya & Magullón, 2015). Cada vez en más estados de la República se integran proyectos de investigación y desarrollo de este sistema de producción.

Los estados de Chihuahua y Estado de México están desarrollando este sistema de producción en donde se utilizan los desechos orgánicos de la tilapia (*Oreochromis niloticus*) para producir alimentos y con ello reducir el impacto ambiental generados por la actividad acuícola de estos peces de manera independiente (Sánchez, 2017). En el estado de Sinaloa se desarrollan sistemas acuapónicos produciendo tilapia y lechuga (*Lactuca sativa* L.) para evaluar tecnologías de recirculación y tratamiento de agua mediante la producción de biomasa del sistema (Rodríguez et al., 2015). En el estado de Veracruz se realizan investigaciones en sistemas acuapónicos abiertos, donde se produce tilapia y tomate (*Solanum lycopersicum* L.) variedad saladette, para evaluar sustratos como el tezontle y la composta y diferentes tipos de agua de riego (agua potable y agua acuícola), relacionando los resultados con el crecimiento y supervivencia de los peces y el crecimiento del cultivo (Ortega, Trejo, Gómez, Alonso & Salazar, 2015).

El sector agrícola ha generado interés por la integración de sensores y comunicación en los sistemas de producción con la finalidad de monitorear parámetros que deben ser medidos constantemente. Los avances tecnológicos de sensores con redes inalámbricas de comunicación juegan un papel importante en la agricultura ya que facilita la posibilidad de monitorear y controlar variables vitales para los sistemas agrícolas (Trinidad et al., 2018).

Estos sistemas para monitorear, procesar y controlar variables físicas del sistema agrícola, que permiten la transmisión de datos en tiempo real de manera local o remota para el usuario interesado son desarrollados mediante el paradigma tecnológico llamado Internet de las cosas (IoT) con el cual se pueden tomar decisiones rápidamente de acuerdo a la información obtenida por la red de sensores con las que cuenta el sistema agrícola (Archbold et al., 2019).

Para lograr que un sistema acuapónico sea sostenible es necesario equiparlo con tecnología que puedan optimizar y monitorear parámetros importantes para la

estabilidad y eficiencia, por ejemplo, el uso de sensores permite monitorear variables físico-químicas que permiten tener una buena calidad de agua. Por ello, cada día más, los productores con sistemas de acuaponía se interesan e invierten en tecnología que les permita tener el control y monitoreo del sistema, buscando obtener una mayor eficiencia del agua y una mayor productividad (Rodríguez et al., 2015).

Una de las principales desventajas en la acuaponía es el manejo diario (Cutíño, 2018) para mantener en niveles óptimos parámetros del sistema que cambian constantemente y afectan a la producción del sistema. Caló (2011) menciona que para mantener una buena calidad de agua en un sistema acuapónico deben medirse parámetros físico-químicos como lo son: temperatura, oxígeno disuelto y pH de manera diaria, y parámetros como: niveles de nitrógeno amoniacal, nitritos y nitratos de manera periódica, esto con la finalidad de asegurar una óptima interacción planta-bacteria-peces.

En la acuaponía la fuente de nutrientes de las plantas esta generada por los desechos metabólicos de los peces. Sólo un 35 a 40% del alimento que es consumido por los peces es asimilado por su metabolismo, el resto del alimento consumido es excretado. La cantidad de nutrientes disponible está relacionada en primera instancia con la cantidad de alimento que ingieren los peces y a su vez con la cantidad de plantas que puede sostener el sistema (Caló, 2011).

Pineda et al. (2017) mencionan que una preocupación importante en los sistemas acuapónicos es la eliminación de amoníaco el cual es generado por los desechos metabólico de los peces. El amoníaco, el nitrito y el nitrato producidos por el sistema son tóxicos para los peces a diferentes niveles. Las plantas del sistema acuapónico controlan los niveles de nitrato, mientras que los niveles de amoníaco y el nitrito son regulados por un proceso de nitrificación llevado a cabo por las bacterias benéficas (*Nitrosomonas sp* y *Nitrobacter sp.*) presentes en el sistema (Nelson, 2008).

El pH como parámetro para mantener la calidad del agua del sistema tiene varias funciones, pero la más importante es que interviene en el proceso de nitrificación (Tyson, 2007). De igual manera afecta la solubilidad de los nutrientes para las plantas presentes en el agua y afecta el desarrollo metabólico de los peces que se utilizan en el sistema (Calo, 2011).

Con base a lo expuesto anteriormente, mantener niveles de pH óptimos en el sistema generará una excelente actividad nitrificante y un buen desarrollo de los peces. Esto tiene como consecuencia una mayor tasa de alimentación de estos, lo cual provocará una mayor cantidad de desechos metabólicos, lo que significa una mayor cantidad de nutrientes para las plantas y así evitar la utilización de fertilizantes químicos externos al sistema, reduciendo el impacto ambiental del sistema de producción.

Esta investigación tuvo como objetivo desarrollar un sistema automático de control y monitoreo de pH en tiempo real utilizando tecnología IoT, con el fin de mantener este parámetro en niveles óptimos. Esto para asegurar el flujo de nutrientes para las plantas, ya que la actividad nitrificante de las bacterias aumenta la eficiencia de producción del sistema y asegurando la calidad del agua para los elementos que componen al sistema acuapónico.

1.1. Objetivo general

- Diseñar e implementar una red de comunicación IoT que permita monitorear el pH de un sistema acuapónico a través de una aplicación Android.

1.2. Objetivos particulares

- Diseñar un Gateway de fácil acceso a internet y de bajo costo que se pueda instalar en un sistema acuapónico.
- Intercomunicar inalámbricamente la estación de monitoreo a la base de datos alojada en Firebase y esta a su vez con la aplicación móvil de Android Studio.
- Seleccionar las herramientas necesarias de software y hardware que permitan una fácil conexión del sistema IoT y una implementación sencilla.
- Elaborar una interfaz gráfica de monitoreo en tiempo real del pH de fácil comprensión y acceso para el usuario.
- Vincular la red de comunicación IoT a Gmail para el envío de reportes de monitoreo mediante archivos con extensión csv.

1.3. Estructura de la tesis

Este trabajo de investigación se realizó en seis etapas. La primera fue una revisión de literatura sobre el estado del arte de la acuaponía y la aplicación de tecnología IoT en la agricultura. La segunda etapa consistió en desarrollar el sistema de comunicación IoT; en esta etapa se descargaron las librerías, se crearon las credenciales y atributos necesarios para que la estación de monitoreo se comunicara remotamente con la nube de datos y esta a su vez con una aplicación Android. En la tercera etapa se desarrolló el software para poder monitorear el pH del agua, enviar los datos a través de la nube de datos a la aplicación Android; en esta etapa se instalaron las librerías y el hardware necesarios para poder realizar las mediciones del pH. En la cuarta etapa se instalaron los paquetes necesarios para crear archivos con formato csv con los datos de monitoreo, comunicar el sistema a Gmail y de esta manera enviar los archivos csv; en la interfaz de la aplicación Android se creó una gráfica para observar el comportamiento del parámetro a través del tiempo y por último se realizó una comunicación de la aplicación Android a la estación de monitoreo para activar actuadores y solicitar los reportes de monitoreo. En la última etapa se realizaron las pruebas de comunicación general del sistema IoT.

1.4. Literatura citada

- Archbold, G. W., Beltrán, A. H., Ruiz, F., Narducci, M. S., Méndez, D., Trujillo, L. C., Parra, C., Carrillo & Mouazen, A. M. (2019). pH measurement IoT system for precision agriculture applications. *IEEE Latin America Transactions*, 17(5), 823-832.
- Caló P. (2011). *Introducción a la acuaponía*. Centro Nacional de Desarrollo acuícola (CENADAC). Recuperado de: http://www.minagri.gob.ar/site/pesca/acuicultura/06_Publicaciones/_archivos/130423_Introducci%C3%B3n%20a%20la%20ACUAPONIA.pdf
- Cutiño, V. (2018). *Acuaponia como alternativa productiva social* (Tesis de pregrado). Facultad de Ciencias Veterinarias UNCPBA, Buenos Aires, Argentina.
- Nelson, R. (2008). Aquaponic equipment: the bio filter. *Aquaponics Journal*, 48, 22–23.
- Ortega L., N.E., Trejo T., L.I., Gómez M., F.C., Alonso L., A. & Salazar O., J. (2015). Crecimiento de tilapia (*Oreochromis niloticus*) y tomate (*Solanum lycopersicum* L.) en un sistema acuapónico abierto. *Agroproductividad*, 8, 20-25.
- Pineda, J., Miranda, I., Rodríguez, J., Ramirez, J., Perez, E., Garcia, I. & Morales, J. (2017). Nutritional balance in aquaponic lettuce production. *Acta Hort.*, 1170, 1093-1100, doi: 10.17660/ActaHortic.2017.1170.141.
- Rodríguez G., H. Rubio C., S.G., García U., M., Montoya M., M. & Magullón B., F.J. (2015). Análisis técnico de la producción de tilapia (*Oreochromis niloticus*) y lechuga (*Lactuca sativa*) en dos sistemas de acuaponia. *Agroproductividad*, 8, 15-19.
- Ronzón O., M., Hernández V., M.P. & Pérez R., C.I. (2012). Producción hidropónica y acuapónica de albahaca (*Ocimum basilicum*) y langostino malayo (*Macrobrachium rosenbergu*). *Tropical and Subtropical Agroecosystems*, 15, 63-71.
- Sánchez J., A. (2017). Sistema de producción sostenible de plantas y peces (acuaponia): Caso de México. *Monfrague Desarrollo Resiliente*, 8, 152-157.
- Trinidad, G., Meis, V., López, C., Italo, J., Ríos, C., Saldaña, G., Hernández, J., & Molina, M. (2018). Wireless Sensor Network for Monitoring Physical Variables Applied to Green Technology (IoT Green Technology). *European*

Journal of Electrical and Computer, Engineering, 2,1-7, doi:
10.24018/ejece.2018.2.2.15.

Tyson, R. (2007). *Reconciling pH for ammonia biofiltration in a cucumber/tilapia aquaponics system using a perlite medium* (Tesis doctoral) Universidad de Florida, Estados Unidos. Recuperado de:
http://etd.fcla.edu/UF/UFE0019861/tyson_r.pdf.

2. REVISIÓN DE LITERATURA

Resumen

El desarrollo e implementación de tecnología en la agricultura protegida es una necesidad que cubre demanda la producción de alimentos para optimizar los recursos naturales y económicos. La acuaponía es un sistema de producción agrícola en el cual se deben monitorear y regular diaria y periódicamente parámetros físico-químicos como: pH, temperatura, oxígeno disuelto, niveles de nitrógeno amoniacal (N-NH_3 , N-NH_4^+), nitratos (N-NO_3^-) y nitritos (N-NO_2^-). El internet de las cosas (IoT) es una herramienta tecnológica que ofrece el monitoreo y el control de parámetros físico-químicos de manera remota a través de aplicaciones web o móviles. Por lo anterior, el objetivo de este trabajo es realizar una revisión de literatura para conocer el estado del arte en el que se encuentra la aplicación de tecnología basada en Internet de las cosas en los sistemas agrícolas. Se describen los sistemas acuapónicos actuales que se encuentran en la agricultura y se mencionan los principales parámetros físico-químicos que afectan estos. De acuerdo con la literatura consultada, la aplicación de tecnología IoT es una herramienta que cubre las necesidades de control y monitoreo de los sistemas acuapónicos, la cual puede llegar a reducir considerablemente los costos de producción y el impacto ambiental.

Palabras claves: acuaponía, internet de las cosas, agricultura protegida.

2.1. Introducción

El principal reto de la agricultura es satisfacer la demanda de alimentos de la población en continuo crecimiento, manejando de manera sustentable los recursos naturales con los que se cuenta. Para lograr lo anterior, este sector tiene que llevar a cabo la implementación de tecnologías que mantenga un sistema de producción eficiente y redituable, teniendo como uno de los objetivos principales generar un impacto ambiental bajo, como es el caso de la acuaponía (Gutiérrez, Montiel, Carreto & Menchaca, 2019).

La acuaponia es un sistema de producción de alimentos que combina la acuicultura y la hidroponía en un mismo sistema integrado (Pineda et al., 2017). Este modelo de producción puede controlar ciertas variables físico-químicas con el objetivo de mejorar la calidad y la producción (Aguirre, Monje, Sierra & Vega, 2018).

Para lograr un sistema acuapónico redituable es necesario tecnificarlo, para lograr un control óptimo de las variables físico-químicas que afectan de manera directa la producción y funcionamiento de este. Cada vez más productores se interesan por sistemas de monitoreo remoto que les permita conocer el estado actual de su sistema de producción y de esta manera, tomar las acciones necesarias para mantener los parámetros del sistema en niveles óptimos (Rodríguez, Rubio, García, Montoya & Magullón, 2015).

Los sistemas de control y monitoreo basados en el paradigma tecnológico de Internet de las Cosas (IoT) ofrecen la transmisión de datos en tiempo real y de manera remota, además de que muchos de estos sistemas son de bajo costo, haciendo redituable su desarrollo e instalación en la agricultura (Archbold et al., 2019).

2.2. Acuaponía

La acuaponía es un sistema recirculante, integrado y controlado, de producción de alimentos en el que se cultivan plantas y peces simultáneamente, combinando la hidroponía con la acuicultura. Este sistema se basa en la interacción de 3 elementos biológicos: plantas, peces y bacterias nitrificantes los cuales deben sobrevivir de manera simbiótica ya que cada uno necesita diferentes parámetros ambientales (Pérez, 2018).

En este sistema el agua que recircula contiene nutrientes provenientes de los desechos metabólicos de los peces, los cuales son procesados químicamente por las bacterias nitrificantes y de esta manera son aprovechados por las plantas. La combinación de estos dos sistemas proporciona una solución a uno de los principales problemas en cada uno; la materia orgánica que se acumula por los desechos metabólicos de los peces es filtrada por las plantas aportándole la mayoría de los nutrientes necesarios en forma iónica para su desarrollo, y estas

a su vez actúan como un biofiltro que mantiene la calidad del agua en condiciones óptimas para el desarrollo de los peces (Zarzoza, 2018). De igual manera, Roosta & Hamidpour (2013) mencionan que el objetivo principal de un sistema acuapónico es reutilizar los nutrientes liberados por los peces para producir plantas.

La acuaponía surge como alternativa de solución para las principales necesidades de producción agrícola, como lo son: ahorro de agua, disminuir el impacto ambiental de los sistemas de producción, disminuir el uso de fertilizantes químicos (López, 2017). Al reducir el uso de agua, así como la cantidad de pesticidas y fertilizantes este sistema puede definirse como un sistema de producción orgánica dando un valor agregado a sus productos finales (Pineda et al., 2017).

2.2.1. Tipo de sistemas acuapónicos

En un sistema de acuaponía se tienen los siguientes elementos básicos necesariamente: un tanque de peces, un biofiltro y una cama o lecho de cultivo. Caló (2011) menciona que hay tres sistemas acuapónicos más comúnmente utilizados y los clasifica según su cama de crecimiento: técnica de película de nutrientes (NTF), balsa flotante y rellenos de medios o lecho de sustrato (Figura 1).

Sistema NTF: este sistema consiste en correr una película de solución nutritiva a lo largo de un canal de cultivo y al terminar su recorrido esta regresa al estanque de peces. Las plantas son contenidas en un recipiente de plástico o similar el cual está suspendido sobre el canal, permitiendo que las raíces de las plantas tengan contacto con la película que corre sobre este.

Balsas flotantes: en este sistema el lecho de cultivo se encuentra suspendido sobre el reservorio de agua o estanque.

Relleno de medios: en este sistema el lecho de cultivo se monta sobre un contenedor lleno de un sustrato inerte que servirá para el sostén de las plantas. El agua es llevada del estanque al contenedor mediante tuberías accionadas por bombas, y es retornada mediante gravedad por tuberías instaladas en el fondo del contenedor. Este sistema no requiere biofiltros separados ya que el relleno del contenedor actúa propiamente como uno y en el mismo se utiliza un sifón que sirve como medio de aireación (Zou et al., 2016).

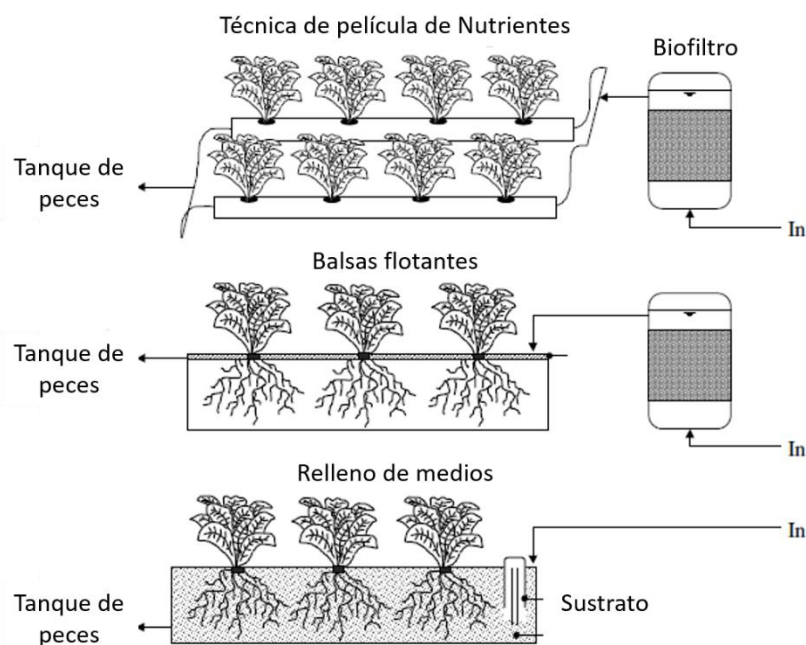


Figura 1. Sistemas acuapónicos según el tipo de cama de crecimiento del cultivo (Wongkiewa, Hub, Chandranc, Woo & Kumar, 2017).

2.2.2. Parámetros de un sistema acuapónico

En un sistema acuapónico uno de los principales puntos de interés es mantener una buena calidad de agua, ya que este medio es el encargado de que las bacterias nitrificantes puedan realizar su actividad de manera óptima, generando una taza de nutrientes asimilables para las plantas, provocando una reacción en cadena que devuelve agua “limpia” al estanque. Caló (2011) menciona que

existen parámetros que deben ser medidos diariamente y otros periódicamente para asegurar una buena calidad de agua y que el sistema sea eficiente. Los parámetros que deben medir diariamente son: pH, temperatura y oxígeno disuelto, mientras que los parámetros que deben ser medidos periódicamente son: niveles de nitrógeno amoniacal (N-NH_3 , N-NH_4^+), de nitratos (N-NO_3^-) y nitritos (N-NO_2^-), los cuales pueden llegar a niveles que pueden ser tóxicos para los peces.

2.2.3. pH en un sistema acuapónico

El pH es un parámetro de suma importancia en cualquier sistema acuático. En la acuaponía este es el principal factor que controla y regula el metabolismo de los peces, la actividad microbiana y afecta directamente la disponibilidad de nitrógeno para las plantas. Además, este parámetro gobierna la solubilidad de otros nutrientes como el calcio, fósforo, potasio, magnesio, etc., que son nutrientes importantes para el óptimo desarrollo de las plantas (Resh, 2012).

El alimento para los peces que se utiliza en los sistemas acuapónicos agrega una gran cantidad de proteínas al sistema que son digeridos por los peces. Estos excretan la mayor parte del nitrógeno consumido en forma de amoníaco total (TAN). El TAN es la suma de amonio no ionizado (NH_3) y amonio en forma ionizada (NH_4^+) que produce el sistema en el metabolismo de las proteínas realizada por los peces y la descomposición orgánica de los desechos sólidos (Caló, 2011). El TAN es un compuesto tóxico para los peces, pero este es convertido en nitrato (NO_3^-) a través del proceso químico de nitrificación de la actividad bacteriana del mismo sistema y esta es la forma nitrogenada menos tóxica para los peces. Según Caló (2011) el amonio no ionizado es extremadamente tóxico para los peces y su concentración en el sistema depende del pH y la temperatura del agua.

Las plantas solo pueden asimilar nitrógeno mediante dos formas: el nitrato (NO_3^-) y el amonio (NH_4^+) pero son mayormente asimilables los nitratos. El proceso de nitrificación del sistema se lleva a cabo mediante dos pasos: Primero el amoniaco es convertido en nitrito (NO_2^-) y luego este en nitrato (NO_3^-) (Figura 2). El nitrato se reduce a dinitrógeno (N_2) mediante un proceso de desnitrificación por las bacterias, el cual es un nutriente de suma importancia para el desarrollo del cultivo (Zou et al., 2016). Un buen proceso de nitrificación en el sistema acuapónico además de producir nitrógeno como nutriente para las plantas generará ácido carbónico el cual provocará que el pH disminuya en el sistema (Caló, 2011).

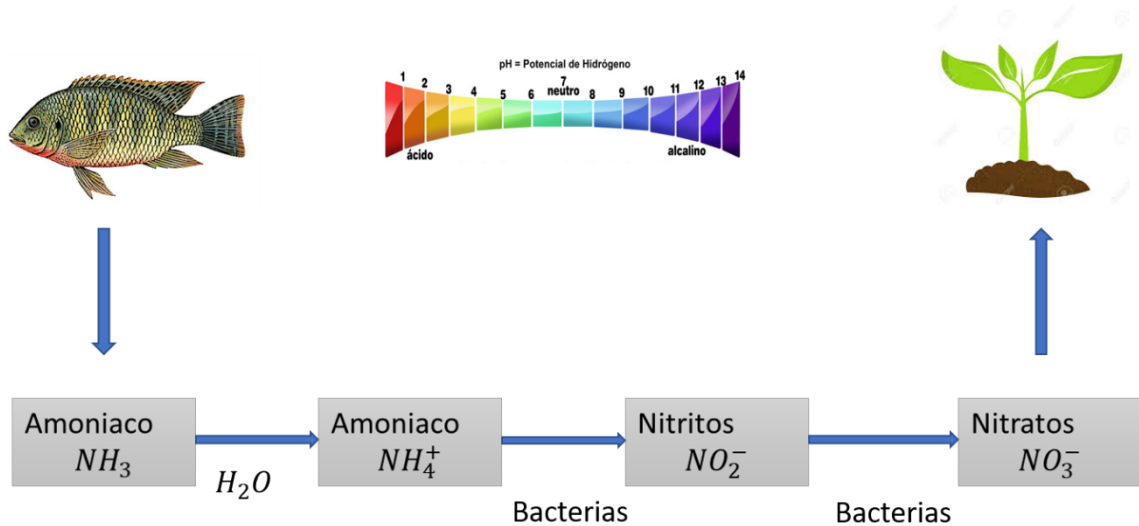


Figura 2. Proceso de nitrificación en un sistema acuapónico (fuente propia).

Zou et al. (2016) mencionan que el pH es uno de los factores de regulación más importante de un sistema acuapónico y es necesario equilibrarlo para peces, plantas y bacterias nitrificantes al mismo tiempo. Las plantas requieren generalmente un pH ligeramente ácido, entre 5.5 a 5.8, mientras que el pH de las bacterias nitrificantes es de 7.5 a 8.0 y los peces manejan un rango óptimo de pH dependiendo la especie. Zou et al. (2016) citan un rango óptimo para un sistema acuapónico de 7.5 a 8.0 para una relación simbiótica de los 3 organismos vivos del sistema (planta, pez, bacteria).

Caló (2011) menciona que el rango óptimo para el proceso de Nitrificación es de 7.2 a 7.8, sin embargo, este proceso se puede dar entre un rango de 6.0 a 9.0. De la misma manera menciona que la disponibilidad de nutrientes de la planta como el hierro, manganeso, cobre, zinc y boro se encuentran escasos con valores de pH mayores a 7.5; mientras que el fósforo, calcio y magnesio se ven afectados con valores de pH menores a 6.0. Por lo anterior este autor recomienda un valor de pH de 7.0 para un buen manejo del sistema.

2.3. Internet de las cosas

La evolución del internet a lo largo del tiempo desde su existencia ha sufrido cambios y etapas que tienen como resultado aumentar las aplicaciones de este en la industria, la vida social y en la economía. Mercado & Gonzales (2018) y Mena (2018) mencionan que el internet actualmente se puede dividir en 4 fases de desarrollo y aplicación: la primera fase denominada “conectividad” donde el uso del correo electrónico, la búsqueda de información y contenido por medio de la navegación web fue posible. La segunda etapa “economía interconectada” llegó a finales de los años noventa y es caracterizada por comercio electrónico, es decir, el uso de páginas web para la venta-compra de productos y servicios. La tercera etapa “experiencias cooperativas” es donde el uso de redes sociales, redes de movilidad, servicios de video y la computación en la nube aparecen transformando el mundo laboral completamente. Y la cuarta fase es denominada “Internet de las Cosas” y es donde las personas, procesos, datos y objetos están conectados y en comunicación en tiempo real.

Rouse (2017) define a el Internet de las Cosas (IoT, por sus siglas en inglés) como un sistema de dispositivos de computación interrelacionados, máquinas mecánicas y digitales, objetos, animales o personas que tienen identificadores únicos y la capacidad de transferir datos a través de una red, sin requerir de interacciones humano a humano o humano a computadora. Las nuevas

tecnologías digitales hacen posible recolectar y aprovechar enormes cantidades de datos críticos a un costo mínimo, logrando así una operación agrícola más orientada a la información y potencialmente más productiva y eficiente.

Un sistema IoT incorpora 4 pilares: personas, procesos, datos y objetos. Esta red de conexión a través de internet de manera remota permite que la toma de decisiones y acciones en un sistema sean más precisas creando nuevas experiencias de control y monitoreo y oportunidades económicas mayores (Mena, 2018).

Como en otras implementaciones del Internet de las Cosas, las soluciones para agricultura protegida de capturar, almacenar, analizar los datos y presentarlos en forma que se puedan ejecutar acciones apropiadas por parte del usuario final, son necesarias para mantener parámetros y condiciones óptimas del sistema de producción.

2.3.1. Internet de las cosas en la agricultura

El sector agrícola ha generado interés por la integración de redes de sensores y actuadores en los diferentes sistemas de producción existentes, en especial aquellos que necesitan un monitoreo constante para asegurar la eficiencia y producción del sistema, tal es el caso de los sistemas acuapónicos (Trinidad et al., 2018).

El desarrollo de herramientas tecnológicas que tienen como resultado la generación de información sobre el estado en que se encuentran los sistemas de producción agrícolas, obedece a la demanda que tienen los productores sobre la búsqueda de formas más precisas en la toma de decisiones para el acondicionamiento del sistema, la corrección de parámetros físicos-químicos y la

optimización de recursos que se dirijan a la producción del cultivo durante todo su ciclo de vida.

De acuerdo con Archbold et al. (2019) los sistemas agrícolas requieren sistemas tecnológicos que permitan procesar y transmitir información, con el objetivo de presentar al usuario de manera interactiva y remota el estado en que se encuentra un parámetro del sistema durante todo el proceso de producción. Una manera de abordar esta necesidad es a través de aplicaciones IoT, ya que este paradigma tecnológico integra una red de sensores y múltiples fuentes de información que aseguran cobertura, calidad y seguridad en la toma de datos de manera inalámbrica.

Como plantean Tovar, Solórzano, Badillo & Rodríguez (2019) la automatización agroalimentaria en procesos de producción y obtención de información es uno de los sectores donde se verá más aplicada la tecnología IoT en la próxima década. De la misma manera estos autores argumentan que la aplicación de tecnología IoT lograría aumentar la producción global de los sistemas agrícolas en un 70% generando un impacto positivo para satisfacer la demanda de alimentos en el año 2050, donde se estima una población de 9.5 billones de personas en el mundo.

Pérez, Mendoza & Suarez (2019) dan a conocer los resultados de búsqueda bibliográfica sobre el internet de las cosas aplicado en la agricultura en las bases de datos científicas: Scopus y Science Direct, obteniendo como resultado la información que se presenta en el Cuadro 1.

Cuadro 1. Resultados de criterios búsqueda en bases de datos científicas sobre internet de las cosas (Pérez et al., 2019).

Combinación de áreas del conocimiento	Rango de años búsqueda	Resultados de búsqueda
IoT + Sensor	2012-2018	3485
IoT + Agriculture	2012-2018	459
IoT + Agriculture + Sensor	2012-2018	279
IoT + Agriculture + Sensor + Control	2012-2018	110

El uso de redes de sensores y actuadores permiten monitorear parámetros físicos del sistema creando una interacción con el usuario. De acuerdo con Trinidad et al. (2018) el monitoreo en tiempo real de variables físicas de un sistema con ambiente controlado aplicando la tecnología IoT, presenta un desafío tecnológico donde se espera un crecimiento exponencial y un impacto directo en las actividades diarias de este sector.

2.4. Conclusiones

La revisión del estado del arte reflejó el impacto y la importancia que tiene la implementación de tecnología IoT que permitan monitorear y controlar parámetros físico-químicos (pH, temperatura, oxígeno disuelto, niveles de nitrógeno amoniacal, niveles de nitratos y niveles nitritos) en un sistema de producción acuapónico.

En la acuaponía los parámetros de pH, temperatura y oxígeno disuelto deben ser monitoreados diariamente para poderlos llevar a un rango de control óptimo que garanticen la relación simbiótica de sus principales elementos biológicos (plantas, peces y bacterias nitrificantes). Este manejo diario del sistema es una de las principales desventajas que presenta, ya que descuidar estos parámetros pueden ocasionar grandes daños en el sistema llegando a perder la producción de peces y/o plantas. La aplicación del paradigma tecnológico IoT en la

agricultura permite desarrollar herramientas de control y monitoreo de los sistemas acuapónicos que mantengan una óptima calidad de agua para los elementos biológicos presentes. Esta herramienta puede llegar a reducir considerablemente los costos de producción y el impacto ambiental optimizando los recursos naturales presentes y reduciendo el uso de fertilizantes en el sistema.

2.5. Literatura citada

Aguirre B., E. A., Monje C., J. J., Sierra S., C. & Vega C., D. A. (2018). Desarrollo de un sistema de monitoreo para acuaponía en hogares basado en IOT. *Inventum*, 13, 15-21, doi: 10.26620/uniminuto.inventum.13.24.2018.15-21.

Archbold, G. W., Beltrán, A. H., Ruiz, F., Narducci, M. S., Méndez, D., Trujillo, L. C., Parra, C., Carrillo & Mouazen, A. M. (2019). pH measurement IoT system for precision agriculture applications. *IEEE Latin America Transactions*, 17(5), 823-832.

Caló P. (2011). *Introducción a la acuaponía*. Centro Nacional de Desarrollo acuícola (CENADAC). Recuperado de: http://www.minagri.gob.ar/site/pesca/acuicultura/06_Publicaciones/_archivos/130423_Introducci%C3%B3n%20a%20la%20ACUAPONIA.pdf

Gutiérrez L., E., Montiel A., J.E., Carreto A., C., & Menchaca G., F.R. (2019). Propuesta de sistema de gestión inteligente basado en IoT para hidroponía. *Computing Science*, 10, 219-233.

López, G. (2017). *Absorción del fósforo en diferente pH en acuaponía con jitomate-tilapia* (Tesis de maestría). Colegio de Postgraduados, México.

- Mena, A. (2018). *Sistema iot para el monitoreo y control de fuentes de luz artificial aplicado a la agricultura de precisión* (Tesis de pregrado). Universidad Distrital Francisco José de Caldas, Bogotá, D.C.. Recuperada de <http://repository.udistrital.edu.co/handle/11349/7991>
- Mercado, A. & Gonzales, A. (2018). *Propuesta para el diseño de una red de sensores implementando el internet de las cosas aplicado en la agricultura de precisión* (Tesis de pregrado). Instituto Politécnico Nacional, México.
- Pérez, J. (2018). *Nutritional composition of lettuce (Lactuca sativa L.) cultivated in aquaponic, hydroponic and soil* (Tesis de pregrado). Universidad Autónoma Chapingo, México.
- Pérez, M. R., Mendoza A., M. & Suarez J., M. (2019). Paradigma IoT: desde su conceptualización hacia su aplicación en la agricultura. *Espacios*, 40(8), 6.
- Pineda, J., Miranda, I., Rodriguez, J., Ramirez, J., Perez, E., Garcia, I. & Morales, J. (2017). Nutritional balance in aquaponic lettuce production. *Acta Hortic.*, 1170, 1093-1100, doi: 10.17660/ActaHortic.2017.1170.141.
- Resh, H.M. (2012). *Hydroponic Food Production: A Definitive Guidebook for the Advanced Home Gardener and the Commercial Hydroponic Grower* (7thed.). Northwestern, Estados Unidos: Editorial CRC Press,
- Rodríguez G., H., Rubio C., S. G., García U., M., Montoya M., M. & Magullón B., F.J. (2015). Análisis técnico de la producción de tilapia (*Oreochromis niloticus*) y lechuga (*Lactuca sativa*) en dos sistemas de acuaponia. *Agroproductividad*, 8, 15-19.
- Roosta, H. & Hamidpour, M. (2013). Mineral Nitrient content of tomato plants in aquaponic and hydroponic systems: Effect of foliar application of some macro-and micro-nutrients. *Journal of Plant Nutrition*, 36 (13), 2070-2080, doi: 10.1080/01904167.2013.821707.
- Rouse, M. (2017). *Definición de Internet de las Cosas*. Recuperado de: https://searchdatacenter.techtarget.com/es/definicion/Internet-de-las-cosas-IoT_consultada_en_Enero_2020.
- Tovar S., J., P., Solórzano S., J., S., Badillo R., A. & Rodríguez C., G., O. (2019). Internet de las cosas aplicado a la agricultura: estado actual. *Lámpsakos*, 22, 86-105, doi: 10.21501/21454086.3253.
- Trinidad, G., Meis, V., López, C., Italo, J., Ríos, C., Saldaña, G., Hernández, J., & Molina, M. (2018). Wireless Sensor Network for Monitoring Physical Variables Applied to Green Technology (IoT Green Technology). *European Journal of Electrical and Computer, Engineering*, 2, 1-7, doi: 10.24018/ejece.2018.2.2.15.

- Wongkiewa, S., Hub, Z., Chandranc, K., Woo, L. & J., Kumar, S. (2017). Nitrogen transformations in aquaponic systems: A review. *Aquacultural Engineering*, 76, 9–19, doi: 10.1016/j.aquaeng.2017.01.004.
- Zarzoza, S. (2018). *Phosphorus cycling in an aquaponic System*. (Tesis Profesional, Universidad Autónoma Chapingo, México).
- Zou, Y., Hu, Z., Zhang, J., Xie, H., Guimbaud, C. & Fang, Y. (2016). Effects of pH on nitrogen transformations in media-based aquaponics, *Bioresour Technol*, 210, 81–87, doi: 10.1016/j.biortech.2015.12.079.

3. DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO Y CONTROL DE PH CON TECNOLOGÍA DE INTERNET DE LAS COSAS PARA ACUAPONIA

Resumen

La importancia del monitoreo de parámetros físicos en la acuaponia como: temperatura, pH, oxígeno disuelto y nivel de agua radica en la calidad de agua que tiene el sistema para mantener una interacción simbiótica de sus elementos biológicos (plantas, peces y bacterias) para regular su metabolismo. El objetivo de este trabajo es desarrollar una red de comunicación con para el monitoreo y control en tiempo real de la variable física de pH, de manera remota y de fácil acceso. Se realizó red de intercomunicación entre el sensor de pH (PHE-1304) a una aplicación móvil utilizando las bases del paradigma tecnológico de Internet de las Cosas. Para leer el sensor se utilizó una placa Raspberry la cual se programó en lenguaje Python y se conectó a una nube de datos llamada Firebase la cual de manera simultánea se vincula con una aplicación móvil a través del IDE de Android Studio. Se obtuvo una aplicación Android con una interfaz en la cual el usuario puede visualizar el valor de pH del sistema acuapónico en tiempo real, además de acceder a un archivo en formato csv con las mediciones tomadas por el sensor. El monitoreo remoto en tiempo real permite a los productores mantener las condiciones necesarias del sistema para su producción teniendo bases para aplicar acciones que mantengan los parámetros físicos en rango óptimos gracias a la información enviada por los sensores del sistema.

Palabras Clave: hidroponía, acuacultura, Raspberry, sensor, firebase.

3.1. Introducción

La acuaponía puede desarrollarse en diferentes métodos de producción, dependiendo de la extensión del terreno, el nivel de tecnología y capacitación con la que cuente el productor. Por lo tanto, un sistema de control y monitoreo del invernadero basado en tecnología IoT debe ser flexible, configurable y de bajo costo, para que pueda ser redituable su integración al sistema.

Un sistema de internet de las cosas se basa en la comunicación de sus principales elementos (Estación de monitoreo, Gateway y la nube de datos) para que el usuario pueda acceder a los datos monitoreados de manera remota y pueda tener un control sobre estos.

May, Quijano & Fernández (2015) comentan que: “La idea general del concepto IoT es que todos los objetos y dispositivos cuenten con un identificador que pueda conectarse a internet para ser reconocido por otros objetos o máquinas”. Por lo cual un proyecto IoT tiene una gran utilidad para controlar de manera remota aparatos eléctricos y al mismo tiempo llevar a cabo un control de diferentes procesos.

La instalación de un sistema de control y monitoreo IoT en la acuaponía mejora la productividad del sistema, logrando satisfacer la demanda de producción al mismo tiempo que haría más eficiente el uso de los recursos hídricos. El control y monitoreo permanente de parámetros del sistema que intervienen en el desarrollo del cultivo hacen posible asignar de manera adecuada los recursos disponibles para la producción, garantizando no solo el control de estos, si no de igual manera mejorando el volumen de producción la calidad (Guerrero, Estrada & Medina, 2017). Ruano (2017) menciona que uno de los objetivos que tiene el avance tecnológico en los invernaderos es reducir el esfuerzo y el tiempo necesario para su cuidado.

Este trabajo tiene como objetivo desarrollar una red de comunicación de fácil acceso y bajo costo con tecnología de IoT para el monitoreo y control de pH de un sistema acuapónico. Se propone un sistema de control y monitoreo utilizando Python como lenguaje de programación del Gateway, Firebase como base de datos, y Android estudio como software de monitoreo de datos remoto los cuáles en sus versiones más básicas son de libre acceso y gratuitos. Lo cual ayuda a reducir considerablemente costos del desarrollo del sistema IoT.

3.2. Materiales y Métodos

3.2.1. Estación de monitoreo

La estación de monitoreo que se utilizó para las pruebas se encuentra ubicada en el Invernadero de Acuaponia del Campo Experimental San Ignacio de la Universidad Autónoma Chapingo. Este invernadero cuenta con sistemas acuapónicos de Relleno de medios con las siguientes características: la cama de crecimiento es una tina de polietileno rectangular de 240 litros de capacidad (un metro de largo, 60 centímetros de ancho y 40 centímetros de profundidad) la cual tiene tezontle rojo, el cual funciona como sustrato, así como biofiltro del sistema; cada cama de crecimiento cuenta con un estanque cilíndrico de polietileno de 200 litros de capacidad total (Pineda et al., 2018). El sistema de recirculación de agua se da por medio tubos pvc perforados el cual se acciona con bombas sumergibles individuales. El oxígeno es llevado a ellos a través de una tubería pvc subterránea accionada por una bomba de aire común para todas las camas de crecimiento (Figura 3).



Figura 3. Cama de crecimiento y estanque del sistema acuapónico.

3.2.2. Gateway

El Gateway del sistema está conformado por una Raspberry Pi 4 modelo B+ a través de conexión WIFI. Este recibe, interpreta y envía los datos de los sensores instalados en la estación de monitoreo mediante un módulo programado en Python.

Python es un lenguaje de programación que actualmente se desarrolla como un proyecto de código abierto administrado por Python Software Foundation. Python puede desarrollarse en un ambiente de programación orientada a objetos.

El módulo IoT de Python cumple con la función de monitoreo, control y comunicación del Sistema. Este se desarrolla en un programa que ejecuta unos subprocesos de manera independiente de una clase, con el propósito de que las funciones se realicen de manera independiente y que sus estados de funcionamiento no afecten las demás funciones.

Para el desarrollo del software se definieron clases, objetos y los métodos necesarios. Se definió al objeto general como el invernadero y una clase como la red IoT del sistema, esta es la que tendrá todos los atributos o métodos del sistema, es decir en esta clase estarán todos los actuadores, sensores y la estructura de la base de datos en Firebase. Mediante librerías y comandos específicos se desarrolló la estructura de la base de datos, definiendo cada elemento de la estructura como atributo de nuestro sistema, esto con la finalidad de poder tener un control independiente de cada variable y así lograr el control de varios actuadores de manera remota y simultánea.

Para lograr que el software pueda leer el estado de varios actuadores y recibir señales de la base de datos de manera simultánea, sin que el programa produzca errores de ejecución o se sature por la cantidad de comandos ejecutados, cada atributo, es decir cada sensor o actuador, de la clase, se ejecutan por subprocesos por métodos de hilos. De esta manera cuando sea necesario saber o enviar el estado de un sensor o actuador solo se ejecutan las funciones que son necesarias, evitando el reproceso de algunas funciones (Figura 4).

El Gateway Coordinador envía los datos de la estación de monitoreo obtenido por los sensores a la nube de Firebase para que estos puedan ser enviados a la interfaz gráfica Android y así pueda ser monitoreado por el usuario.

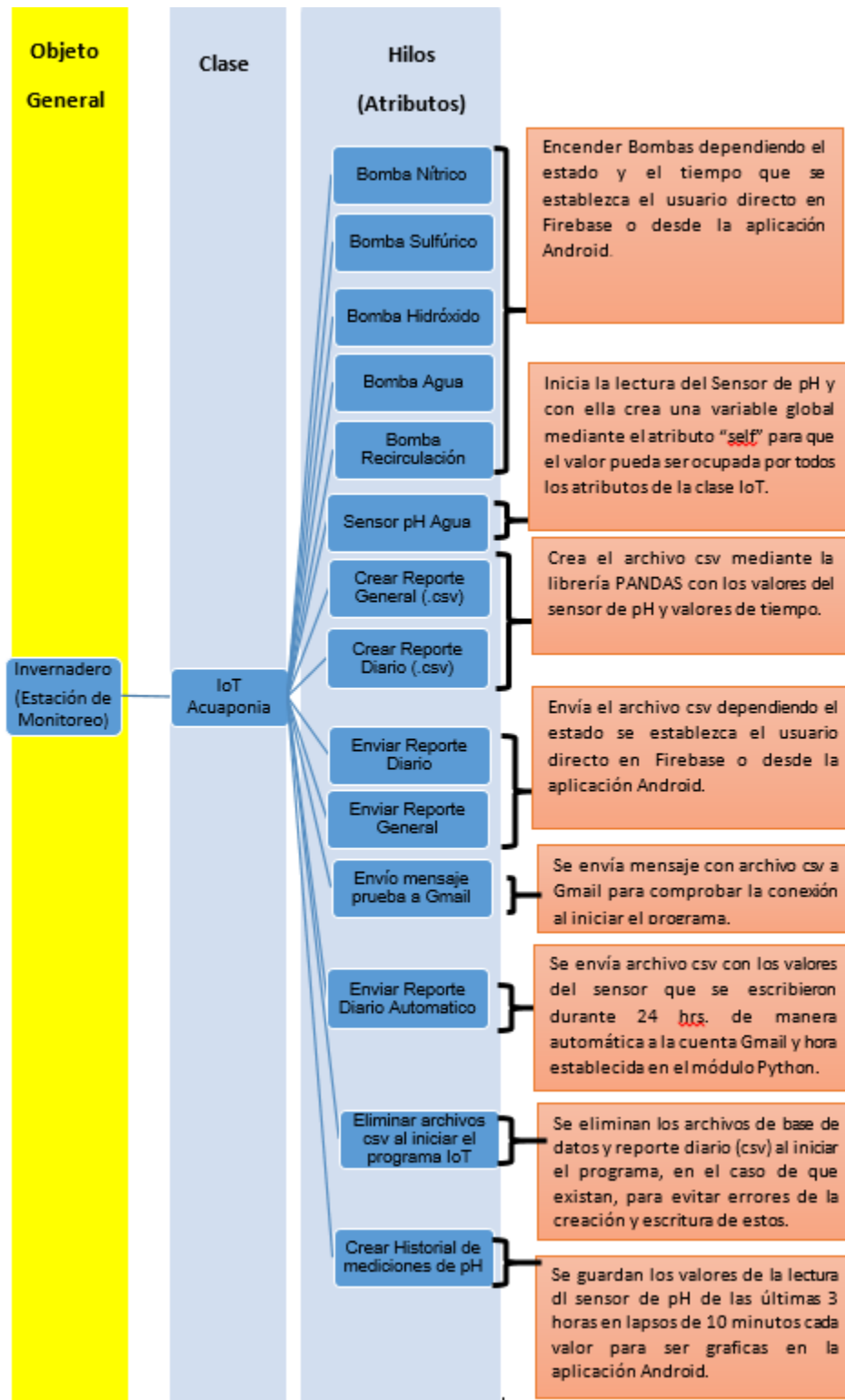


Figura 4. Esquema de funcionamiento de los hilos del código de programación.

3.2.3. Raspberry Pi

Raspberry es un miniordenador de bajo costo y alto desempeño que permite al usuario interactuar con el sistema operativo instalado (Figura 5). Esta utiliza lenguajes de programación como Python, C++ y Java. Estas tarjetas cuentan con todos los componentes para correr un sistema operativo (Martín, 2014).

La tarjeta utilizada en este proyecto es la Raspberry pi 4 model B+. Esta tarjeta cuenta con 2 Gigabytes de memoria interna, 2 salidas micro HDMI para conectar monitores, 2 puertos USB 3.0 y dos puertos USB 2.0, conexión Ethernet, y una ranura para memoria micro sd, conexión wifi y bluetooth 5.0 y una cabecera GPIO de 40 pines.

Martín (2014) menciona que una tarjeta Raspberry pi es recomendable para desarrollar proyectos que requieran interfaz gráfica y conexión a internet.

Una tarjeta Raspberry puede ser configurada para que funcione como servidor Web y que al mismo tiempo pueda enviar y efectuar instrucciones operacionales a través de sus pines GPIO (May et al., 2015).

En este trabajo se configuro la tarjeta como Gateway sistema, es decir cumplirá las funciones de servidor web y controlador del sistema de monitoreo. El sistema operativo que se le instaló fue el oficial ofrecido por Raspberry llamado Raspbian, que es una distribución de Linux, el cual ofrece un entorno de escritorio Linux (Ruano, 2017). Este sistema tiene como ventaja que cuenta con varias librerías y foros para diferentes lenguajes de programación, como lo es Python.



Figura 5. Tarjeta Raspberry Pi 4 model B.

3.2.4. Sensor de pH Omega

El Omega PHE-1304 (PHE 1304; Omega Engineering Inc., Stamford, CT, EE. UU.) es un sensor analógico de electrodo preamplificador con un rango de medición de pH de 0 a 14, una resolución de 0,01 pH, un rango de temperatura de trabajo de 0 a 80 °C y una precisión de $\pm 0,02$ pH (Figura 6). La calibración de este sensor se realizó siguiendo las instrucciones del fabricante, la cual consiste en forzar una lectura de voltaje de 2.5 del electrodo mediante un potenciómetro; posterior a esto se miden con el electrodo los voltajes de diferentes soluciones buffer conocidas para poder realizar una regresión lineal entre el valor de pH y la lectura de voltaje de la solución buffer conocida. Con la ecuación obtenida se puede conocer el valor de pH de cualquier solución.



Figura 6. Electrodo Omega PHE-1304.

Buck et al. (2002) definen al pH como la concentración de iones de Hidrogeno en una solución. De esta manera se puede cuantificar que tan ácida o alcalina esta la solución. Oreggioni (2006) define al pH mediante la siguiente ecuación:

$$pH = -\log_{10} f_{H^+}[H^+]$$

Donde: f_{H^+} es el coeficiente de actividad, $[H^+]$ es la concentración de iones de hidrogeno en moles·L⁻¹ y el producto de ambos es la actividad del ion H⁺.

Pero debido a que en la mayoría de las aplicaciones la cantidad de agua que se produce o se consume es muy pequeña comparada con la cantidad de agua total que está presente en la solución: f_{H^+} se considera como uno, quedando simplificada la ecuación de la siguiente manera:

$$pH = -\log_{10}[H^+]$$

El funcionamiento de trabajo de un electrodo de pH se basa en registrar una diferencia de potencial entre una solución de referencia base con un pH fijo y la solución que se desea medir (Figura 7). El electrodo tiene un bulbo con una membrana sensible selectiva a protones de Hidrogeno los cuales toma de la solución de la que se desea conocer, lo que genera una señal eléctrica. Esta

medida de cargas la compara con la carga que tiene la pared de la membrana con la solución de referencia (Oreggioni, 2006). Esta diferencia de cargas es mandada a un módulo que las traduce a un valor de pH.

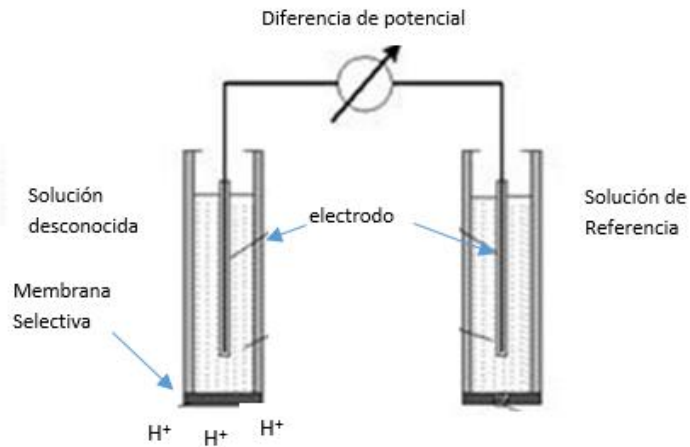


Figura 7. Representación del funcionamiento del bulbo de un electrodo de pH.

Para obtener las lecturas del Electrodo Omega PHE-1304 es necesario de un módulo análogo de lectura y calibración PH-4502 C con conexión BNC, alimentación a 5 volts, rango de medición de pH de 0 a 14, tiempo de respuesta < 5 s y una salida de voltaje analógico (Figura 8).

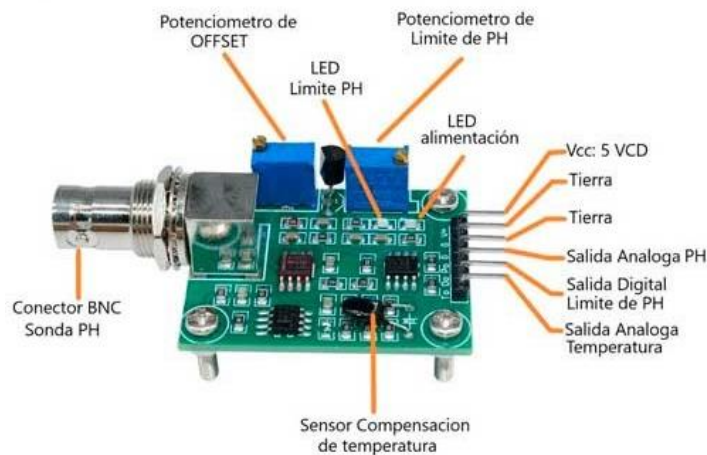


Figura 8. Módulo PH-4502C.

3.2.5. Convertidor Analógico a Digital ADC ADS1115.

La placa Raspberry Pi 4 B+ cuenta con pines que son capaces de leer señales únicamente digitales por lo cual para poder utilizar sensores Analógicos es necesario conectar un convertidor analógico-digital. El ADS1115 es un conversor analógico a digital con ganancia de 16 bits totales y una resolución de 15 bits. Este conversor se conecta a la placa Raspberry mediante el protocolo i2c capaz de leer 4 señales analógicas simultáneamente (Figura 9).

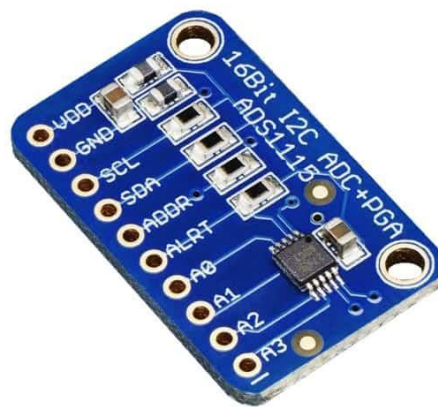
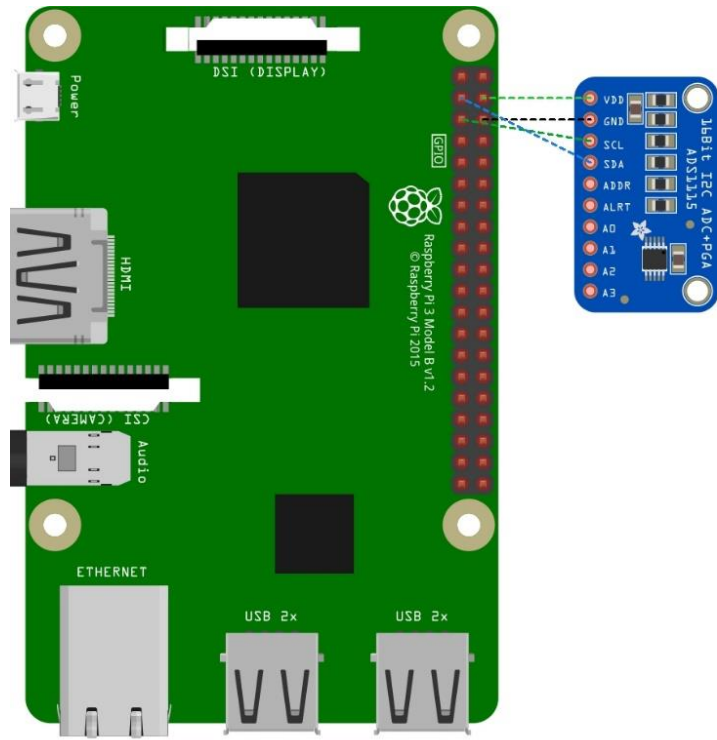


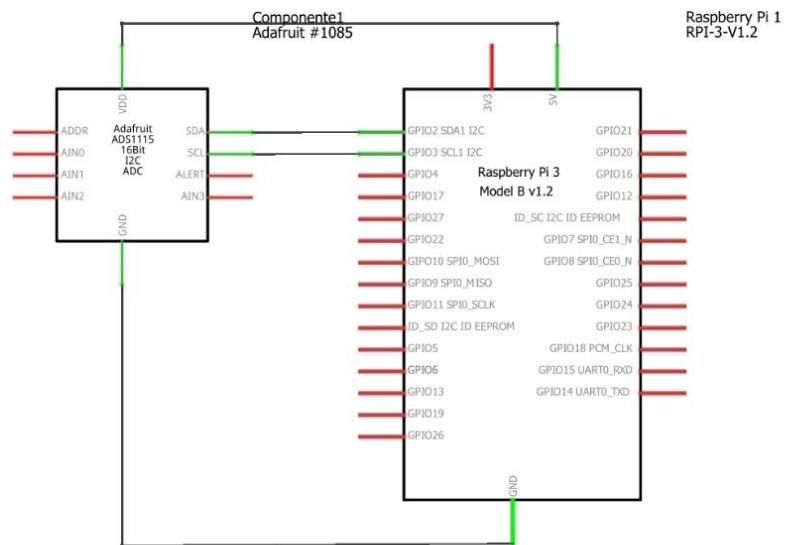
Figura 9. Convertidor analógico a Digital ADS1115

Un protocolo I2C es un puerto de comunicación serial que permite tener a varios maestros controlando uno o múltiples esclavos utilizando solamente 2 líneas de comunicación y un común en masa de los microcontroladores (Carletti, 2007) (Figura 10). Es decir, mediante los hilos de comunicación (SDA y SCL) se puedan transmitir o recibir señales de dos o más sensores o actuadores mediante bits en forma coordinada (Pimentel, Yate & Porras, 2015) (Figura 11). La línea de comunicación SDA (System Data) es la vía de comunican entre el maestro y los esclavos, es decir es la línea por donde se mueven los datos entre los dispositivos conectados y la línea SCL (System Clock) es la vía por donde viaja la señal de reloj que sincroniza el sistema (Carletti, 2007).



fritzing

Figura 10. Conexión de SDA y SCL de Raspberry y ADS115.



fritzing

Figura 11. Esquema de adaptación de protocolo de comunicación I2C por medio de las líneas SDA y SCL.

3.2.6. Módulo de relevadores para bombas

Es un dispositivo electrónico que permite alternar los caminos por donde seguirá una corriente eléctrica de un circuito. Funcionan como un interruptor que mediante un electroimán al ser activado permite el flujo de corriente, y viceversa, si el electroimán se desactiva corta el flujo de corriente (Figura 12). La aplicación de estos dispositivos electrónicos es para ceder el flujo de corrientes relativamente altas al activar el electroimán con corrientes relativamente pequeñas. El módulo de Relevadores que se utiliza en este proyecto permite el flujo de corrientes de 12 volts que accionan bombas (Figura 13) de diagrama al enviar señales de 5 volts mediante los GPIO de la placa Raspberry.

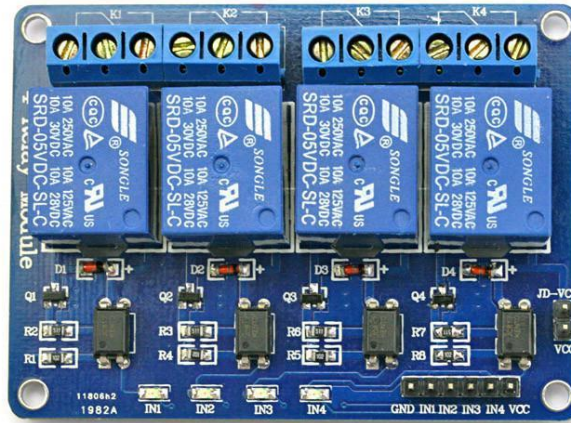


Figura 12. Módulo de 4 relevadores



Figura 13. Bomba de diafragma de 12 volts

3.2.7. Nube de Datos Firebase

En un proyecto IoT el almacenamiento y visualización de datos puede darse comúnmente por dos vías, una es a través de aplicaciones y almacenamiento de dispositivos electrónicos y la segunda a través de un servidor que almacene los datos en una nube (García, 2017).

Firestore es una plataforma de datos creada por Google, la cual permite el desarrollo de aplicaciones móviles para el monitoreo y obtención de datos de manera inalámbrica. Esta plataforma tiene integrada una nube de datos y herramientas que pueden utilizarse para la sincronización de proyectos en los que se requiere una lectura de datos en tiempo real. Sus características principales son: Funciones Analíticas, las cuales proveen una solución gratuita para poder monitorear hasta 500 eventos en un mismo panel. Funciones de desarrollo, con las cuales se puede almacenar, testear, configurar remotamente, mensajería en la nube o autenticación, etc.

Se utilizó esta nube de datos por su comunicación eficaz y sencilla con Python y Android Studio, gracias a las librerías existentes de desarrolladores de estas plataformas.

De acuerdo con Zamora (2019), Firestore es una base de datos no estructurada o relacional que almacena datos en formato JSON (JavaScript Object Notation), esta ofrece flexibilidad con cualquier fase de desarrollo en la que se desee agregar campos de datos y permitiendo acceder a ellos desde cualquier punto con conexión a internet con la posibilidad acceder a ellos sin la necesidad de un hardware destinado especialmente a esta función. Mercado & Gonzales (2018) se refieren a JSON como un formato ideal de intercambio de datos con un entorno de lectura y escritura sencillo para el usuario.

El proyecto que se desarrolla utiliza la versión gratuita de Firebase ya que esta ofrece la extensión Realtime Database que se utiliza para guardar, enviar y recibir datos en tiempo real y es el Gateway el que analiza y ejecuta las demás funciones de control, almacenamiento, envío y monitoreo de datos. Es decir, la base nube de datos principalmente sirve como puente de comunicación entre el Gateway y la aplicación móvil que utiliza el usuario para la lectura, pero esta no almacena datos.

3.2.8. Estructura de la nube de datos Firebase mediante Python

La Realdatabase de la nube de datos de Firebase se estructuró directamente en el módulo Python del Gateway coordinador. Para crear la estructura se utilizó el mecanismo de programación “herencia” que ofrece Python; en donde cada celda de Firebase creada se vincula con una referencia en Python, permitiendo la importación y exportación de datos entre la celda de Firebase y la referencia en Python.

Para crear la referencia en el módulo de Python, se crean variables globales de referencia con el nombre de todas las celdas que tendrá Firebase; posteriormente, dentro de la clase en el método principal (“_init_”) se crean variables con el atributo “self”, cada variable está vinculada a una variable de referencia global y mediante el mecanismo de herencia damos la estructura de ramificación de la base de datos. En la Figura 14 se muestra un ejemplo en el que se crea una celda para el parámetro de lectura del pH; en esta se puede apreciar que la celda principal “home” tiene una herencia llamada “parámetros” y esta última a su vez una herencia llamada “pH”. Utilizando el anterior mecanismo se estructura toda la base de datos creando todas las celdas necesarias con sus respectivas herencias y variables de referencia para que se dé el intercambio de datos entre Firebase y Python.


```

1  import firebase_admin
2  from firebase_admin import credentials
3  from firebase_admin import db
4  REF_HOME = 'home'
5  REF_PARAMETROS = 'Parametros'
6  REF_pH = 'pH'
7  class IOT:
8      def __init__(self):
9          cred = credentials.Certificate(PAHT_CRED)
10         firebase_admin.initialize_app(cred, {
11             'databaseURL': URL_DB
12         })
13         self.refHome = db.reference(REF_HOME)
14         self.estructuraInicialDB()
15         self.refParametros = self.refHome.child(REF_PARAMETROS)
16         self.refpH = self.refParametros.child(REF_pH)
17     def estructuraInicialDB(self): #Estructura de como se mostrara en firebase
18         self.refHome.set({
19             'Parametros':{
20                 'pH':7,
21             },
22         })
23

```

Figura 14. Ejemplo de estructura de base de datos en Firebase creada mediante Python.

3.2.9. Creación archivos csv mediante la librería Pandas

El módulo Python del Gateway tiene las herramientas y librerías necesarias para crear automáticamente archivos en formato csv. Este crea dos archivos para enviarlos como reportes de monitoreo: uno con las lecturas de los sensores tomadas cada minuto durante periodos de 24 horas llamado “Reporte_diario.csv” y el segundo “Base_de_datos.csv” con las lecturas de los sensores tomadas cada 10 minutos desde el arranque del programa hasta su paro.

Al arrancar el módulo de Gateway uno de los hilos o subprocessos programados se encarga de eliminar los archivos en formato csv, mediante la librería “Remove”, con el mismo nombre de los archivos que se crearán, es decir: “Reporte_diario.csv” y “Base_de_datos.csv” como se muestra en la Figura 15. Si

existe un archivo con el mismo nombre de un archivo que se pretenda crear, Python no marca error en el hilo, pero puede ocasionar que este no tenga la capacidad para sobrescribir un archivo ya existente.

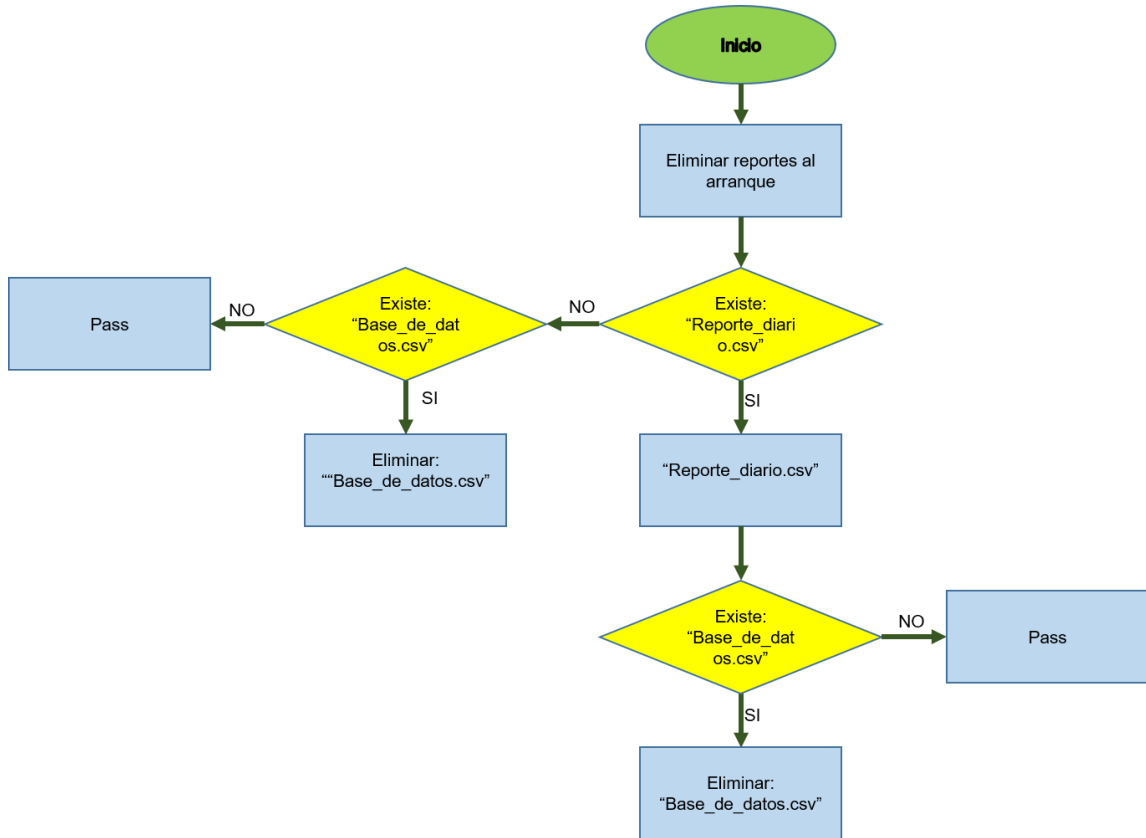


Figura 15. Método de hilo para la eliminacion de archivos al arranque del programa.

Pandas es una librería de Python que tiene como objetivo el análisis estadístico de datos. Esta librería se utilizó para que el módulo Python del Gateway genere un Dataframe con los datos de la lectura los sensores y con la librería “datetime” se obtiene la fecha y hora actual de las lecturas, los cual queda registrados en el Dataframe.

El módulo Python cuenta con dos hilos, cada uno encargado de crear uno los dos archivos csv (“Reporte_diario” y “Base_de_datos”). En la Figura 16 se muestra el diagrama de flujo del método de hilo encargado de crear el archivo “Reporte_diario.csv”. Este hilo consta de dos ciclos “While”. El primer ciclo “While” infinito se crea utilizando una condicional “True” el cual se repite cada 24 horas,

este ciclo es el encargado de crear el Dataframe con las columnas y los datos que tendrá nuestro reporte de monitoreo. El segundo ciclo "While" tiene condición "if", o bien expresada en lenguaje Python como un "elif", este repite el número de veces necesarios para completar las 24 horas del primer ciclo; esto se logra utilizando una operación lógica "or" en la cual se establece la hora y el minuto en que se reinicia el primer ciclo "While". De igual manera el segundo ciclo "While" está encargado de escribir los renglones de las columnas creadas en el Dataframe con los datos de la lectura de los sensores, hora y fecha en que son tomados, al finalizar esta acción convierte el Dataframe en archivo csv con el nombre "Reporte_diario". Cuando la condición "or" se cumple se ejecuta un tiempo de espera de 60 segundos, tiempo esperado para que el archivo creado sea adjuntado y enviado por Gmail automáticamente por el hilo programado especialmente para este proceso, al terminar el tiempo de espera se elimina el archivo creado y se inicia un nuevo ciclo en el que se crea un nuevo un Dataframe reemplazando el anterior con nuevos valores.

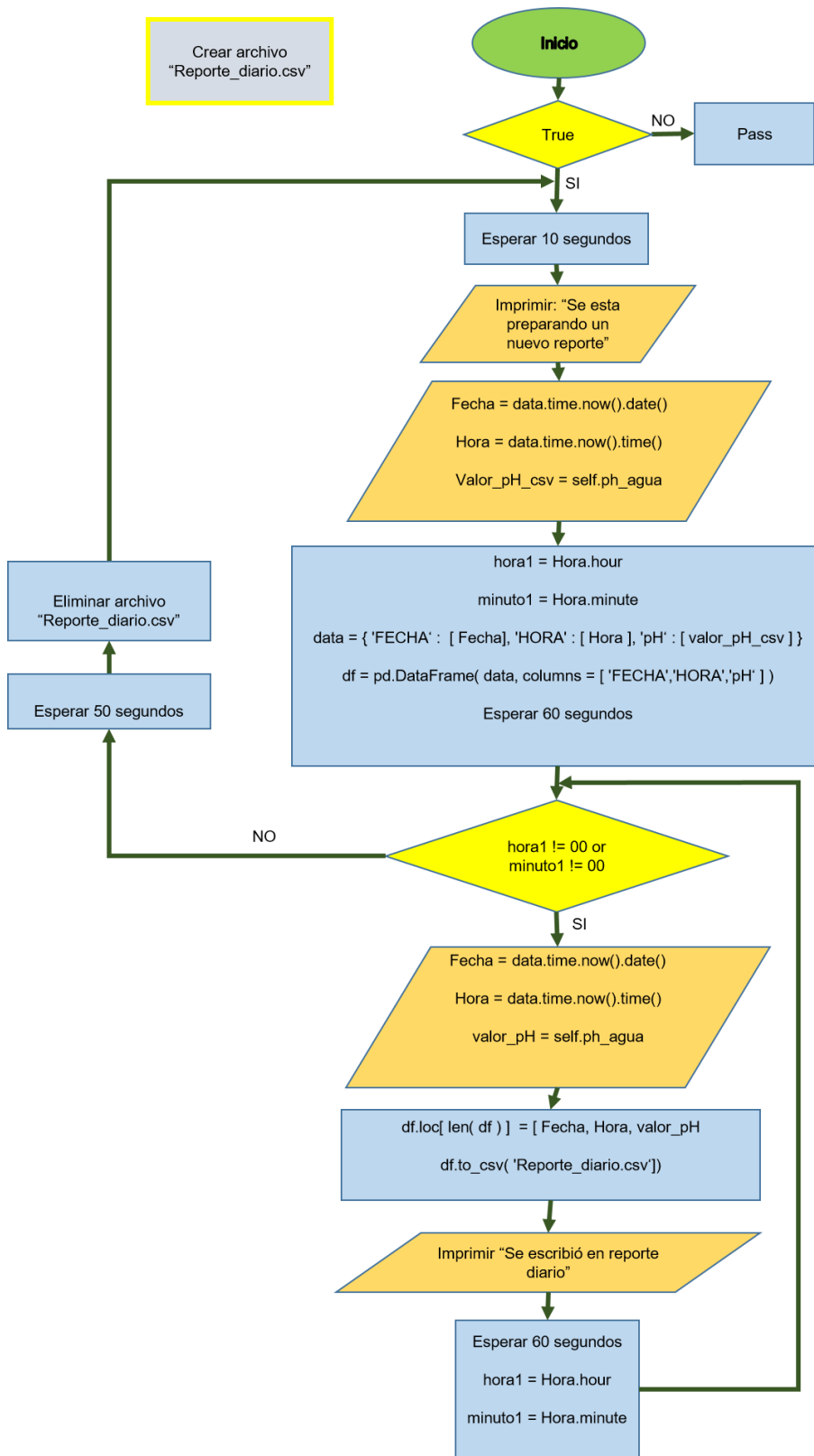


Figura 16. Flujograma del hilo de Python para la creación y escritura del archivo "Reporte_diario.csv".

En el caso del archivo “Base_de_datos.csv” sigue un flujo de operaciones similar al de “Reporte_diario.csv” a excepción de que esta cuenta solo con un ciclo While el cual se repite cada 10 minutos como se muestra en la Figura 17.

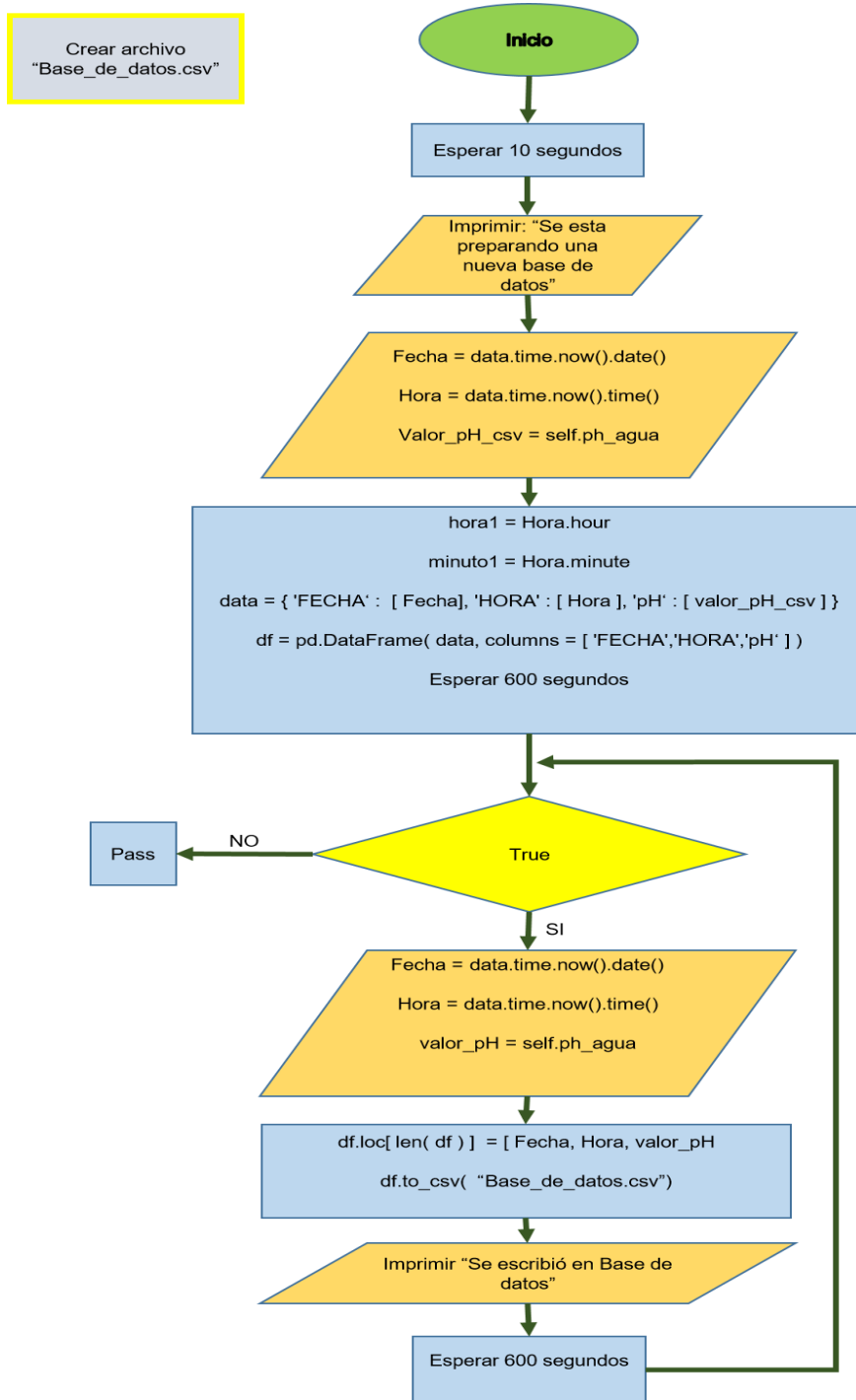


Figura 17. Flujograma del hilo de Python para la creación y escritura del archivo "Base_de_datos.csv".

3.2.10. Envío de reportes de monitoreo mediante Gmail

El envío de los archivos csv generados por el Gateway coordinador son enviados a través de un correo electrónico de Gmail que es vinculado al módulo Python mediante las librerías: MIMEMultipart, MIMEBase y smtplib. Para realizar la vinculación entre Gmail y Python es necesario activar la opción “Permitir el acceso de apps menos seguras” directamente de la página de configuraciones de cuenta de Gmail (Figura18).

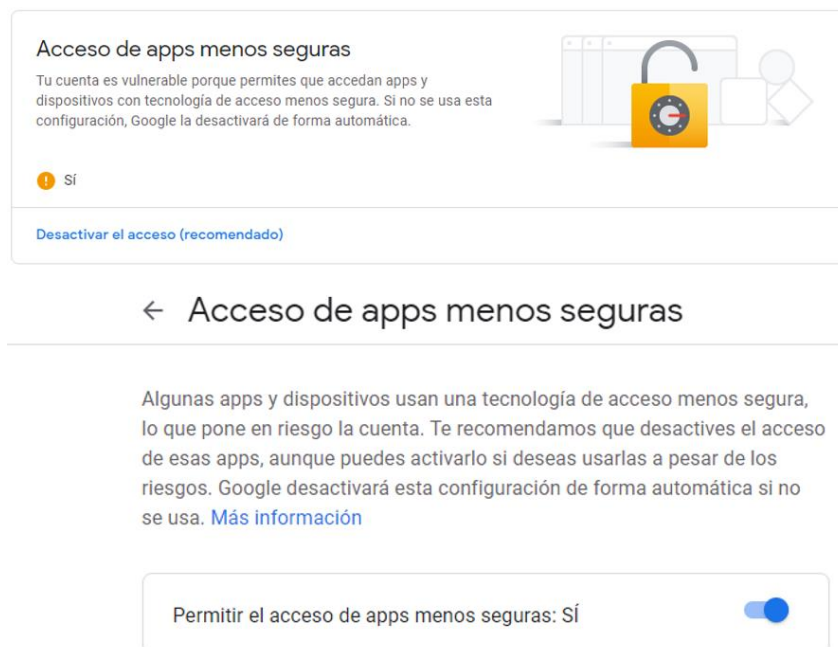


Figura 18. Configuración Gmail para permitir el acceso de aplicaciones.

Para comprobar el estado la vinculación exitosa del Gateway con Gmail, al arrancar el módulo Python se ejecuta un hilo el cual tiene las operaciones necesarias para enviar un archivo csv, previamente guardado en la memoria del sistema operativo de la tarjeta raspberry (Figura 19). Si la vinculación es exitosa en el monitor del Gateway aparece un mensaje de conexión exitosa, si falla algo en el proceso de vinculación aparece el error en el método de hilo.

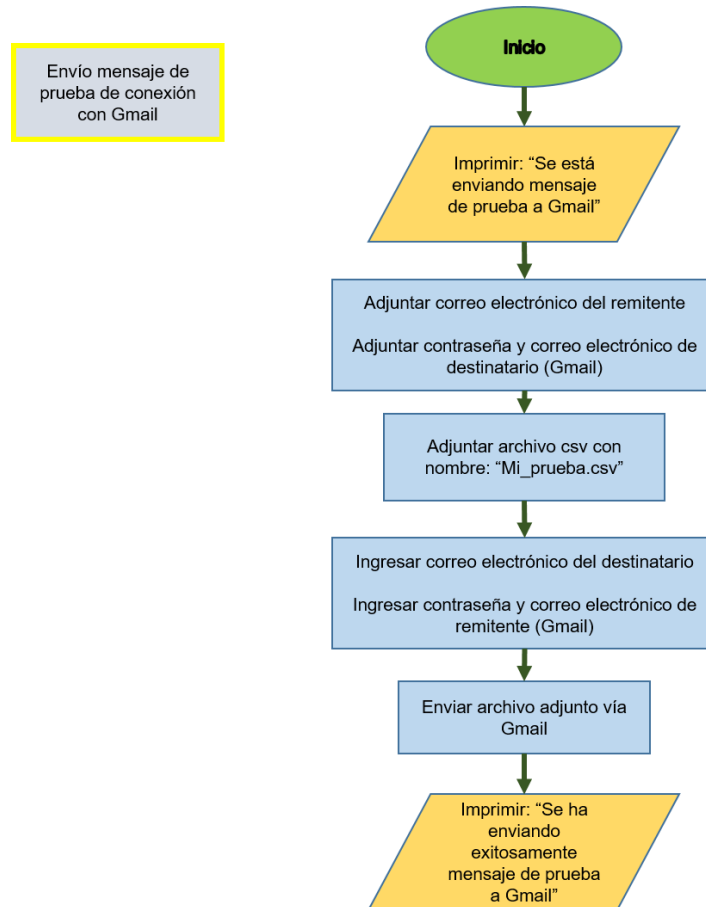


Figura 19. Diagrama de flujo del método de hilo Python para enviar mensaje de prueba a través de Gmail.

Los archivos csv creados como reportes de monitoreo se envían a través de Gmail mediante dos métodos de hilo: El primer método de hilo consiste en enviar de manera automática programando una fecha y hora en módulo Python; el segundo método de hilo consiste en solicitarlo directamente desde la nube de datos de Firebase.

Para el envío automático en el método de hilo se creó un ciclo "While" infinito en el cual se importa la hora y el minuto actual, los cuales se actualizan en cada repetición que tenga el ciclo, dentro del mismo ciclo se encuentra una condicional "if", la cual mediante el operador lógico "and" pregunta si la hora y el minuto actual es igual al establecido por el usuario; si la condición es verdadera el ciclo realiza las operaciones necesarias para enviar el archivo csv seleccionado en el método de hilo y termina con una orden de espera de 61 segundos para evitar

que se vuelva a reenviar el reporte; en caso de que la condición sea falsa se vuelve a iniciar el ciclo “While” (Figura 20). Este método se repite en dos hilos, cada uno para enviar uno de los archivos csv creados por Pandas, a diferencia de que el hilo que envía el archivo llamado “Base_de_datos.csv” la condicional está sujeta a dos operadores lógicos “and” para poder ingresar el día del mes, la hora y el minuto en que se desee enviar el archivo.

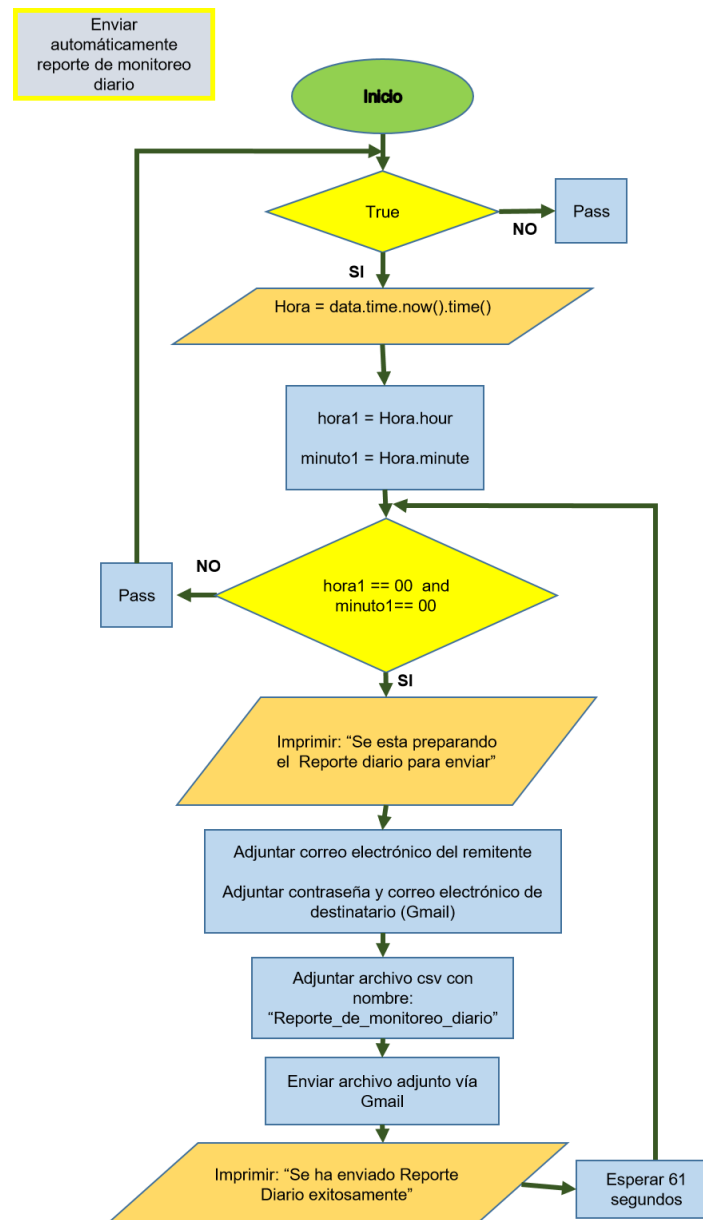


Figura 20. Diagrama de flujo del método de hilo Python para el envío automático vía Gmail del reporte de monitoreo diario en formato csv

Para obtener los archivos csv a través de la base de datos de Firebase se crean dos métodos de hilos cada uno vinculado a una celda de la nube de datos Firebase. Una celda llamada “General” mediante valores booleanos solicita el envío del archivo “Base_de_datos.csv” y la segunda celda llamada “Diario” mediante valores booleanos solicita el archivo “Reporte_diario.csv”.

La lógica de programación de estos hilos se basa en la comparación de valores que tiene la celda de Firebase (Figura 21). Para ello el método de hilo crea un vector de datos ($X []$) junto con una variable con valor cero ($i = 0$); después se importa el valor booleano actual que tenga la celda de la nube de datos Firebase vinculada con el archivo que queremos llamar (“General” o “Diario”); se guarda este valor en el vector que acabamos de crear con el atributo de Python “.append” el cual por lógica del programa toma el valor $X [0]$; a partir de este punto el hilo inicia un ciclo “While” infinito; después de iniciar el ciclo se vuelve a importar el valor booleano actual de la celda de Firebase y se guarda como un nuevo valor del vector el cual es $X [1]$; mediante una condicional preguntamos si el valor si el valor del último elemento ($X [-1]$) es diferente al valor del elemento “i” del vector ($X [0]$), si esto es falso elimina el primer elemento del vector y vuelve al inicio del ciclo infinito; esto trae como consecuencia que el valor del vector anterior $X [1]$ tome la posición del elemento eliminado $X [0]$ y por consiguiente el nuevo valor que se importa toma el valor $X [1]$ de esta forma siempre existe un vector con dos elementos para comparar; en caso de que la condicional sea cierta se ejecuta un proceso en el que se envía el último valor booleano importado de la nube de datos Firebase a una función específica y terminando su proceso eliminando el primer elemento del vector para volver al inicio del ciclo infinito.

La función específica que ejecuta cuando la comparación de la condicional es cierta tiene como proceso adjuntar y enviar el archivo csv mediante Gmail a la cuenta del usuario. Utilizando la condicional “if” se ejecutan dos acciones: si es “false” se ejecuta un “pass” y si es true ejecuta todo el proceso para enviar el

archivo y al final envía un valor booleano “False” a la celda de la nube de datos para terminar el proceso de envío del archivo (Figura 22).

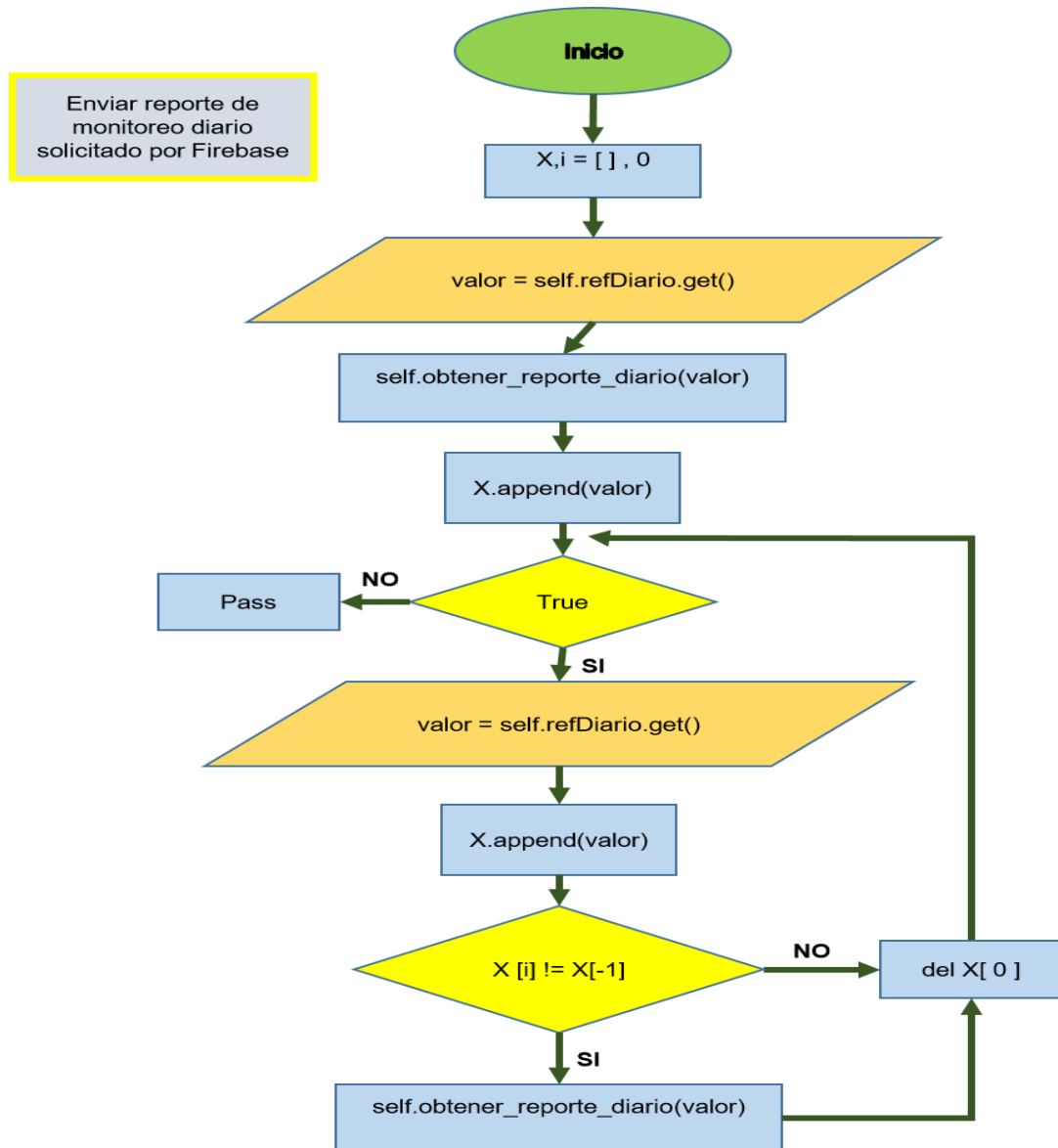


Figura 21. Método de hilo de solicitud de reporte de monitoreo diario por Firebase.

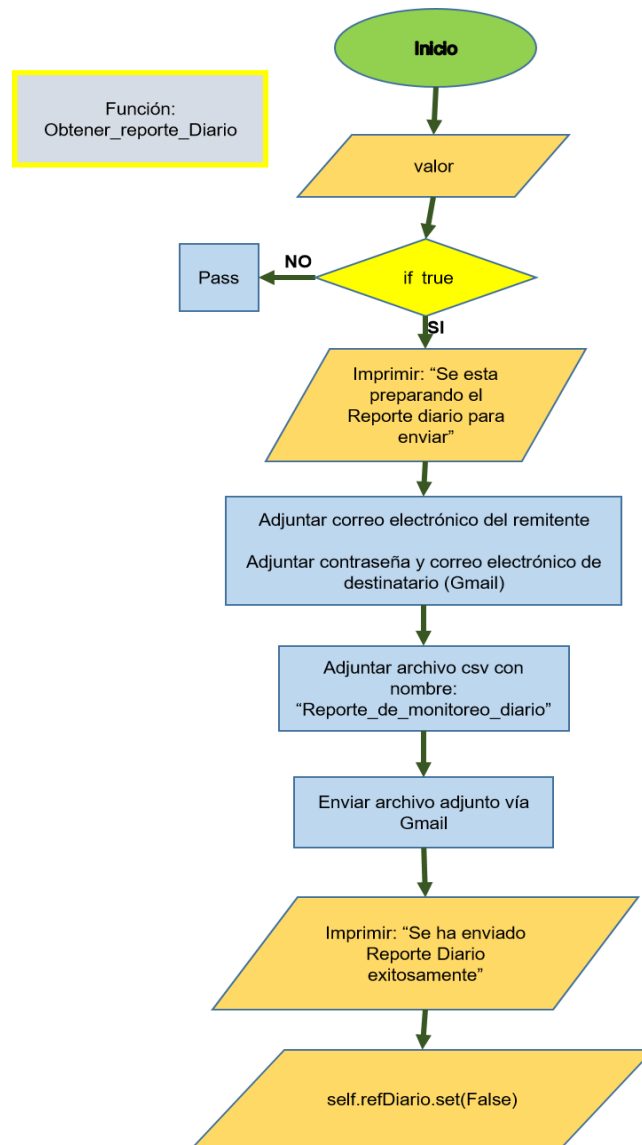


Figura 22. Subproceso de envío de reporte de monitoreo diario solicitado por el método de hilo.

3.2.11. Envío de datos de lectura del sensor de pH a Firebase

Las lecturas del sensor de pH se envían en tiempo real a la nube de datos de Firebase mediante un método de hilo que se vincula con una celda de la Realdatabase de Firebase llamada "pH_agua". Este método de hilo realiza una lectura del sensor cada 10 segundos y la envía a la celda específica para este sensor en la nube de datos, siguiendo una lógica de programación similar al método de hilo que envía los archivos csv cuando son solicitados por Firebase

figura (22), se crea un vector de datos el cual compara la lectura actual del sensor con la lectura anterior, en caso de que sea diferente mediante el comando “. set” envía el nuevo valor a la celda en la nube de datos. Este hilo también tiene la función de crear las variables o atributos que se usaran en los métodos de hilo que crean y escriben los archivos csv, para esto guarda la lectura del sensor en una variable con el prefijo “self” el cual permite que esta pueda utilizarse por cualquier método de la misma clase del módulo Python del Gateway (Figura 23).

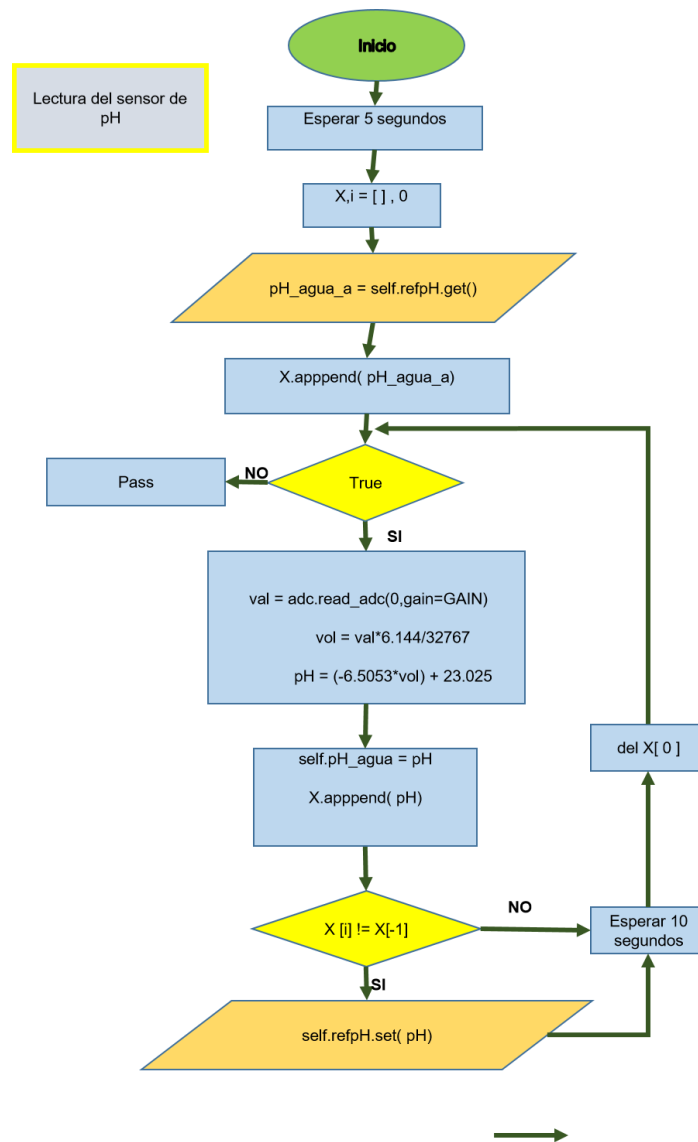


Figura 23. Método de hilo de lectura y envío de dato de pH a Firebase.

3.2.12. Control de bombas a través de la nube de datos Firebase

La activación de las diferentes bombas con las que cuenta el sistema IoT son activadas directamente de la nube de datos de Firebase. Cada bomba que se equipe debe tener un método de hilo similar al que se muestra en la Figura 24. Este hilo crea un vector de datos y una variable con valor cero; después importa el valor actual booleano que tiene la celda de datos de la nube de Firebase que se vinculó con la bomba, lo guarda en un elemento del vector y lo manda a un subproceso que se encarga de activar o desactivar la bomba por un tiempo determinado; después de este proceso el hilo entra en un ciclo “While” infinito; en este ciclo se importa el nuevo valor actual de la celda de datos de Firebase y lo guarda en otro elemento del vector para ser comparado con el anterior; si los elementos del vector son diferentes: envía el último elemento del vector a un subproceso que se encarga de activar o desactivar la bomba.

El subproceso ejecutado activa o mantiene desactivada la bomba; mediante una condición pregunta si el valor booleano recibido es “True” o “False”; si es “False” importa el tiempo de activación de la celda de Firebase en donde el usuario manipula los tiempos de encendido, posterior a esto activa la bomba y la mantiene así por el tiempo de espera importado, una vez terminado el tiempo de espera desactiva la bomba y envía el valor booleano “True” a la celda de nube de datos de Firebase vinculada con la bomba para terminar el proceso de activación (Figura 25).

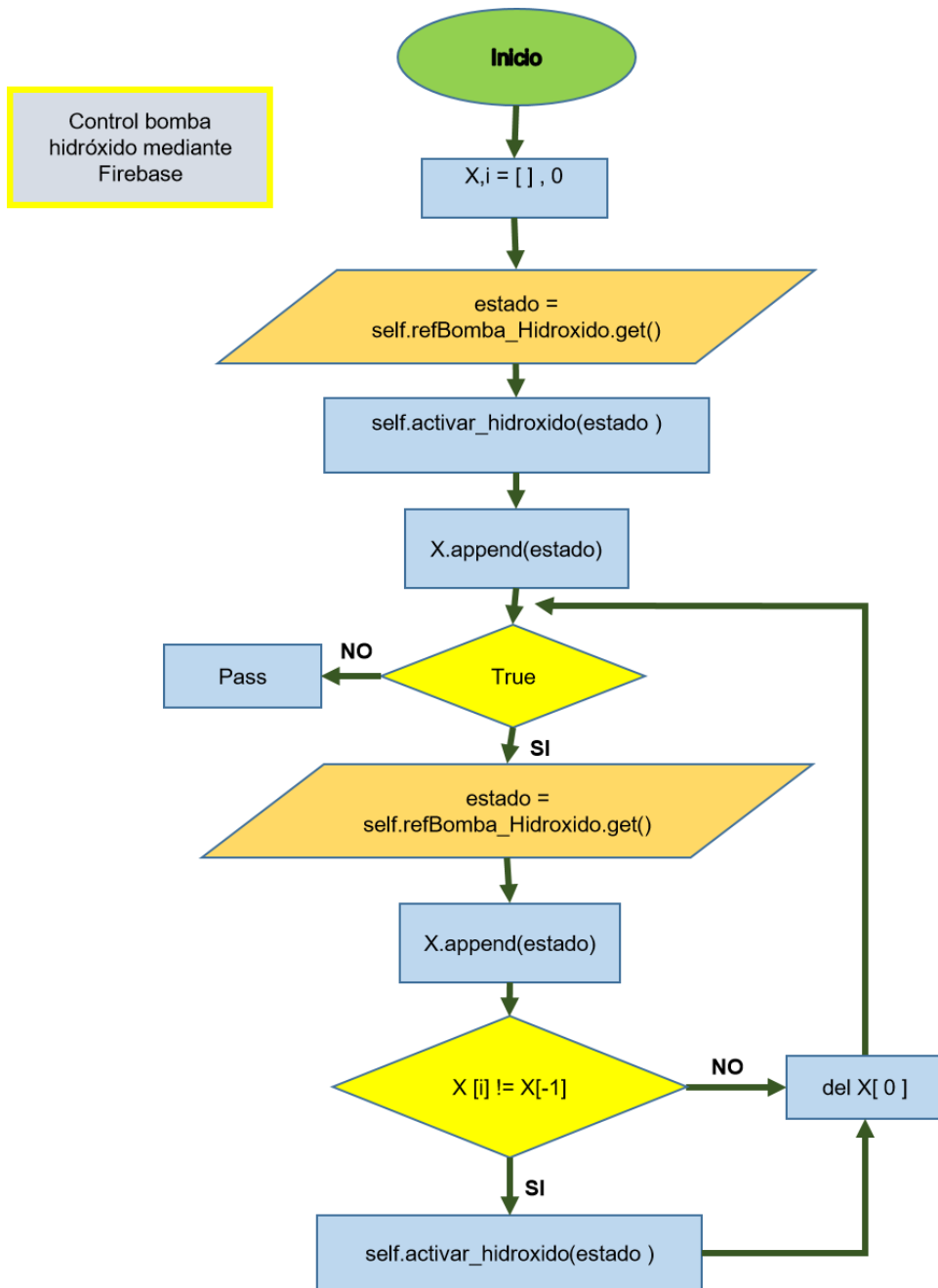


Figura 24. Método de hilo para la activación de bombas por Firebase.

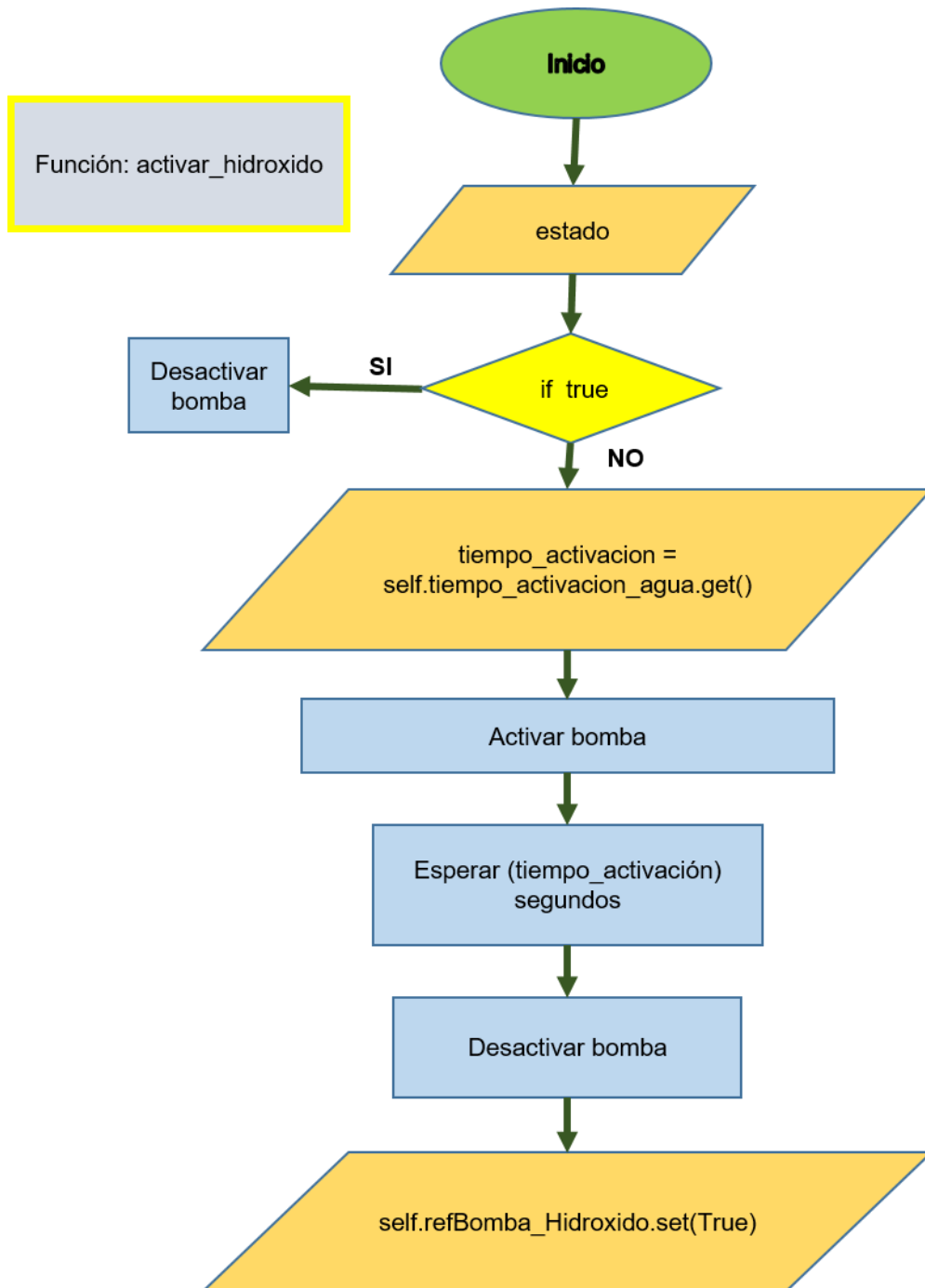


Figura 25. Subproceso para activar bomba por el tiempo determinado por el usuario.

3.2.13. Aplicación móvil en Android Studio

Android Studio es un software libre con IDE que permite desarrollar aplicaciones móviles con la opción de conectarlas a distintas nubes de datos entre ellas Firebase.

La aplicación móvil que se desarrolló para el monitoreo de pH maneja una estructura en donde el usuario puede visualizar la lectura de los datos del sensor en tiempo real los cuales son obtenidos directamente de la nube de datos.



Figura 26. Logo Android Studio.

3.2.14. HelloCharts

HelloCharts es una librería de Android Studio que permite crear gráficas en las aplicaciones que se desarrollan a través de este software. Esta librería permite crear graficas de líneas, barras y circulares; hacer zoom para detallar la escala de los ejes y desplazarse por estos para enfocar un área de interés de la gráfica que se ha creado. Estas características permiten al usuario tener manipulación e interacción con las gráficas y datos observados.

Sensor de temperatura DS18B20 y ultrasónico HC-SR04

Se instalaron dos sensores extra para hacer más complejo el sistema de monitoreo: un sensor de temperatura de agua DS18B20 y un sensor ultrasónico para medir el nivel del agua HC-SR04 (Figura 27). El DS18B20 es un sensor digital de temperatura que utiliza el protocolo 1-Wire para comunicarse, este protocolo necesita solo un GPIO de datos para comunicarse (GPCLK0) que permite conectar más de un sensor en el mismo bus. El sensor ultrasónico HC-SR04 permite medir distancias desde 2cm a 450 cm, consta de dos transductores: un emisor y un receptor; el emisor se encarga de enviar pulsos de ultrasonido que rebotan con el objeto más cercano y son recibidos por el receptor,

el tiempo de demora que tarda el pulso en rebotar y llegar con el receptor es medido y este es procesado por un microcontrolador para calcular la distancia.

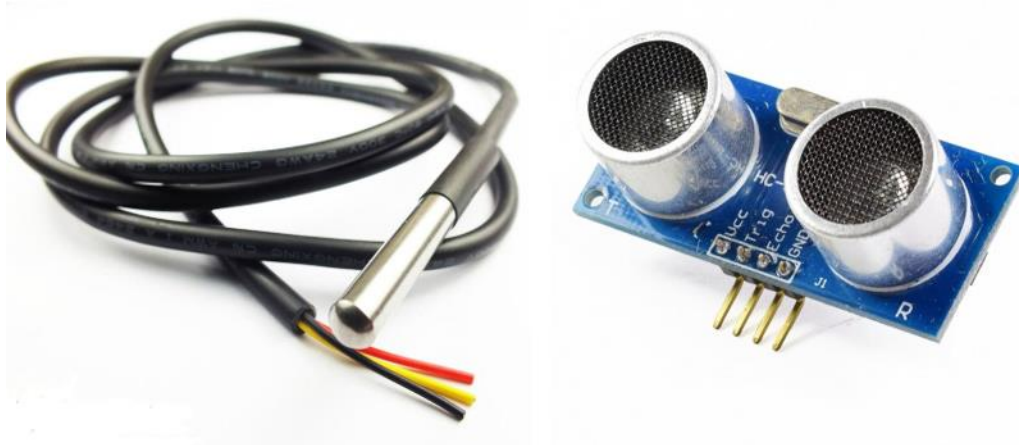


Figura 27. Sensores DS18B20 y HC-SR04

3.3. Resultados y discusión

3.3.1. Comunicación general del sistema IoT

La estación de monitoreo se comunicó vía alámbrica al Gateway coordinador mediante los pines GPIO de la placa Raspberry, enviando las lecturas de los sensores cada 10 segundos y esperando la acción de los actuadores mediante señales de estado (TRUE/FALSE) que recibe directo de los pines GPIO durante el tiempo que el usuario indique directamente en la aplicación móvil.

El Gateway coordinador se comunica vía WIFI con Firebase, esto gracias al módulo wifi con el que cuenta la placa Raspberry. La vinculación de Firebase y el módulo desarrollado en Python para el Gateway se realizó mediante un archivo SDK único que genera de manera automática la plataforma de Firebase, el cual contiene todas las credenciales, certificados y contraseñas que permiten el acceso a la Realtime Database. Firebase y Android se vinculan de manera automática a través de IDE de Android Studio (Figura 28).

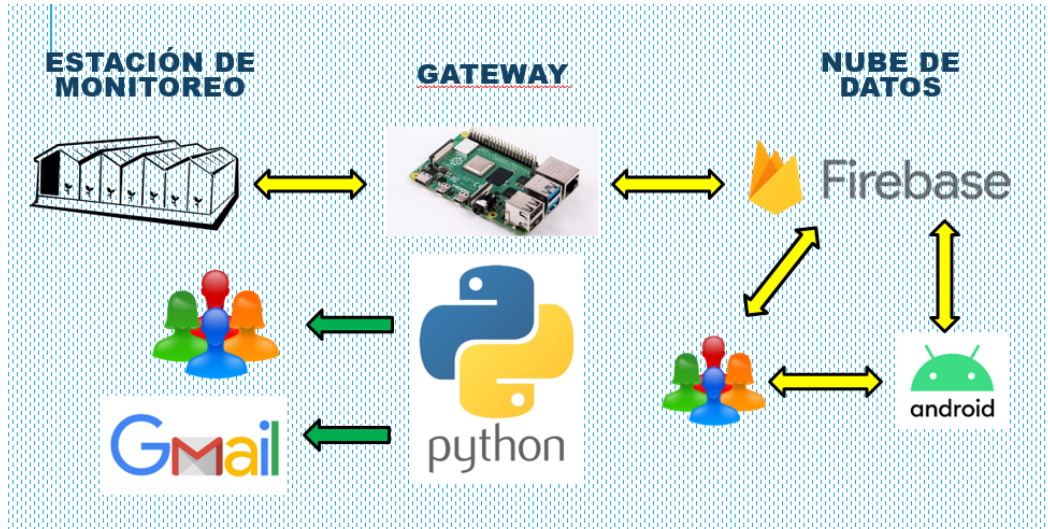


Figura 28. Estructura general de comunicación del sistema IoT.

3.3.2. Lectura del sensor de pH

El sensor omega PHE-1304 se calibró mediante una regresión lineal de 3 soluciones buffers conocidas: solución Omega pH 4.00 ± 0.01 a $25\text{ }^{\circ}\text{C}$, solución Omega pH 7.00 ± 0.01 a $25\text{ }^{\circ}\text{C}$ y solución Omega pH 10.00 ± 0.01 a $25\text{ }^{\circ}\text{C}$ obteniendo la siguiente ecuación: $y = -6.5053x + 23.025$. Con esta configuración se realizaron las pruebas de obtención de datos.

La sonda de pH se conectó al módulo PH-4502C y este al convertidor analógico a digital ADS115 para poder conectarse con los pines SDA y SCL de la placa Raspberry (Figura 29). Para realizar la lectura se instaló la librería Adafruit_ADS1x15.

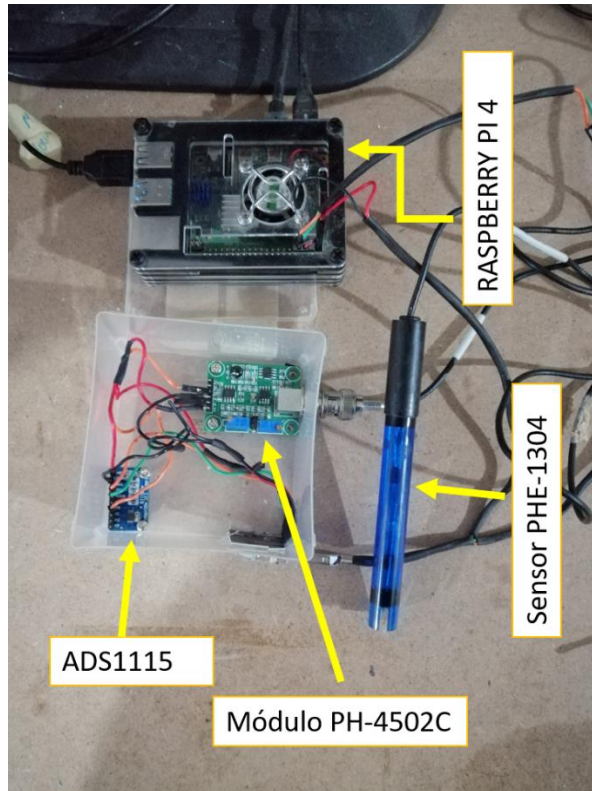


Figura 29. Conexión de sensor de pH con la placa Raspberry.

3.3.3. Visualización de datos del Gateway en la estación de Monitoreo

En seguida que se ejecuta el módulo Python del Gateway en el monitor conectado a la placa Raspberry aparece la conexión exitosa con Firebase, mediante un mensaje de prueba se comprueba la conexión con Gmail, nos muestra el estado de los actuadores, las primeras lecturas de los sensores y el mensaje de que se empiezan a crear los reportes de monitoreo en archivo CSV como se muestra en la Figura 30. En caso de que alguno de los elementos que componen al sistema IoT falle, aparece el mensaje de error y se tiene que parar el programa para eliminar el error para volverlo a ejecutar de manera exitosa (Figura 31).

```

Python 3.7.3 (/usr/bin/python3)
>>> %Run IAUIA.py

START IOT IAUIA!
Se ha conectado EXITOSAMENTE con FIREBASE
Se está enviando mensaje de prueba a Gmail
Se está preparando base de datos
Se está preparando el Nuevo reporte
Bomba_RECIRCULACION OFF
Bomba_ACIDO_SULFURICO OFFBomba_AGUA OFF

Bomba_ACIDO_NITRICO OFF
Bomba_HIDROXIDO OFF
Se ha enviado MENSAJE DE PRUEBA a Gmail EXITOSAMENTE
El pH anterior del agua es: 7.577693735624255
El pH actual del agua es: 7.653320156059449
la temperatura anterior del agua es: 17.812 °Centigrados
la temperatura actual del agua es: 17.812 °Centigrados
Nivel de agua anterior es: 48.27
nivel de agua actual es: 48.69
El pH anterior del agua es: 7.653320156059449
El pH actual del agua es: 7.525243153709523
la temperatura anterior del agua es: 17.812 °Centigrados
la temperatura actual del agua es: 17.812 °Centigrados
Nivel de agua anterior es: 48.69
nivel de agua actual es: 48.24

```

Figura 30 Estado de arranque exitoso del módulo Python del Gateway.

```

Shell x
START IOT IAUIA!
Se está preparando base de datos
Se está preparando el Nuevo reporte
Se está enviando mensaje de prueba a Gmail
Traceback (most recent call last):
  File "/home/pi/IAUIA.py", line 885, in <module>
    envio_mensaje_prueba
NameError: name 'envio_mensaje_prueba' is not defined

>>> Bomba_RECIRCULACION OFF
Bomba_AGUA OFF
Bomba_ACIDO_NITRICO OFF
Bomba_HIDROXIDO OFF
Bomba_ACIDO_SULFURICO OFF
Se ha enviado MENSAJE DE PRUEBA a Gmail EXITOSAMENTE
Exception in thread Thread-9:
Traceback (most recent call last):
  File "/usr/lib/python3.7/threading.py", line 917, in bootstrap_inner
    self.run()
  File "/usr/lib/python3.7/threading.py", line 865, in run
    self._target(*self._args, **self._kwargs)
  File "/home/pi/IAUIA.py", line 497, in crear_csv
    valor_pH_csv = self.pH_agua_csv
AttributeError: 'IOT' object has no attribute 'pH_agua_csv'

Exception in thread Thread-11:
Traceback (most recent call last):
  File "/usr/lib/python3.7/threading.py", line 917, in bootstrap_inner
    self.run()
  File "/usr/lib/python3.7/threading.py", line 865, in run
    self._target(*self._args, **self._kwargs)
  File "/home/pi/IAUIA.py", line 573, in reportediario_csv
    valor_pH_csv1 = self.pH_agua_csv
AttributeError: 'IOT' object has no attribute 'pH_agua_csv'

El pH anterior del agua es: 7.277627615832998
El pH actual del agua es: 7.671616870680866

```

Figura 31. Arranque con error de software en el módulo Python del Gateway.

3.3.4. Estructura de la nube de datos Firebase

Se logró una conexión exitosa entre el Gateway coordinador (Raspberry con Python) y la Nube de datos (Firebase). Visualizando en tiempo real la lectura del sensor de pH y otros más que se instalaron en la estación de monitoreo y se agregaron a la red de comunicación IoT. La base de datos de Firebase muestra los datos a través de una tabla en donde las celdas son capaces de recibir y enviar valores booleanos, los cuales sirven para poder controlar actuadores o iniciar algún subproceso del módulo Python (Figura 32) .

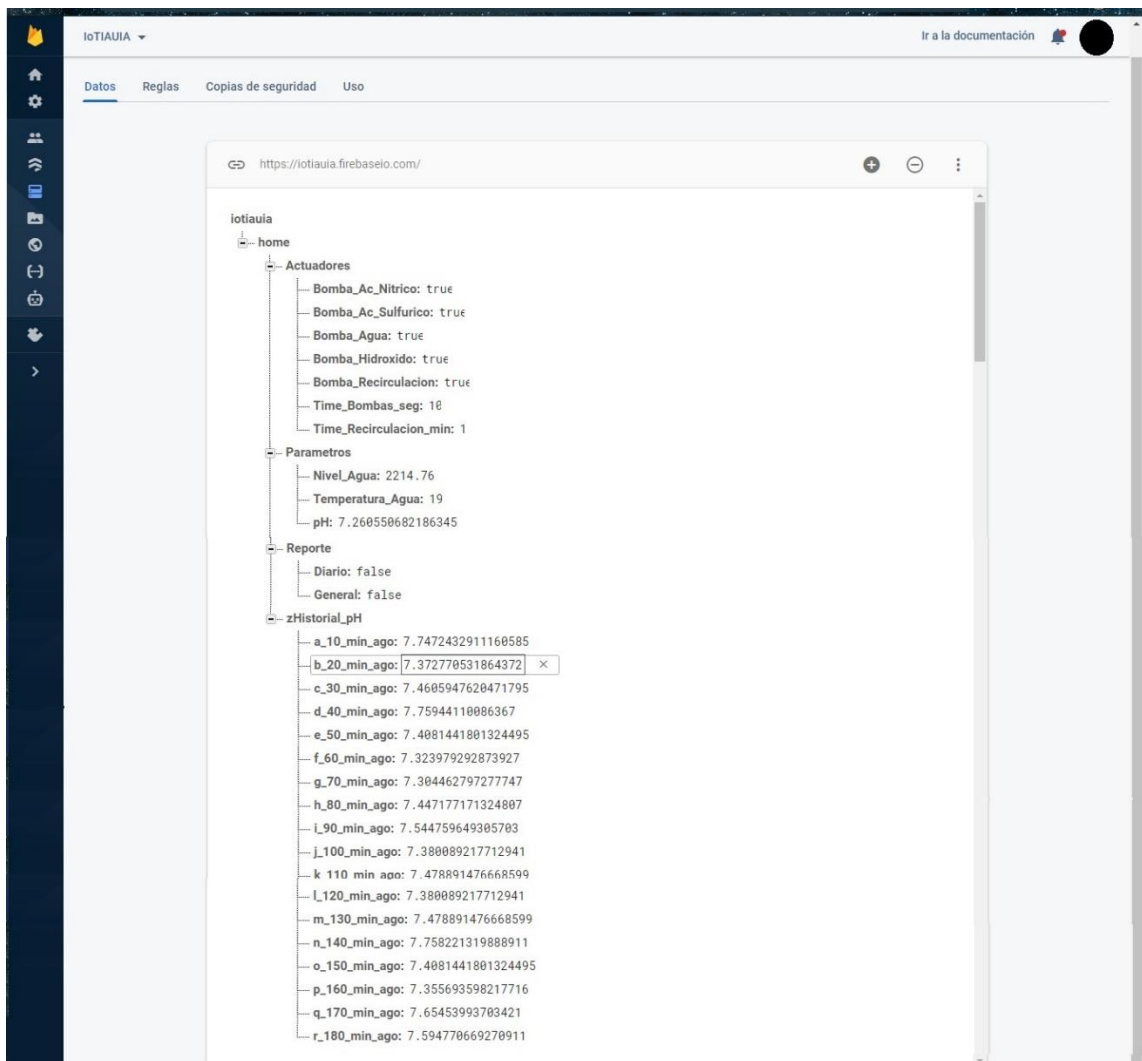


Figura 32. Estructura en formato JSON de la nube de datos Firebase.

3.3.5. Obtención de datos de la estación de monitoreo

El sistema IoT para el monitoreo de parámetros del sistema en tiempo real, da la información del estado de los actuadores o sensores que se encuentran en el sistema, la cual es mostrada a través de la página web de Firebase o la aplicación Android desarrollada y puede ser consultada por el usuario de manera remota gracias a la autenticación que se configura en el programa principal. Para lograr esta autenticación Firebase ofrece un conjunto de llaves y credenciales únicas, cada vez que se desarrolla un proyecto, y el acceso se logra con la dirección del correo electrónico de Google, facilitando la creación de nuevos proyectos y evitando la necesidad de crear páginas web para poder obtener el estado del sistema.

Para lograr la actualización automática de la plataforma creada en la base de datos de Firebase, se da la prioridad al método del programa que crea la estructura de la base de datos, esto se logró gracias al comando “`__init__`” de Python, el cual define a la función principal de la clase. De esta manera cualquier actualización de estado en algún parámetro o actuador se envía en tiempo real a Firebase.

La clase Thread o Hilos utilizada en el módulo de Python del Gateway permitió la ejecución individual de los subprocesos incluso si un atributo o subproceso detiene su ejecución debido a un error (Figura 33).

```
Shell »
El pH anterior del agua es: 7.675276213605148
El pH actual del agua es: 7.774078472560808
Bomba_ACIDO_NITRICO ON
Exception in thread Thread-10:
Traceback (most recent call last):
  File "/usr/lib/python3.7/threading.py", line 917, in bootstrap inner
    self.run()
  File "/usr/lib/python3.7/threading.py", line 865, in run
    self._target(*self._args, **self._kwargs)
  File "/home/pi/IAUIA.py", line 444, in crear_csv
    valor_nivel_csv = self.nivel_csv
AttributeError: 'IOT' object has no attribute 'nivel_csv'

la temperatura anterior del agua es: 21.125 °Centigrados
la temperatura actual del agua es: 21.125 °Centigrados
bomba nitrico on
Bomba_ACIDO_NITRICO ON
El pH anterior del agua es: 7.774078472560808
El pH actual del agua es: 7.720408109671316
Bomba_ACIDO_NITRICO OFF
bomba nitrico off
la temperatura anterior del agua es: 21.125 °Centigrados
la temperatura actual del agua es: 21.125 °Centigrados
Bomba_ACIDO_NITRICO OFF
sulfurico = 1 nitrico = 1
Bomba_ACIDO_SULFURICO OFF
bomba sulfurico on
Bomba_ACIDO_SULFURICO OFF
El pH anterior del agua es: 7.720408109671316
El pH actual del agua es: 7.6435619082613595
```

Figura 33. Fallo en hilo en módulo python.

3.3.6. Creación y envío de Reportes de Monitoreo (csv) mediante Gmail

Mediante la librería de PANDAS de Python se crearon dos archivos csv: uno con los valores que toma el sensor de pH cada minuto durante 24 Horas y otro con los valores que toma el sensor de pH cada 10 minutos desde el arranque del sistema IoT hasta su detención.

Los reportes de monitoreo diario son enviados de manera automática cada 24 horas, a la que se especifique en el módulo Python a través de Gmail (Figura 34 y 35), pero también pueden ser solicitados por el usuario en cualquier instante mediante la aplicación Android que se desarrolló. Estos reportes se eliminan de manera automática una vez que se enviaron para iniciar otro ciclo de captura de datos durante otras 24 horas (Figura 36).


```

def envio_reporte_diario(self):
    while True:
        hora_actual = datetime.now().time()
        #print(hora_actual)
        hora = hora_actual.hour
        #print(hora)
        minuto = hora_actual.minute
        #print(minuto)
        minuto = hora_actual.minute
        if hora == 3 and minuto == 15:
            print('se está PREPARANDO para Enviar el REPORTE DIARIO')
            mensaje = MIME multipart("plain")
            mensaje["From"] = "*****"
            mensaje["To"] = "*****"
            mensaje["Subject"] = "Reporte monitoreo diario"
            adjunto = MIMEBase("application", "octect-stream")
            adjunto.set_payload(open("Reporte_diario.csv", "rb").read())
            adjunto.add_header("content-Disposition", 'attachment; filename = "Reporte_diario.csv"')
            mensaje.attach(adjunto)
            server = smtplib.SMTP('smtp.gmail.com', 587)
            server.starttls()
            server.login('*****', '*****')
            server.sendmail(*****, *****+mensaje.as_string())
            server.quit()
            print('Se ha enviado REPORTE DIARIO EXITOSAMENTE')
            time.sleep(61)
        else:
            pass

```

Figura 34. Código para enviar reporte de monitoreo mediante Gmail automáticamente.

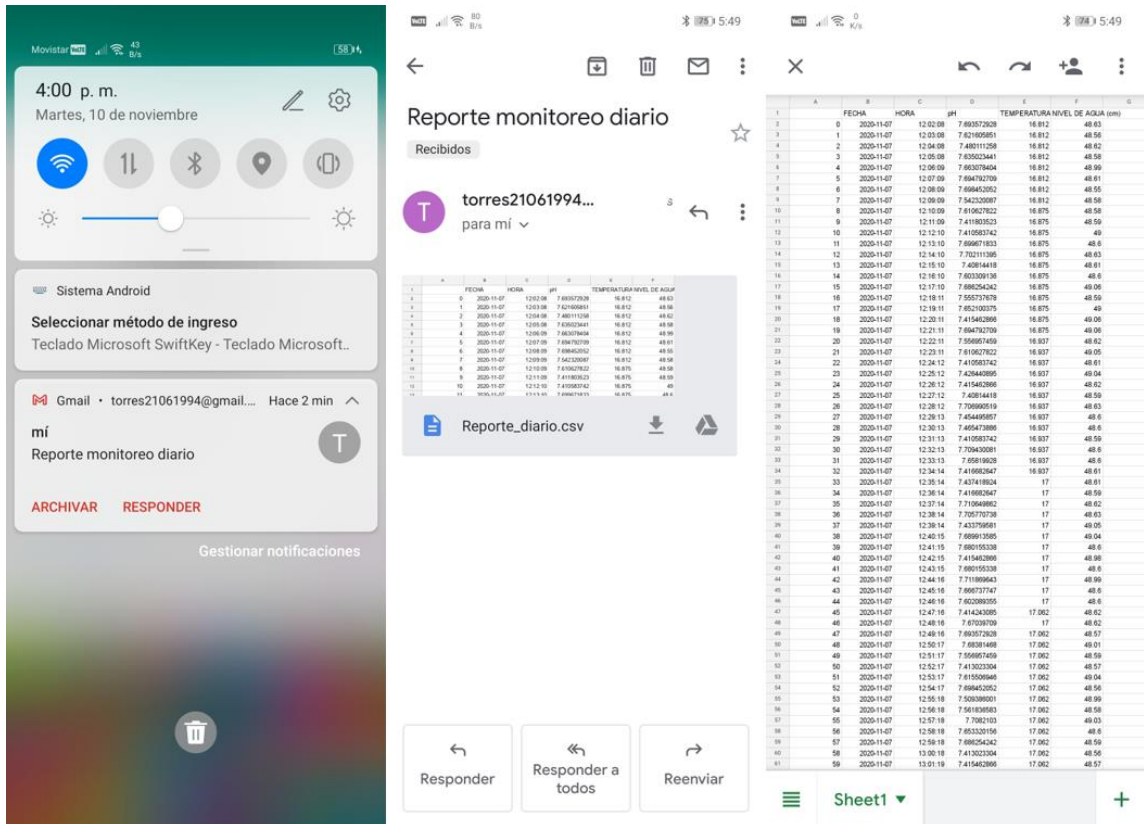


Figura 35. Recepción de Reporte Diario mediante Gmail en celular.


```
Shell x
la temperatura actual del agua es: 17.012 Centigrados
Nivel de agua anterior es: 26.02
nivel de agua actual es: 27.7
El pH anterior del agua es: 7.898496131986446
El pH actual del agua es: 7.860682921768852
la temperatura anterior del agua es: 17.812 °Centigrados
la temperatura actual del agua es: 17.012 Centigrados
se está PREPARANDO para Enviar el REPORTE DIARIO
nivel de agua anterior es: 27.7
nivel de agua actual es: 212.55
El pH anterior del agua es: 7.860682921768852
El pH actual del agua es: 7.775298253535569
Se ha enviado REPORTE DIARIO EXITOSAMENTE
la temperatura anterior del agua es: 17.812 Centigrados
la temperatura actual del agua es: 17.812 °Centigrados
El pH anterior del agua es: 7.775298253535569
El pH actual del agua es: 7.530122277608566
Nivel de agua anterior es: 212.55
nivel de agua actual es: 26.86
la temperatura anterior del agua es: 17.812 °Centigrados
la temperatura actual del agua es: 17.812 °Centigrados
El pH anterior del agua es: 7.530122277608566
El pH actual del agua es: 7.53256183955809
Nivel de agua anterior es: 26.86
nivel de agua actual es: 26.86
Se termino reporte diario
se escribio en base de datos
la temperatura anterior del agua es: 17.812 °Centigrados
la temperatura actual del agua es: 17.812 °Centigrados
El pH anterior del agua es: 7.53256183955809
El pH actual del agua es: 7.538660744431896
Nivel de agua anterior es: 26.86
nivel de agua actual es: 27.25
El pH anterior del agua es: 7.542320087356181
El pH actual del agua es: 7.904595036860252
la temperatura anterior del agua es: 17.875 °Centigrados
la temperatura actual del agua es: 17.812 °Centigrados
El pH anterior del agua es: 7.904595036860252
El pH actual del agua es: 7.905814817835013
Nivel de agua anterior es: 27.25
nivel de agua actual es: 27.27
Se elimino reporte anterior
Se está preparando el Nuevo reporte
la temperatura anterior del agua es: 17.875 Centigrados
la temperatura actual del agua es: 17.812 °Centigrados
se escribio en base de datos
El pH anterior del agua es: 7.905814817835013
El pH actual del agua es: 7.545979430280463
```

Figura 36. Notificación en el Gateway de preparación, envío mediante Gmail, eliminación y comienzo de nuevo de reporte diario.

El reporte General es enviado de manera automática una vez en el día del mes que especifique el usuario en el módulo Python, y al igual que el reporte de monitoreo diario, este puede ser enviado en cualquier instante que el usuario lo solicite a través de la aplicación Android.

Para vincular la cuenta de Gmail con el módulo Python es necesario instalar las librerías: MIMEMultipart, MIMEBase y smtplib; Además de que se tiene que permitir el acceso de apps menos seguras directamente de la página de configuración de cuenta de Google.

Al iniciar el sistema IoT se envía un mensaje de prueba para comprobar la conexión del sistema con Gmail (Figura 37 y 38) De la misma manera se eliminan, si es que existen, el reporte de monitoreo diario y/o el reporte general, esto para

evitar problemas de sobre escritura en los archivos csv, lo cual puede provocar que se dañe el archivo (Figura 37).

```
START IOT IAUIA!  
Se está enviando mensaje de prueba a Gmail  
Se eliminó reporte anterior  
Se eliminó Base de Datos anterior  
Se está preparando base de datos  
Se está preparando el Nuevo reporte  
Base IAUIA OFF
```

Figura 37. Notificación de eliminación de reportes al ejecutar el programa IoT.

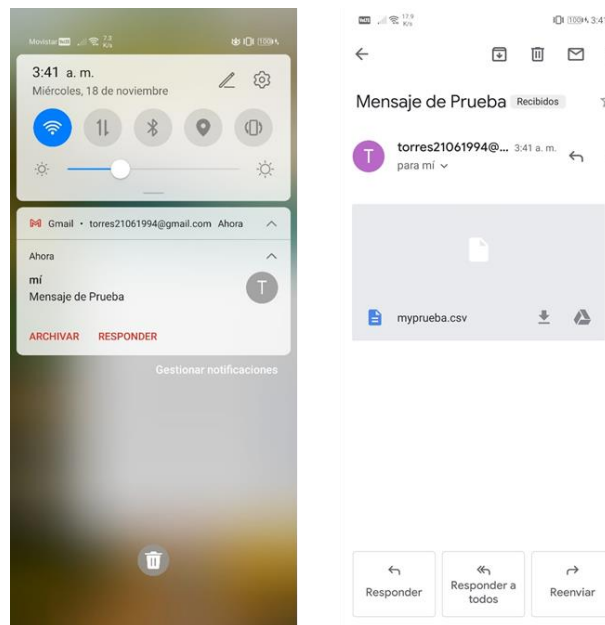


Figura 38. Recepción de mensaje de prueba para comprobar la conexión del Gateway con Gmail.

3.3.7. Aplicación móvil Android

En la interfaz gráfica de la aplicación Android (Figura 39) se encuentran dos botones con el que se obtienen los archivos csv: el primero obtiene el reporte de lecturas diario y el segundo reporte general de lecturas, es decir, desde que se inició el sistema de monitoreo. Estos archivos son solicitados a Firebase y enviados vía Gmail por el Gateway (Figura 40). Debajo de estos botones se muestran la lectura de los sensores en tiempo real y una gráfica con el historial

de lecturas tomadas en las últimas 3 horas en lapsos de 10 minutos cada punto. En la sección final de la aplicación encontramos los botones de control que encienden las bombas y dos cuadros en los que se modifican los tiempos de encendido (Figura 40).

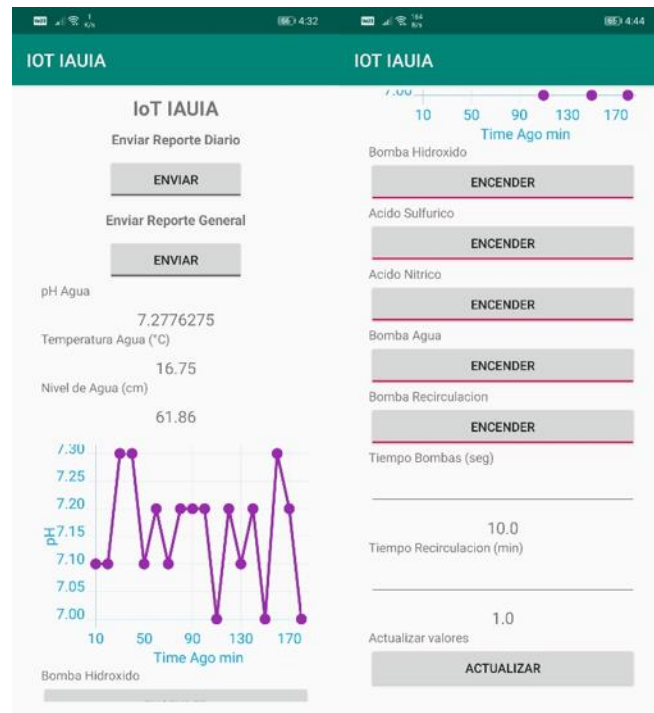


Figura 39. Interfaz gráfica de la aplicación móvil.

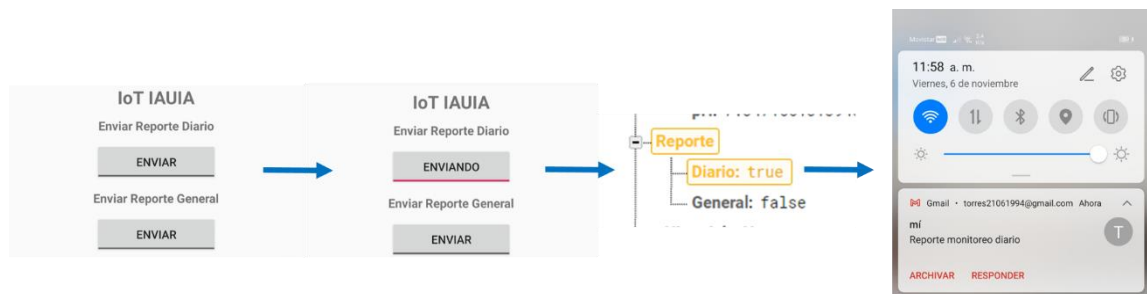


Figura 40. Obtención de reportes de monitoreo a través de Gmail mediante la aplicación móvil.

La grafica mostrada en la aplicación tiene programado el eje “y” de manera que la escala, y el valor máximo y mínimo mostrado se configuren de manera automática dependiendo de los valores obtenidos por Firebase de las lecturas del sensor de pH en los últimos 180 minutos. Esto con la finalidad de que la

visualización por el usuario sea más amena y ubique entre que valores fluctúa el parámetro en el lapso de tiempo mencionado anteriormente (Figura 41).

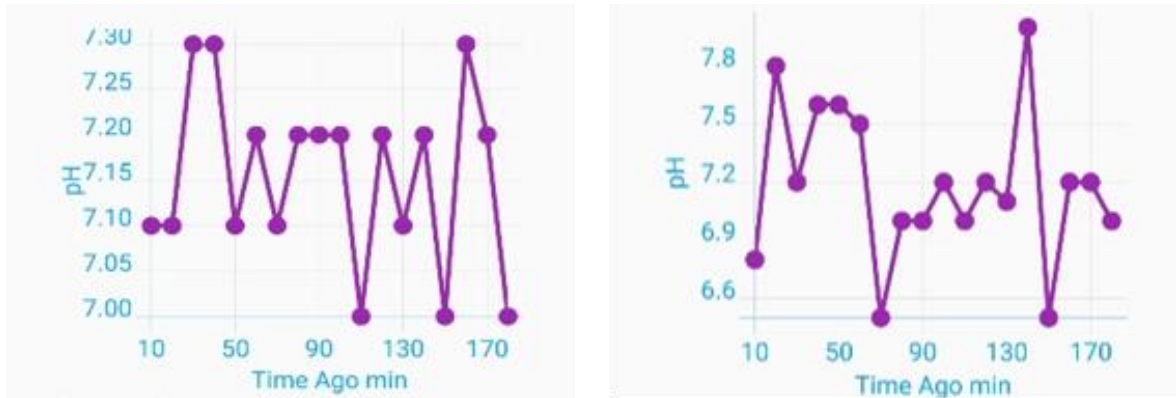


Figura 41. Ejemplo de la configuración de la escala del eje “Y” en la grafica de aplicación móvil.

Para comprobar que se envíen las señales de control de los actuadores a través de la aplicación Android, se conectaron bombas de 12 volts mediante un módulo de relevadores a los pines de la placa Raspberry. Los botones instalados en aplicación Android para el control de bombas envían valores booleanos a la nube de datos Firebase y esta envía los valores al Gateway que los procese como estado de Encendido/Apagado mediante los pines GPIO seleccionados en el módulo Python (Figura 42 y 43). El estado de encendido activa el relevador de la bomba por el tiempo seleccionado por el usuario en la aplicación móvil (Figura 44).

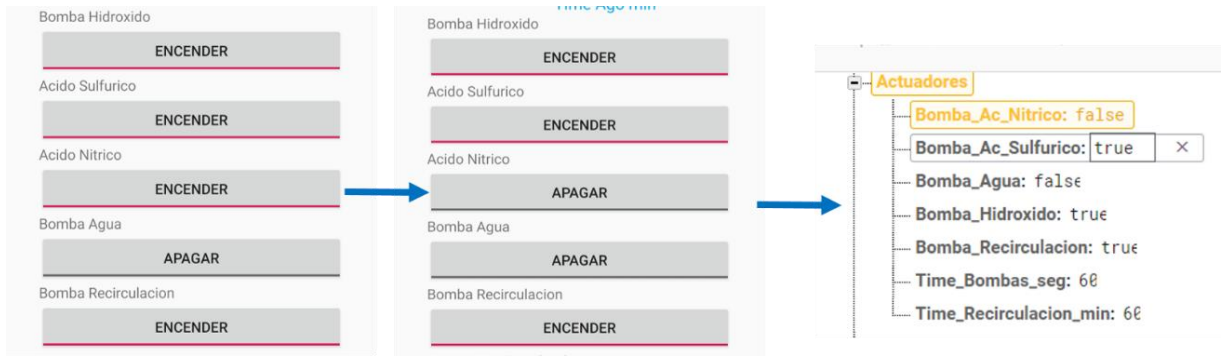


Figura 42. Encendido de bomba mediante la aplicación Android.

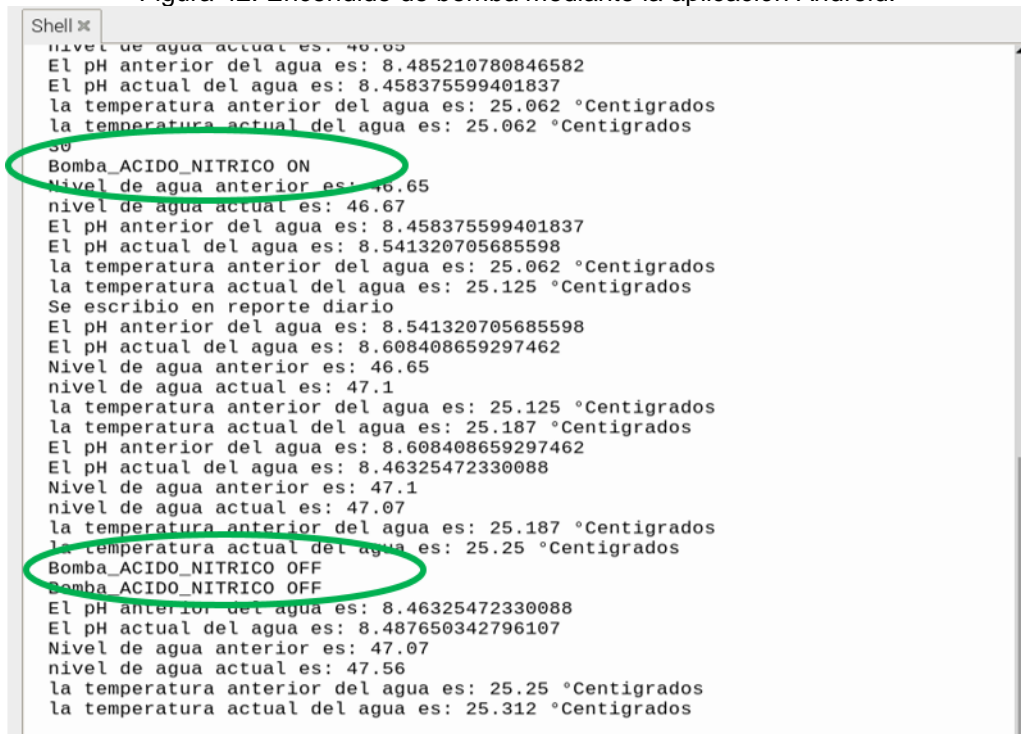


Figura 43. Interfaz de Gateway con el mensaje de encendido/ Apago de Bombas.



Figura 44. Cambio de tiempo de encendido de bombas desde la aplicación Android.

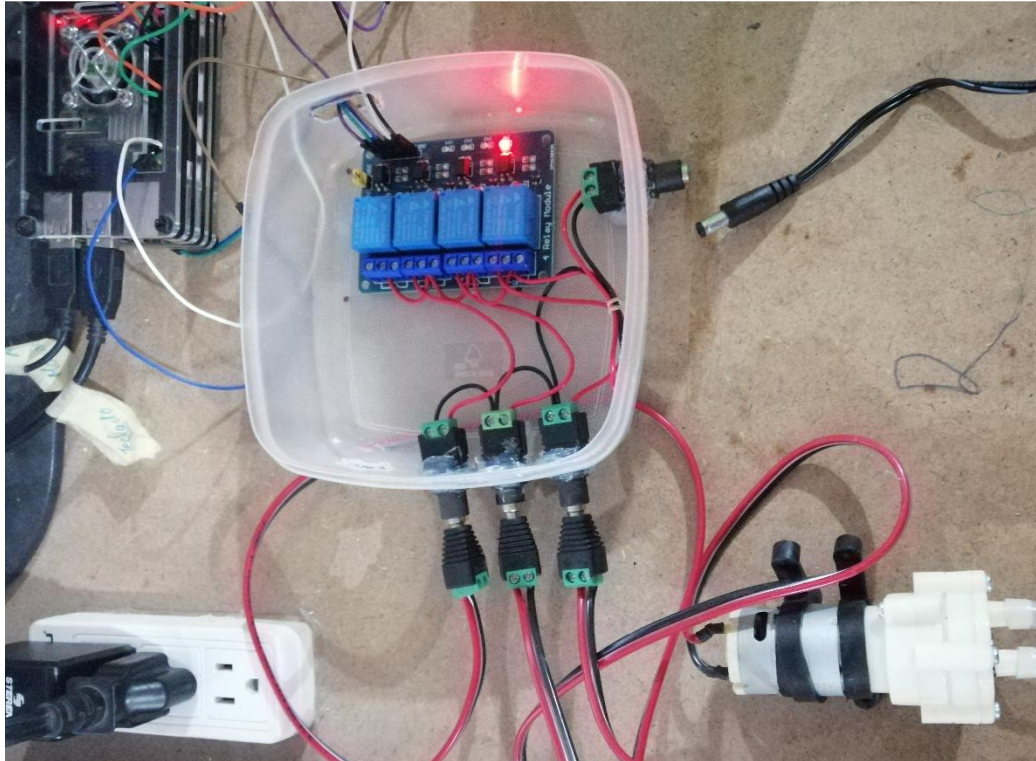


Figura 45. Prueba de encendido de Bombas mediante androd, Firebase y python.

3.4. Conclusiones

Este trabajo muestra la exitosa implementación en la acuaponía de la tecnología basada en Internet de las Cosas (IoT) con una red de comunicación de bajo costo, con la cual se pudo monitorear datos de pH en tiempo real y transmitir señales de control a la estación de monitoreo de manera remota. Los resultados indican que IoT podría ser una alternativa al campo mexicano para la modernización y adaptación a la tecnología basada en Internet de las Cosas, cuyo principal objetivo es administrar mejor los insumos durante el ciclo de los cultivos, disminuyendo los costos de producción y aumentando la productividad.

Se comunicó mediante redes inalámbricas al sistema IoT. Utilizando WIFI y un plan de Firebase sin costo de operación, lo cual reduce en gran medida el costo total del sistema de comunicación.

Este trabajo no mide la eficiencia del sistema IoT en la producción del cultivo directamente, ya que las pruebas de monitoreo y control de los sensores y actuadores se realizaron de manera externa al invernadero, probando los nodos de comunicación solamente.

3.5. Literatura citada

- Buck, P., Rondinini, Covington, K., Baucke, G., Brett, M., Camoes, F., Milton, J., Mussini, T., Naumann, R., Pratt, W., Spitzer P., & Wilson, S. (2002). Measurement of pH definitions, standards, and procedures (IUPAC Recommendations 2002). *Pure Appl. Chem*, 74 , 2169–2200.
- Carletti, E.J. (2007). *Comunicación - Bus I2C: Descripción y funcionamiento*. Recuperado de <http://www.bolanosdj.com.ar/MOVIL/ARDUINO2/ComunicacionBusI2C.pdf>
- García, D. (2017). *Solución domótica mediante el desarrollo de una aplicación en iOS y plan de empresa basada en Internet de las Cosas* (Tesis de pregrado). Universidad de Carlos III de Madrid, España. Recuperado de <https://e-archivo.uc3m.es/handle/10016/27035>.
- Guerrero I., J.A., Estrada G., F.P., Medina T., M. A. (2017). SGreenH-IoT: Plataforma IoT para Agricultura de Precisión. *Sistemas, cibernética e informática*, 14, 53 – 58.
- Martín C.,S. (2014). Raspberry Pi, Arduino y Beaglebone Black Comparación y Aplicaciones. *Ingeniería y electrónica*, 1, 4-8.
- May T., O.J., Quijano X., R. G., & Fernández C., F. (2015). Sistema domótico de control de luces mediante una aplicación Android basado en una tarjeta Raspberry-Pi. *Tendencias de Investigación y Desarrollo en la Ingeniería*, 1, 45-49, doi: 10.13140/RG.2.1.1315.2087.
- Mercado, A. & Gonzales, A. (2018). *Propuesta para el diseño de una red de sensores implementando el internet de las cosas aplicado en la agricultura de precisión* (Tesis de pregrado). Instituto Politécnico Nacional, México.
- Oreggioni, J. (2006). Métodos de medida del pH para estudios invasivos prolongados: ISFET y fibra óptica. *XV Seminario de Ingeniería Biomédica – Facultades de Medicina e Ingeniería*. Seminario llevado a cabo en la Universidad de la República Oriental del Uruguay, Uruguay.
- Pimentel O., E., Yate M., Y. & Porras B., J. E. (2015). Actuadores inteligentes con protocolo de comunicación I2C y medio físico RS-485. *Tekhné*, 12, 59-66.
- Pineda, J., Miranda, I., Ramirez, J., Rivera, R., Vargas, M., Roldán, V. & García, A. (2018). Culture of lily (*Lilium sp.*) 'Table Dance' in an aquaponic system. *Acta Horti.*, 1227, 355-364, doi: 10.17660/ActaHortic.2018.1227.43.

Ruano, G. (2017). *Construcción de un invernadero con un sistema de control y monitorización de las variables del entorno* (Tesis de pregrado). Universidad de Almería, España. Recuperado de http://repositorio.ual.es/bitstream/handle/10835/6622/16201_TFG.pdf?sequence=1.

Zamora, G. (2019). *Sistema de vermicompostaje con internet de las cosas*. (Tesis de pregrado). Universidad de Chile, Chile. Recuperado de <http://repositorio.uchile.cl/handle/2250/173417>.

4. ANEXOS

ANEXO 1. Costo del sistema IoT

Es el costo total de los elementos empelados para desarrollar la intercomunicación IoT del sistema. Incluye todos los elementos utilizados en este proyecto para poder desarrollar el software y hardware. En este análisis se dividió en dos categorías: a) Costos del Gateway y la red de sensores y actuadores utilizados en la estación d monitoreo; b) Costos de conexión con la nube de datos y aplicación móvil.

Costo del Gateway

Cuadro 2. Costos de elementos del Gateway.

Item	Elemento	cantidad	Precio unitario (MXN)	Total (MXN)
1	Placa Raspberry pi 4 Modelo B+ de 2 Gb	1	1228.45	1228.45
2	Case de protección de acrílico, con ventilador y disipadores de calor para placa Placa Raspberry pi 4 B+	1	240.00	240.00

3	Sensor de pH Omega PHE- 1304	1	3,035.00	3,035.00
4	Convertidor Analógico- Digital ADS1115	1	72.58	72.58
5	Cable ultraslim HDMI a Micro HDMI	1	189.66	189.66
6	Convertidor HDMI a VGA	1	240.00	240.00
7	Cargador USB C 5V 3A 15W	1	232.76	232.76
8	Arnés USB C	1	215.52	215.52
9	Teclado Alfanumérico USB	1	155.17	155.17
10	Mouse alámbrico USB	1	130.00	130.00
11	Monitor Dell 17 pulgadas Lcd	1	1240	1240
12	Memoria Micro SD de 16 Gb clase 10	1	159.20	159.20

13	Módulo de 4 Relevadores	1	69.31	69.31
14	Mini bomba de diafragma 12v	3	106.80	320.40
15	Cables, soldadura, cautín y extras	1	600.00	600.00
Costo Total				8,125.05

Costos de la Nube de datos

En el cuadro 3 se desglosan los costos de conexión del Gateway coordinador con la nube de datos y esta a su vez con la aplicación móvil durante un año utilizando redes de internet WIFI Telmex.

Cuadro 3. Costo de la nube de datos por un año.

Ítem	Elemento	Cantidad	Precio unitario (MXN)	Total (MXN)
1	Plan Spark	12	0.00	0.00
	Firestore			
2	Servicio de internet inalámbrico Telmex	12	380	3800
Costo Total				3,800

ANEXO 2. Librerías Python

Para el desarrollo del software del módulo de Python fue necesario instalar las librerías que se muestran en el siguiente cuadro (Cuadro 4):

Cuadro 4. Librerías utilizadas en el módulo Python del sistema IoT.

Librería	Aplicación en el Software
Sys	Crea lista de elementos y/o módulos que podemos importar en el módulo principal.
Time	Lapso de tiempo para funciones de sleep.
Gpiozero	Uso de pines GPIO.
Firebase_admin	Vinculación con la nube de datos Firebase.
Adafruit_ADS1x15	Comunicación con el convertidor Analógico a Digital.
Threading	Crear Hilos en el Módulo.
Pandas	Manipulación de datos, Creación de archivos CSV.
Remove	Eliminar archivos.
Datetime	Uso de fecha, Hora.
MIMEMultipart	Comunicación con correo electrónico.
MIMEBase	Comunicación con correo electrónico.
smtplib	Comunicación con correo electrónico.

```

1  #LIBRERIAS DEL SISTEMA
2  #*****
3  import sys
4  from time import sleep
5  import signal
6  from gpiozero import LED, Button
7  from threading import Thread
8  #-----
9  #Librerías para Firebase
10 import firebase_admin
11 from firebase_admin import credentials
12 from firebase_admin import db
13 #-----
14 #LIBRERIAS PARA SENSOR de pH DE AGUA
15 #import time
16 import Adafruit_ADS1x15
17 #LIBRERIAS PARA LA BASE DE DATOS ARCHIVO CSV
18 import pandas as pd
19 import datetime
20 from datetime import date,datetime
21 from os import remove
22 #LIBRERIAS PARA EL ENVIO DE E-MAIL
23 from email.mime.multipart import MIMEMultipart
24 from email.mime.base import MIMEBase
25 import smtplib
26

```

Figura 46. Importación de librerías en el módulo Python.

ANEXO 3. Librerías Android Studio

Cuadro 5. Librerías utilizadas para la aplicación móvil en Android Studio.

Librería	Aplicación en el Software
Hellocharts	Crear graficas en la aplicación móvil.
Android.graphics.Color	Crear graficas en la aplicación móvil.
Android.os.Bundle	Crear graficas en la aplicación móvil.
Java.util.ArrayList	Crear graficas en la aplicación móvil.

Java.util.List	Crear graficas en la aplicación móvil.
androidx.appcompat.app.AppCompatActivity	Funciones de programación Android, programación orientada a objetos .
com.google.firebase.database	Vinculación con Firebase.
android.view.View;	Componente de aplicación Android.
android.widget.Button	Componente de aplicación Android.
android.widget.CompoundButton	Componente de aplicación Android.
android.widget.EditText	Componente de aplicación Android.
android.widget.ToggleButton	Componente de aplicación Android.

```

1 package com.iotiaua;
2
3 //-----
4 //LIBRERIAS PARA GRAFICAR
5 import lecho.lib.hellocharts.model.Axis;
6 import lecho.lib.hellocharts.model.AxisValue;
7 import lecho.lib.hellocharts.model.Line;
8 import lecho.lib.hellocharts.model.LineChartData;
9 import lecho.lib.hellocharts.model.PointValue;
10 import lecho.lib.hellocharts.model.Viewport;
11 import lecho.lib.hellocharts.view.LineChartView;
12 import android.graphics.Color;
13 import android.os.Bundle;
14 import java.util.ArrayList;
15 import java.util.List;
16 //FIN LIBRERIAS PARA GRAFICAR
17 //-----
18 import androidx.appcompat.app.AppCompatActivity;
19 import android.view.View;
20 import android.widget.Button;
21 import android.widget.CompoundButton;
22 import android.widget.EditText;
23 import android.widget.TextView;
24 import android.widget.ToggleButton;
25 import com.google.firebase.database.DataSnapshot;
26 import com.google.firebase.database.DatabaseError;
27 import com.google.firebase.database.DatabaseReference;
28 import com.google.firebase.database.FirebaseDatabase;
29 import com.google.firebase.database.ValueEventListener;
30

```

Figura 47. Importación de librerías en el IDE de Android Studio.

ANEXO 4. GPIO de la placa Raspberry utilizados en el Software.

Cuadro 6. GPIO utilizados en el sistema IOT.

GPIO de la Raspberry	Función
5V POWER	Alimentación de ADS1115, PH-4502C y módulo de Relevadores.
GROUND	Tierra de ADS1115, PH-4502C y módulo de Relevadores.
GPIO 2 (SDA)	Conexión con pin SDA de ADS1115.
GPIO 3 (SCL)	Conexión con pin SCL de ADS1115.

GPIO 21	Conexión con pin de módulo de relevadores para probar bombas.
GPIO 10	Conexión con pin de módulo de relevadores para probar bombas.
GPIO 16	Conexión con pin de módulo de relevadores para probar bombas.
GPIO 12	Conexión con pin de módulo de relevadores para probar bombas.
GPIO 26	Conexión con pin de módulo de relevadores para probar bombas.
