

UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS CARRERA DE INGENIERÍA EN MECATRÓNICA

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN MECATRÓNICA

TEMA:

*“Sistema de monitoreo y control de irrigación usando Internet de
las Cosas (IoT)”*

AUTOR

Román Alvarado Rogger Estalin

DIRECTOR

Ing. Milton Alejandro Gavilánez Villalobos, MSc.

IBARRA – ECUADOR

2022



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DEL AUTOR			
CÉDULA DE IDENTIDAD:	110555203-6		
APELLIDOS Y NOMBRES:	Román Alvarado Rogger Estalin		
DIRECCIÓN:	Macará – Loja – Ecuador		
EMAIL:	reromana@utn.edu.ec		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0985440424
DATOS DE LA OBRA			
TÍTULO:	“Sistema de monitoreo y control de irrigación usando Internet de las Cosas (IoT)”		
AUTOR:	Rogger Román Alvarado		
FECHA (DD/MM/AAAA):	03/02/2022		
SOLO PARA TRABAJOS DE GRADO			
PROGRAMA:	Pregrado		
TÍTULO POR EL QUE OPTA:	Ingeniero en Mecatrónica		
ASESOR/DIRECTOR:	Ing. Milton Alejandro Gavilánez Villalobos, Msc.		

CONSTANCIAS

El autor manifiesta que la obra, objeto de la presente autorización, es original y se la desarrolló sin violar derechos de autores de terceros, por lo tanto, la obra es original, y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 07 días del mes de febrero del 2022

A handwritten signature in blue ink that reads "Rogger Román". The signature is written in a cursive style and is positioned above a horizontal line.

Firma

Nombre: Rogger Román

Cédula: 1105552036



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERIA EN CIENCIAS APLICADAS

CERTIFICACIÓN

En calidad de director del trabajo de grado “Sistema de monitoreo y control de irrigación usando Internet de las Cosas (IoT)”, presentado por el egresado Rogger Estalin Román Alvarado, para optar por el título de Ingeniero en Mecatrónica, certifico que el mencionado proyecto fue realizado bajo mi dirección.

Ibarra, 25 de mayo del 2021

Ing. Milton Gavilanes Mg.

TUTOR DEL TRABAJO DE AGRADO

DEDICATORIA

A mi madre, Marcia Alvarado

A mi abuelita, Maura Alvarado

A mi hermana, Ariana Román

AGRADECIMIENTO

Agradezco primeramente a la persona que me cuidó, me guió y me apoyó toda la vida, mi madre Marcía Alvarado. Gracias a ella pude tener una educación de calidad, y el tiempo suficiente para desarrollar mi pasión por las ciencias. Con su amor incondicional, he podido salir adelante.

A Ariana Román por impulsarme, ser mi guía cuando más lo necesitaba, por ser mi inspiración y entregarme ese amor de hermanos que solo alguien que siempre ha estado contigo te puede brindar. A mi mamá Maura Alvarado por ser la mejor abuelita del mundo, por ser una verdadera guerrera y siempre brindarnos lo que está a su alcance. A mi Padre, mis tíos, familiares, amigos y docentes que con su cariño y apoyo he logrado este título.

A mi tutor Milton por su acompañamiento, comprensión y por confiar en mí para el desarrollo de este trabajo de grado. A mis cotutores Luz María e Iván por compartir sus conocimientos, cultivar la curiosidad del extenso e interesante mundo de la mecatrónica y por brindarme de su tiempo.

Resumen

La presente tesis “Sistema de monitoreo y control de irrigación usando Internet de las Cosas (IoT)”, es fruto del resultado de encuestas realizadas a pequeños productores de frutas y hortalizas, en el cual se evidencia que el principal proceso de automatización es la irrigación de sus cultivos, esto junto con la necesidad de poder realizar el monitoreo y control de irrigación remotamente, se planteó lograr el diseño y construcción de un prototipo que cubra esta necesidad. Por lo cual se construyó un sistema de riego inteligente que permite el monitoreo y control utilizando internet de las cosas.

La metodología que se utilizó fue seleccionar una finca para su implementación, por lo que el proyecto se realizó en base con los requisitos y restricciones brindadas por el propietario del lugar, luego se procedió a diseñar el sistema con la selección de la tecnología (WiFi), protocolos de comunicación (MQTT), hardware (NodeMCU 8266, ESP32 Devkit, etc) y software (IDE de Arduino, C++, Ubidots) que mejor se adaptan al proyecto. Haciendo uso de la plataforma de Ubidots se diseñó y se desarrolló una interfaz de usuario amigable con el usuario. Consiguiendo la construcción del sistema de monitoreo y control de irrigación usando Internet de las Cosas, para su respectiva instalación en la finca seleccionada.

El sistema funciona correctamente logrando un sistema de riego automático, el cual recibe los valores de humedad del suelo mínima y máxima permitidas, y con la ayuda de un sensor se activa o desactiva el riego, alertando mediante correo electrónico en caso de algún fallo. Finalmente, el monitoreo y control de irrigación pueden realizarse desde cualquier lugar en el mundo con acceso a internet.

Abstract

This thesis "Irrigation monitoring and control system using the Internet of Things (IoT)", is the result of surveys conducted to small fruit and vegetable producers, in which it is evident that the main automation process is the irrigation of their crops, along with the need to be able to monitor and irrigation control remotely, it was proposed to achieve the design and construction of a prototype that meets this need. Therefore, an intelligent irrigation system was built that allows monitoring and control using the internet of things.

The methodology used was to select a farm for its implementation, so the project was carried out based on the requirements and restrictions provided by the owner of the place, then proceeded to design the system with the selection of technology (WiFi), communication protocols (MQTT), hardware (NodeMCU 8266, ESP32 Devkit, etc) and software (Arduino IDE, C++, Ubidots) that best fit the project. Making use of the Ubidots platform, a user-friendly user interface was designed and developed. Achieving the construction of the irrigation monitoring and control system using the Internet of Things, for its respective installation in the selected farm.

The system works correctly achieving an automatic irrigation system, which receives the minimum and maximum soil moisture values allowed, and with the help of a sensor activates or deactivates irrigation, alerting by email in case of any failure. Finally, irrigation monitoring and control can be done from anywhere in the world with internet access.

Índice General

Aspectos Generales	1
1. Problema	1
2. Objetivos	2
2.1. Objetivo General	2
2.2. Objetivos específicos	2
3. Alcance	3
4. Justificación	3
Capítulo I	5
1. Marco teórico	5
1.1. Mecatrónica	5
1.2. Agricultura	7
1.2.1. Agricultura moderna	8
1.3. Uso del agua en la agricultura	9
1.4. Parámetros que influyen en un riego adecuado	10
1.5. Sistemas de riego automático	11
1.6. Internet de las cosas	12
1.7. Tecnologías de comunicación para IoT	14
1.7.1. Cableada ethernet	14
1.7.2. Comunicaciones inalámbricas	15
1.7.2.1. WIFI	16
1.7.2.2. Bluetooth	17

1.7.2.3. Redes celulares	18
1.7.2.3.1. 2G	18
1.7.2.3.2. 3G	18
1.7.2.3.3. LTE, 4G y 5G	19
1.7.2.4. LoRa / LoRaWAN	20
1.7.2.5. ZIFBEE	21
1.7.2.6. SigFox	21
1.8. Protocolos WIFI	21
1.8.1. Protocolo 802.11b	22
1.8.2. Protocolo 802.11a	23
1.8.3. Protocolo 802.11g	23
1.8.4. Protocolo 802.11n	23
1.8.5. Protocolo IEEE 802.11ah	24
1.8.6. Protocolo IEEE 802.11ax	25
1.9. Tipos de seguridad de conexiones WIFI	25
1.9.1. WEP (Wired Equivalent Privacy)	25
1.9.2. WPA (Wi-Fi Protected Access)	25
1.9.3. WPA2	26
1.9.4. WPA3	26
1.10. Tecnología LoRa y protocolo LoRaWAN	26
1.11. Protocolos de comunicación IoT	30

1.12.	HTTP (REST/JSON)	30
1.13.	MQTT.....	31
1.14.	CoAP.....	31
Capítulo II.....	32	
2.1.	METODOLOGÍA.....	32
2.2.	Requisitos del prototipo para sistemas de irrigación piloto	33
2.2.1.	Requerimientos.....	34
2.2.2.	Restricciones	34
2.2.3.	Deseables	35
2.3.	Análisis del sitio de trabajo.....	35
2.4.	Tipo de control.....	35
2.5.	Selección de elementos.....	37
2.5.1.	Selección de hardware	39
2.5.1.1.	Módulo principal	40
2.5.1.1.1.	Node MCU.....	40
2.5.1.1.2.	Electroválvula solenoide.....	41
2.5.1.1.3.	Sensor de flujo YF-S201.....	42
2.5.1.2.	Módulo sensor.....	42
2.5.1.2.1.	ESP32 Devkit.....	43
2.5.1.2.2.	Sensor de humedad YL-69.....	43
2.5.2.	Selección de software	44

2.5.3. Diseño de la interfaz.....	47
Capítulo III	52
3. Construcción e implementación	52
3.1. Desarrollo del prototipo	52
3.2. Construcción del prototipo.....	59
3.3. Implementación del prototipo.....	61
3.4. Validación	61
Capítulo IV	64
4. Conclusiones y recomendaciones	64
4.1. Conclusiones	64
4.2. Recomendaciones	65
Bibliografía	66
Linkografía.....	68
Anexos	69
Anexo 1. Programación del módulo principal o de control	69
Anexo 2. Programación del módulo sensor	78
Anexo 3. Instalación.....	83
Apéndices	87
Apéndice A. Datasheet de NodeMCU ESP8266	87
Apéndice C. Datasheet 2n2222	90

Índice de tablas

Contenido	Pág.
Tabla 1 Características de las tecnologías para IoT	38
Tabla 2 Selección de la tecnología.....	38
Tabla 3 Protocolos de comunicación para IoT.....	39
Tabla 4 Comparación de plataformas IoT de pago	45

Índice de figuras

Contenido	Pág.
Figura 1. La mecatrónica como intersección de muchas áreas (Bishop, 2018).....	6
Figura 2. Flujo de trabajo de un sistema mecatrónico (Bishop, 2018)	6
Figura 3. Tipos de comunicación IoT (Lopez, 2020)	14
Figura 4. Sistema de control en lazo cerrado(<i>Ogata, 2010</i>)	36
Figura 5. Diagrama de flujo del prototipo (Autoría propia)	37
Figura 6. Tarjeta NodeMCU 8266 y su configuración de pines (Electronilab, 2020).....	41
Figura 7. Electroválvula solenoide 12V (Valparaíso, n.d.).....	41
Figura 8. Sensor de flujo de agua de 1/2" YF-S201 (ElectronicsComp, 2021)	42
Figura 9. ESP32 Devkit y su configuración de pines (Klimov, 2020).....	43
Figura 10. Sensor de humedad YL-69 (Caldas, 2021).....	44
Figura 11. Diseño de la interfaz del proyecto (Autoría propia)	47
Figura 12. Sección de control (Autoría propia)	48
Figura 13. Sección de visualización (Autoría propia).....	48
Figura 14. Evolución de los datos en el tiempo (Autoría propia)	49
Figura 15. Sección de los datos o puntos recibidos en el bróker (Autoría propia)	50
Figura 16. Visualización de datos de sensores (Autoría propia).....	50
Figura 17. Circuito acoplador para relé de 5V (Alex, 2020)	54
Figura 18. Visualización del proyecto en Ubidots (Autoría propia).....	55
Figura 19. Variables del proyecto (Autoría propia).....	55
Figura 20. Configuración de Widgets (Autoría propia).....	56
Figura 21. Configuración de la visualización de los datos (Autoría propia)	56
Figura 22. Máximos y mínimos de la función de riego. (Autoría propia)	58
Figura 23. Mínimos y máximos de las alertas (Autoría propia)	58

Figura 24. Diagrama de conexión del módulo sensor (Autoría propia).....	60
Figura 25. Conexión de módulo de control (Autoría propia)	61
Figura 26. Sistema de monitoreo y control de irrigación.....	62
Figura 27. Validación de la humedad del suelo	62
Figura 28. Derivación del suministro de agua	83
Figura 29. Instalación del módulo de control	83
Figura 30. Instalación de la electroválvula y sensor de flujo.....	84
Figura 31. Instalación del sistema para riego por goteo	84
Figura 32. Módulo sensor	85
Figura 33. Sensor de humedad junto a los goteros	85
Figura 34. Sistema automático de monitoreo y control de irrigación con IoT	86

Introducción

Aspectos Generales

1. Problema

Debido al cambio climático mundialmente conocido y al inminente crecimiento poblacional, el agua dulce es cada vez un recurso más apreciado. Por lo que el uso eficiente del agua y la mejor producción tienen un alto impacto social, y económico. Con la ayuda de la tecnología se puede lograr conseguir estos objetivos.

A nivel internacional el mayor consumo de agua, aproximadamente el 70% es utilizado para la irrigación.(Vermeulen et al., 2012). Según las encuestas realizadas en la materia de probabilidad y estadística (Conocer las necesidades prioritarias de automatización, febrero, 2019) de la Universidad Técnica del Norte, realizada en las zonas externas de la ciudad de Ibarra a los agricultores. Se puede evidenciar que la principal tarea de automatización que se desea alcanzar es el control de la irrigación ya sea automática o remota. Según diversos estudios que se han realizado se puede evidenciar que una de las mayores preocupaciones por parte de los que trabajan la tierra es que sus cultivos perezcan por falta del líquido vital (estrés hídrico) por lo que generalmente suelen hacer un riego abundante o excesivo (saturado) malgastando los recursos hídricos lo que provoca una disminución en la cantidad y calidad final de su cosecha.(Pereira et al., 2002)(Jensen et al., 1990)

Los agricultores con su trabajo suelen buscar la mejor calidad, mayor cantidad y mayor tamaño en sus productos debido a que aquello se traduce en mayores ganancias. Teniendo en cuenta la cantidad de área que tiene que trabajar o irrigar a diario suelen contratar personal, otra de sus limitaciones es que sus terrenos son alejados de su lugar de residencia por lo que en su defecto tienen que a diario ir al lugar.

Por lo que se busca un sistema que sea confiable y a la vez cumpla con sus requerimientos. ¿Por qué? Porque el contratar personal o ir a diario a las parcelas a irrigar les provoca un costo de producción adicional o pérdida de tiempo por cumplir dicha tarea.(Jones, 2004)

Los sistemas de riego inteligente permiten obtener una reducción en el costo de producción de los agricultores, un ahorro de tiempo en su diario vivir, consecuentemente un monitoreo y control a distancia. La automatización llegó para quedarse. Tiene un amplio rango de aplicación entre ellas mejorar la producción, automatizar procesos repetitivos, poder monitorear y controlar desde cualquier lugar del mundo que tenga acceso a internet con el uso de tecnología IOT. Por lo que con el sistema de monitoreo y control de irrigación con internet de las cosas permite al agricultor tener mejor rendimiento de sus finanzas y mejor calidad en su producción.(Rajalakshmi & Mahalakshmi, 2016)

2. Objetivos

2.1. Objetivo General

- Construir un sistema de riego inteligente que permite el monitoreo y control utilizando internet de las cosas (IOT).

2.2. Objetivos específicos

- Identificar el método adecuado para la irrigación al aire libre con el uso de tecnología de internet de las cosas, tomando en cuenta la coyuntura por la pandemia.
- Diseñar el sistema que cumplirá con los requerimientos generales de irrigación, para satisfacer la necesidad del riego automático.

- Implementar el prototipo de irrigación con la metodología adecuada.
- Evaluar el sistema automático de riego con tecnología IOT en un espacio piloto.

3. Alcance

El presente trabajo va a diseñar e implementar un prototipo de riego automático con tecnología del internet de las cosas en un sitio experimental para la prueba de funcionamiento, el usuario tiene un programa de riego automático preestablecido, con la ayuda de un sensor de humedad de suelo, el beneficiario puede obtener información de la humedad del suelo del lugar y si desea activar el riego manualmente desde cualquier lugar conectado a internet. El dispositivo funciona con energía eléctrica cercana al sitio de trabajo.

4. Justificación

En la era moderna se sigue realizando tareas que se han venido ejecutando desde el inicio del sedentarismo, como por ejemplo la labor del riego. La gran ventaja es que ahora con ayuda de la ciencia conocemos: Como se desarrollan las plantas con gran exactitud, que necesitan para un buen desarrollo, en que tiempo y en qué medida. Por lo que podemos hacer un uso adecuado del recurso hídrico más importante para el ser humano como es el agua dulce.(Lennox et al., 2019)

La automatización de procesos como la irrigación trae múltiples beneficios para el agricultor en la era actual, y un sistema de irrigación inteligente trae consigo ventajas sociales, ambientales y económicas. El uso preciso del agua provoca que las plantas produzcan mayor cantidad y mejor calidad de alimentos, asimismo, permite hacer uso eficiente de agua en los lugares con poca capacidad hídrica, permitiendo mayores áreas de cultivos.(Jones, 2004)

La presente investigación además de resolver los problemas mencionados hace uso de la tecnología IOT lo cual le permite poder tener el control de la irrigación desde cualquier lugar del mundo con acceso a internet, por lo cual las personas que tengan una finca lejos de su lugar residencial pueden tener un control automático o manual desde la comodidad de sus casas. En las encuestas realizadas por la Universidad Técnica del Norte se detectó que la principal área de automatización que requiere la agroindustria es la de irrigación.(García et al., 2020)

Capítulo I

MARCO TEORICO Y REFERENCIAL

1. Marco teórico

1.1. Mecatrónica.

Con el tiempo, la mecatrónica ha venido a representar sistemas multidisciplinarios. Los sistemas mecatrónicos comprenden muchos dominios, como mecánico, eléctrico, hidráulico, electrónico, etc. Los componentes de estos diferentes dominios trabajan juntos de manera sinérgica junto con microcontroladores o microcomputadoras que controlan o toman decisiones sobre el comportamiento del sistema.

La definición más comúnmente utilizada para un sistema mecatrónico es: "Combinación sinérgica de ingeniería mecánica de precisión, control electrónico y software inteligente en un marco de sistemas, utilizado en el diseño de productos y procesos de fabricación".(Das, 2020). Otra definición que da una mejor idea es: "El campo interdisciplinario de la ingeniería que se ocupa del diseño de productos cuya función se basa en la integración de componentes mecánicos y electrónicos coordinados por una arquitectura de control". (De Silva, 2010)

Todas estas declaraciones transmiten conceptos similares sobre el término mecatrónica. El esquema de la Figura 1. muestra una variedad de temas técnicos y cómo se superponen.

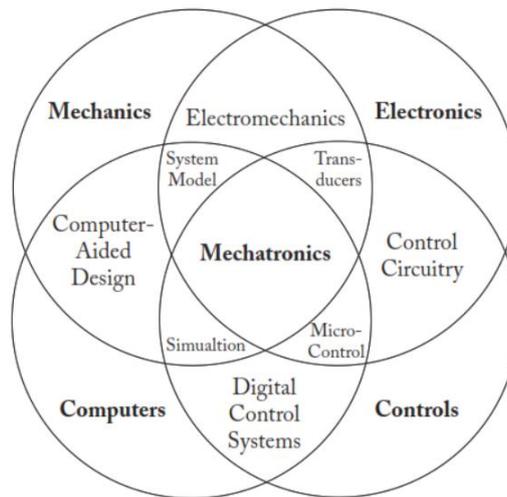


Figura 1. La mecatrónica como intersección de muchas áreas (Bishop, 2018)

Como se aprecia en la Figura 1. la mecatrónica se puede constituir como la sinergia de algunas áreas, esta da una ventaja a la hora de diseñar un sistema o producto mecatrónico, se elimina la idea de diseño separado y se converge en un diseño general e iterativo. En la Figura 2. Se presenta un ejemplo del flujo de trabajo para un sistema mecatrónico.

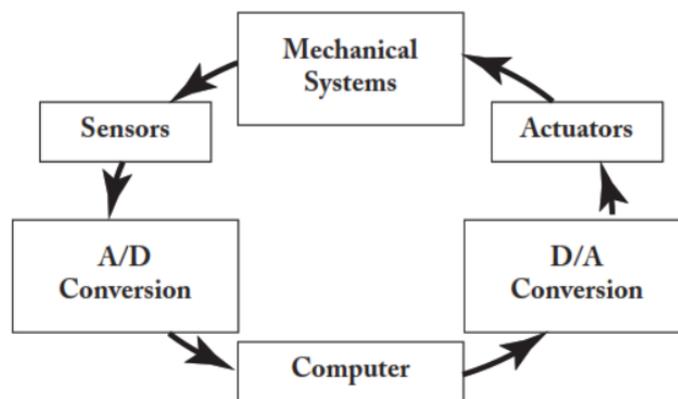


Figura 2. Flujo de trabajo de un sistema mecatrónico (Bishop, 2018)

De la variedad de conceptos que existen para un “Sistema” se ha discernido y para la realización de esta tesis se tomará esta definición: “Un sistema puede definirse como una entidad que es separable del resto del universo a través de límites físicos y/o conceptuales.”(Karnopp et al., 2012)

1.2. Agricultura.

La agricultura viene del latín agri = ‘campo’ y cultura = ‘cultivo’, ‘crianza’, se denomina agricultura al conjunto de técnicas y actividades económicas relacionadas con el cultivo de la tierra y el tratamiento del suelo para la producción de alimentos. Se podría definir como las actividades que abarcan la explotación del recurso que la tierra origina y la ciencia que estudia la práctica de la agricultura es la agronomía.

Según la Academia Internacional de Ingeniería de Estados Unidos, se considera que la mecanización agraria es uno de los 20 mayores logros de la ingeniería del siglo XX, pasando de la cifra donde un granjero podía alimentar de 2 a 5 personas a la cifra de 130 personas por cada granjero.(Vermeulen et al., 2012)

La agricultura puede dividirse según muchos criterios distintos de clasificación, como, por ejemplo:

Según su dependencia del agua:

- De secano: es la agricultura producida como su nombre lo indica sin aporte de agua por parte del mismo agricultor. Siendo la lluvia o aguas subterráneas las que lo nutren al suelo.
- De regadío: como su nombre lo indica el agricultor es el que produce con el aporte de agua, mediante el suministro que se capta de fuentes de agua superficiales naturales o artificiales, o mediante la extracción de aguas subterráneas de los pozos.

Obteniendo otras clasificaciones como:

- Agricultura de subsistencia
- Agricultura industrial
- Agricultura intensiva
- Agricultura extensiva

- Agricultura tradicional
- Agricultura orgánica
- Agricultura natural
- Agricultura moderna

De todas las clasificaciones y las divisiones que puede tener la agricultura, las que interesan son la que depende del agua denominada “de Regadío” y la denominada “Agricultura Moderna” la cual se estudia a continuación más detalladamente.

1.2.1. Agricultura moderna

La Agricultura Moderna es un término que describe las prácticas agrícolas y las innovaciones que están contribuyendo directamente a reducir el uso de recursos naturales (agua, tierra y energía) con la ayuda de tecnología.

La utilización de tecnología y datos permite usar menos recursos, ayudando a los productores a tomar decisiones sobre sus operaciones. Las nuevas herramientas digitales están ayudando a los agricultores a utilizar cantidades más exactas de agua y de otros recursos, estas aplicaciones brindan información a los productores sobre el clima, la siembra y el suelo para ayudarlos a que sus cultivos prosperen, pueden explorar sus campos de forma remota, plantar más semillas en lugares donde tienen mayores probabilidades de prosperar y resolver problemas antes de que lleguen al punto no deseado de reducir la producción del cultivo.

En los últimos años la agricultura moderna ha avanzado en la adopción de nuevas tecnologías de forma explosiva. Como resultado, se están logrando capacidades y rendimientos de producción muy superiores a las que podían esperarse hace no muchos años. Los principales

avances se deben a la irrupción en el mercado de material genético de gran productividad, a la incorporación de maquinaria muy eficiente y a la alta adopción de eventos biotecnológicos.

Finalmente se puede definir la agricultura de precisión o moderna como el conjunto de técnicas orientadas a optimizar el uso de los insumos agrícolas (agua, semillas, agroquímicos y correctivos) en función de la cuantificación de la variabilidad espacial y temporal de la producción agrícola. Esta optimización se logra con la distribución de la cantidad correcta de esos insumos, dependiendo del potencial y de la necesidad de cada punto de las áreas de manejo. (Bongiovanni et al., 2006)

1.3. Uso del agua en la agricultura.

Del total de la superficie del planeta tierra el 25 % es suelo(tierra), y de uso de la agricultura solo es el 3 %, si bien la mayor parte de la superficie es agua, esta es salada y no es apta para el consumo humano ni para irrigar los cultivos, del total de agua dulce que se obtiene por diferentes formas, el 70 % se utiliza para la agricultura. Y se estima que la mitad de toda el agua usada para el riego es ineficiente.

En los últimos años la comunidad científica ha llegado al consenso que estamos viviendo un cambio climático, entre los efectos de este cambio climático está que se reducirá el agua superficial disponible en regiones secas subtropicales, se intensificará la competencia por el uso de agua, muy probablemente aumentarán las sequías lo que ocasionará un menor regadío en los cultivos y los rendimientos pueden decrecer en algunas zonas y crecer en otras.

Todos estos problemas se podrán evitar si a nivel mundial se realiza las acciones oportunas para revertir esta situación, pero también se puede aprovechar esta crisis para una

modernización del sector, donde al implementar la tecnología a disposición se logrará un uso eficiente del agua y otros recursos, como es el objetivo de la presente tesis.

1.4. Parámetros que influyen en un riego adecuado.

Los parámetros físicos que determinan el resultado de un evento de riego pueden agruparse en: (1) variables del sistema (parámetros físicos cuya magnitud puede cambiar en un amplio rango en función de la decisión del agricultor o persona encargada) y (2) parámetros del sistema, son parámetros físicos propios del sistema y que poco o nada de margen de cambio tienen.

Como parámetros de los sistemas de riego se considera: la lámina de riego a aplicar (d_n), velocidad máxima del agua admitida para evitar erosión, pendiente del terreno (S), coeficiente de rugosidad (n), parámetros de la ecuación de infiltración (A , B , f_0), geometría de surcos. Las variables del sistema son: caudal de manejo y/o caudal unitario (Q_m , q_u), tiempo de aplicación o corte (T_{ap}), y en menor medida pero importante si hablamos de grandes distancias como, la longitud de la unidad de riego (L).

La evaluación de los sistemas de riego al campo juega un papel fundamental para la optimización de estos, brindando valiosa información para la mejora del diseño y prácticas de manejo.

De todos los parámetros físico que pueden obtenerse para el riego de un área en específico, vamos a trabajar con el que más influye en el momento de riego de una planta sin considerar de qué forma circula el agua(canales) pero especificando el tipo de riego empleado, en este caso riego por goteo.

La variable que se considera para realizar un riego según las suposiciones planteadas es la humedad del suelo, porque la cantidad de humedad que se evapora o se pierde con el sol es necesario que la planta recupere. Todo depende de cómo se quiere recuperar la lámina de riego. Por ejemplo, si hoy perdimos 3 milímetros de agua mañana se le proporciona esos 3 milímetros de agua, es decir lo que consumió hoy lo recupero mañana. El riego no se lo aplica a la planta sino al suelo, y debe darse cuenta de la cantidad de agua que va perdiendo por consumo de sus plantas. Se mide la cantidad de humedad que tiene el suelo para que la planta tome el agua sin la necesidad de estresarse. Y quien mide esto puede ser un sensor de humedad del suelo o un tensiómetro.

1.5. Sistemas de riego automático

Los sistemas de riego que se encuentran en el mercado nacional por lo general suelen ser sistemas de riego “automático” simples, basando su funcionamiento en abrir una válvula por un tiempo y luego cerrarla, por lo que son sistemas de control en lazo abierto. Nuestra tesis propone un sistema de control que tome en cuenta la necesidad de la planta para suministrar el líquido vital, abriendo o cerrando la válvula.

Existen sistemas de riego automático con retroalimentación, pero un sistema que se puede gestionar desde cualquier lugar del mundo con internet no hay, no existen proyectos comerciales. Y los que dicen que lo pueden hacer no tienen un producto en específico, sino que ofrecen más bien un servicio donde después de la visita técnica te recomiendan que productos usar y las variantes que todo ello implica.

Por lo que se ha investigado de este tipo de empresas, después de un largo análisis se concluye que la realización de este tipo de sistemas es muy complejo, ya que tiene que disponer

de la información necesaria del lugar, la cual es imprescindible para poder determinar el grupo de bombeo necesario, el cabezal de filtrado a emplear, el tipo de gotero a utilizar, el marco de plantación, así como la realización del diseño y de los cálculos necesarios, la medición del terreno con el equipo apropiado a fin de obtener el plano acotado con las curvas de nivel, un estudio de la calidad del agua de suministro, las condiciones climáticas del lugar, etc.

Entre las ventajas están:

- Ahorro de mano de obra, agua y energía.
- Mayor eficiencia de riego.
- Aumentos de la producción debido a la optimización del riego.

La automatización del riego puede programarse, según el criterio del agricultor como, por ejemplo:

- Programación por tiempos.
- Programación por volúmenes medidos por un medidor de caudal o detectores de nivel en balsas pequeñas.
- Programación en base a sensores del estado hídrico del suelo y de la planta.
- Programación basada en datos meteorológicos proporcionados por una estación meteorológica.

1.6. Internet de las cosas

El internet de las cosas, “IoT” o internet of things, son los dispositivos que se conectan a internet y pueden interactuar tanto en el mundo digital como en el mundo físico. Su objetivo es mejorar la vida de los humanos realizando acciones, ya sean informativas o activas, donde por ejemplo un dispositivo del internet de las cosas puede avisarte si alguien está intentando

entrar a tu casa, sin importar donde te encuentres siempre y cuando tengas acceso a internet. O por ejemplo desde tu oficina activar una aspiradora inteligente para que limpie antes de que tu llegues a casa.

El internet de las cosas hace referencia a una tecnología basada en la conexión de objetos cotidianos a Internet que pueden intercambiar, procesar y agregar información sobre su entorno físico para así dar servicios de valor añadido a los usuarios finales. También reconoce cambios o eventos, ya que tales sistemas pueden reaccionar de forma automática y adecuada. Su finalidad es superar la barrera entre los objetos en el mundo físico y su representación en los sistemas de información.

El Internet de las cosas tiene sus raíces en el trabajo que realizó Mark Weiser para Xerox PARC en la década de 1990. Weiser no estudió cómo serían las redes que conectarían las redes entre sí, sino qué ocurriría cuando la tecnología fuese tan barata que se pudiese incorporar a objetos cotidianos. Entonces acuñó el término Computación ubicua, o ubicomp, también conocida como “tecnología ambiental” (aunque conlleva otras connotaciones que pasan a segundo plano) o “tecnología sosegada”: sistemas que no necesitan nuestra atención para funcionar y que están listos para proporcionar información útil cuando la solicitamos. [1]

Con la ayuda del internet de las cosas los agricultores pueden monitorear las condiciones del campo desde cualquier lugar. La agricultura basada en IoT es altamente eficiente en comparación con la tradicional. En términos de cuestiones ambientales la agricultura basada en IoT puede proporcionar grandes beneficios, como, por ejemplo, la optimización de insumos y tratamientos y por supuesto, un uso más eficiente del agua.

1.7. Tecnologías de comunicación para IoT

Las tecnologías de comunicación para el IoT se están desarrollando rápidamente alrededor del mundo, sin embargo, es una tecnología relativamente nueva. Se pueden considerar actas todas las tecnologías que tienen acceso a la red o las que hacen de intermediarias. Teniendo una clasificación por el medio de comunicación físico como son las conexiones cableadas como el ethernet en todas sus variantes, y las inalámbricas que tienen diferentes clasificaciones.

Mientras que las comunicaciones inalámbricas se pueden subclasificar por la proximidad o distancia de funcionamiento correcto. Como se ve en la figura 3.

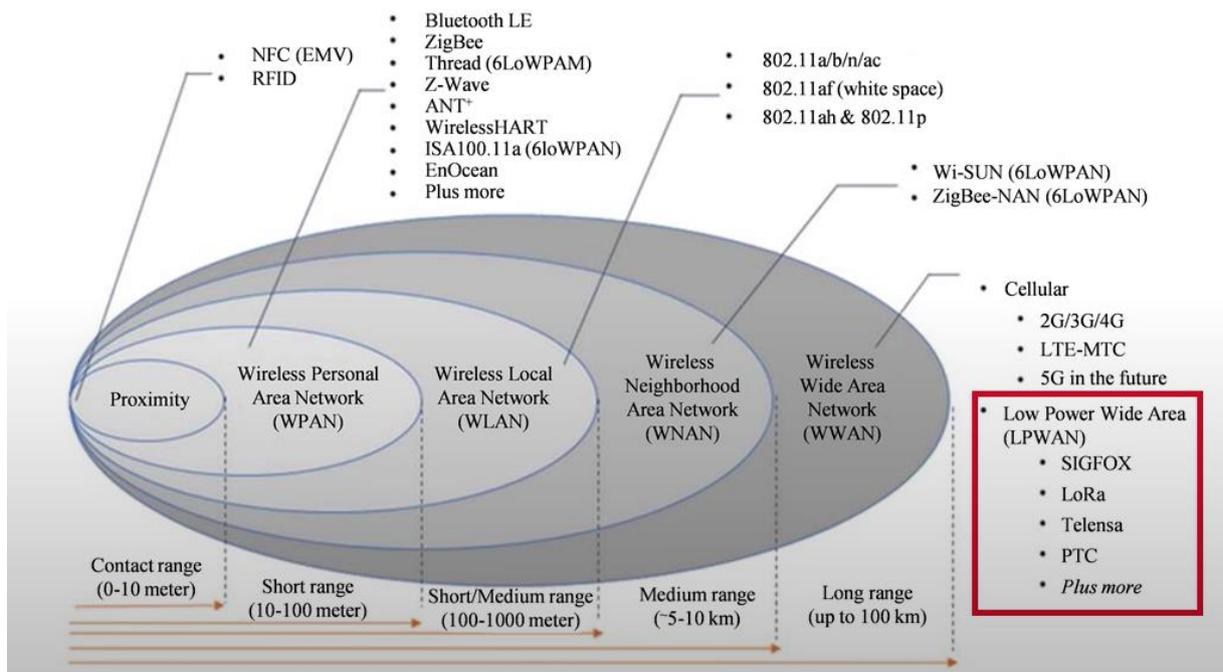


Figura 3. Tipos de comunicación IoT (Lopez, 2020)

1.7.1. Cableada ethernet

Es la más antigua, la más confiable, se tomó como base para la redacción del estándar internacional IEEE 802.3, es un estándar conocido en todo el mundo y tiene pros y contras.

Ventajas:

- Estándar abierto
- Disponibilidad de conexiones
- Alta velocidad (hasta 1 Gbps)
- Sin interferencias
- 100 m de alcance
- Bajo costo
- Soporta alimentación a través de PoE

Desventajas:

- Hay que cablear
- Alto consumo (200/300 mA)
- No incluye un mecanismo de seguridad
- Conector de gran tamaño

Usos:

- Cámaras de seguridad
- Equipamiento industrial (PLCs)
- UPS
- Computadoras compactas
- Equipamiento médico
- Microcontroladores
- Smart TVs, consolas de juegos, etc.

1.7.2. Comunicaciones inalámbricas

Las comunicaciones inalámbricas son caracterizadas por transmitir la información por ondas electromagnéticas por medio del aire, por lo que no necesitan ser cableadas para ni para transmitir ni para recibir. Este tipo de tecnología tienen limitaciones, tienen característicamente 3 tipos de características que deben de seleccionarse de acuerdo con los requisitos de la aplicación.

Estos son el consumo eléctrico, alcance y bitrate. Por lo que si ganamos en una podemos perder en otra, si deseamos alto alcance y bajo consumo debemos de menorar el bitrate que es la cantidad de bits que se transmiten. Si queremos alto bitrate y bajo consumo, el alcance no será mucho.

1.7.2.1. WIFI

Características

- Tecnología WLAN
- Alto bitrate (hasta 1.3 Gbps)
- Bandas en 2.4 GHz y 5GHz
- Estándares 802.11 b/g/n/ac, etc.
- Encriptación

Ventajas:

- Estándares abiertos
- Alto bitrate
- Buen alcance (50 m en interiores y 100 m en exteriores)
- Disponibilidad
- Seguridad incluida

- Chips de bajo costo
- Acceso directo a la red IP

Desventajas

- Requiere autenticación
- Alto consumo
- Interrupciones en la comunicación
- Certificación Wireless
- Interferencia electromagnética

1.7.2.2. Bluetooth

Características

- Dos opciones: Clásico y BLE (propio para dispositivos IoT)
- Banda en 2.4 GHz
- Alcance entre 10-20 m
- Bitrate clásico 2 Mbps
- Bitrate BLE < 1 Mbps

Ventajas

- Bajo consumo
- Soportado por muchos dispositivos
- Alcance hasta de 20 m
- Acople rápido
- Modo Beacon

- Algunos chips permiten multipunto y mesh

Desventajas

- Hay que agregar seguridad
- Interferencias electromagnéticas
- No es IP

1.7.2.3. Redes celulares

Características

Estas redes suelen tener largo alcance, son las que permiten las telecomunicaciones por el celular. Varias tecnologías disponibles a lo largo del tiempo, ha tenido un gran desarrollo y adopción por lo que se encuentran en todo el mundo y tienen disponibilidad global.

1.7.2.3.1. 2G

También llamada GPRS todavía disponible en algunos países en desarrollo donde no se han actualizado, ofrece hasta 200 Kbits/s, debido al bajo bitrate tiene bajo consumo energético. Sus chips son cuatribanda lo que significa que pueden recibir y enviar información por diferentes bandas.

Son económicos, fueron de las primeras generaciones en telefonía celular y aunque son funcionales no se recomienda para nuevos proyectos porque el desarrollo permite mejoras considerables.

1.7.2.3.2. 3G

Gran parte de Sudamérica todavía cuenta con este tipo de redes en ciudades poco pobladas o sus exteriores, por lo que son muy disponibles, cuentan con un bitrate de 1-3 Mbits/s, con un

consumo moderado de energía, también son relativamente económicos dependiendo del país y las bandas también son asignadas según la zona y el proveedor.

1.7.2.3.3. LTE, 4G y 5G

Son lo último en tecnología para celulares implementadas, en algunos países se están implementando la tecnología 5 G que permiten velocidades impresionantes 10 Gbits/s sin embargo son pocos los países y ciudades en el mundo que cuentan con la implementación, las tecnologías de LTE y 4 G si están implementadas en Latinoamérica cuenta con diferentes categorías de 0 – 9, tienen mayor consumo que las redes 3 G pero existen categorías para IoT con muy bajos consumos, además las bandas igualmente son asignadas por la zona y el proveedor.

Ventajas

- Disponibilidad
- Conexión IP
- SMS y voz integradas
- Red confiable
- Seguridad integrada
- Geolocalización

Desventajas

- Módulos costosos
- Alto consumo dependiendo de la solución
- Diferentes bandas por zonas
- Alto costo mensual por MB
- Tarifa mensual

1.7.2.4. LoRa / LoRaWAN

Es una tecnología para el IoT desarrollada por Semtech y está pensada para comunicaciones de largo alcance, muy bajo consumo y bajo bitrate. Cuenta con diferentes bandas anchas según el país, estas pueden ser de 315, 433, 868 y 900 MHz. Funciona con conexiones a redes ip mediante gateways (LoRaWAN).

Según (Mayobre, 2018) LoRa es un protocolo de capa de enlace y es muy útil para usarse en comunicaciones P2P. Es un protocolo de red que está diseñado para optimizar dispositivos alimentados por baterías, que pueden estar posicionados en lugares fijos o pueden ser móviles. Permite variar dinámicamente el bitrate de los equipos que son parte de la red, desde valores de 0,3 Kbps hasta 50 Kbps, dependiendo del chip utilizado y parámetros de modulación.

Ventajas

- Largo alcance (40 km en campo abierto)
- Uso libre
- Hasta 50 Kbps
- Muy bajo consumo
- Inmunidad a la interferencia
- Aprovecha rebotes de señal
- Chips económicos

Desventajas

- Administración de la red
- No IP
- Protocolo propietario

1.7.2.5. ZIFBEE

ZigBee es una tecnología inalámbrica, que no es red de comunicaciones, muy utilizada desde hace años y centrada en aplicaciones domóticas e industriales. Actualmente los perfiles ZigBee PRO y ZigBee Remote Control (RF4CE) cumplen con las especificaciones de tasas de envío de datos bajas, pero con un alcance de cobertura que ronda los 100 metros. Por este motivo se puede descartar las comunicaciones por ZigBee en caso de proyectos donde los dispositivos a comunicar se encuentren muy alejados del modem o concentrador de datos.

Por otro lado, también podemos recordar algunas ventajas para aquellos que se decanten por una comunicación ZigBee en los proyectos de IoT: bajo consumo, seguridad superior al resto de tecnologías, robustez, alta escalabilidad y capacidad para soportar un gran número de nodos.

1.7.2.6. SigFox

Sigfox es la red de comunicaciones LPWAN específica para IoT más extendida a nivel mundial, con una cobertura gran cobertura en el territorio europeo mas no está adecuadamente implementada en el americano. La red de Sigfox está construida sobre una modulación UNB (ultra narrow band) y opera en la banda de 902MHz en Estados Unidos y generalmente a 868MHz en Europa. Es compatible con Azure de Microsoft, lo que acelera en gran medida la ejecución de un proyecto de IoT.

1.8. Protocolos WIFI

Wi-Fi es un conjunto de especificaciones para las redes de área local inalámbricas (WLAN), basadas en el estándar IEEE 802.11. El nombre de “Wi-Fi” es obtenido como una abreviatura del término inglés “Wireless Fidelity”.

Vamos a conocer las versiones más importantes del 802.11, pero antes, para facilitar la comprensión, es conveniente saber que, para una red de este tipo sea establecida, es necesario que los dispositivos (también llamados STA) se conecten a los aparatos que faciliten el acceso. Estos son los genéricamente denominados Access Point (AP). Cuando uno o más de STAs se conectan a un AP, se tiene, por tanto, una red, que es denominada Basic Service Set (BSS).

Por cuestiones de seguridad y por la posibilidad de que haya más de un BSS en un lugar determinado (por ejemplo, dos redes wireless que fueron creadas por diferentes empresas en un área de eventos), es clave que cada uno reciba una identificación llamada Service Set Identifier (SSID), un conjunto de caracteres que, después de definido, se inserta en el encabezado de cada paquete de datos de la red. En otras palabras, el SSID es el nombre dado a cada red inalámbrica.

1.8.1. Protocolo 802.11b

En 1999 nace la publicación de la actualización del estándar 802.11 nombrándolo 802.11b. La principal característica de esta versión es su método de acceso definido en el estándar original CSMA/CA. El estándar 802.11b funciona en la banda de 2,4 GHz y su área de cobertura de una transmisión 802.11b puede alcanzar, en teoría, los 50 metros en lugares cerrados (tales como oficinas y hogares) y hasta 400 metros en entornos abiertos, con velocidades de transmisión de 1 Mb/s, 2 Mb/s, 5,5 Mb/s y 11 Mb/s.

Es interesante notar que, para mantener la transmisión lo más funcional posible, el estándar 802.11b puede hacer que la tasa de transmisión de datos disminuya poco a poco hasta llegar a su límite mínimo (1 Mb/s) a medida que una estación está más lejos del punto de acceso. Lo

contrario también ocurre: cuanto más cerca del punto de acceso, mayor puede ser la velocidad de transmisión.

1.8.2. Protocolo 802.11a

El estándar 802.11a fue lanzado casi en la misma época que la versión 802.11b. entre sus principales características está la posibilidad de operar con tasas de transmisión de datos en los siguientes valores: 6 Mb/s, 9 Mb/s, 12 Mb/s, 18 Mb/s, 24 Mb/s, 36 Mb/s, 48 Mb/s y 54 Mb/s. El alcance geográfico de su transmisión es de 50 metros aproximadamente. Y a diferencia del estándar 802.11b, su frecuencia de operación es de 5 GHz, con canales de 20 MHz.

1.8.3. Protocolo 802.11g

El estándar 802.11g es el sucesor natural de la versión 802.11b, ya que es totalmente compatible con este y fue lanzado en 2003. Esto significa que un dispositivo que opera con 802.11g puede comunicarse con otro que trabaja con 802.11b sin ningún problema, sin embargo, limitando el máximo de transmisión compatible con el estándar 802.11b.

El estándar 802.11g tiene entre sus características poder trabajar con tasas de transmisión de hasta 54 Mb/s, así como sucede con el estándar 802.11a. Pero, a diferencia de esta versión, el 802.11g opera con frecuencias en la banda de 2,4 GHz (canales de 20 MHz) y tiene casi el mismo poder de cobertura, 50m en lugares cerrados y teóricamente hasta 400m en lugares abiertos.

1.8.4. Protocolo 802.11n

El protocolo 802.11n tiene como principal característica el uso de un esquema llamado Multiple-Input Multiple-Output (MIMO), capaz de aumentar considerablemente las tasas de

transferencia de datos por medio de la combinación de varias vías de transmisión (antenas). Con esto, es posible, por ejemplo, el uso de dos, tres o cuatro emisores y receptores para el funcionamiento de la red.

Una de las configuraciones más comunes en este caso es el uso de puntos de acceso que utilizan tres antenas (tres vías de transmisión) y de STAs con la misma cantidad de receptores. Lo interesante del protocolo 802.11n es su capacidad de hacer transmisiones en el rango de 300 Mb/s y, teóricamente, puede alcanzar velocidades de hasta 600 Mb/s. Incluso en el modo de transmisión más simple, con una vía de transmisión, el estándar 802.11n puede llegar a los 150 Mb/s.

En relación con su frecuencia, el estándar 802.11n puede trabajar con ambas bandas, como lo son las bandas 2,4 GHz y 5 GHz, lo que lo hace compatible con los estándares anteriores, incluso con el 802.11a. Cada canal dentro de esas bandas tiene, por defecto, ancho de 40 MHz. Su técnica de transmisión estándar es la OFDM, pero con ciertas modificaciones, debido al uso del esquema MIMO, siendo, por eso, muchas veces llamado MIMO-OFDM. Algunos estudios han probado valores superiores a 400 metros de área de cobertura.

1.8.5. Protocolo IEEE 802.11ah

Esta revisión es conocida también bajo el nombre de HaLow, Mejora el Wi-Fi al trabajar en un espectro por debajo de 1 GHz para ofrecer un mayor alcance y una conectividad con menor consumo de energía. Ideal para dispositivos IoT, que necesitan gran alcance, pero no necesariamente una gran velocidad inalámbrica. Sin embargo, son pocos o nulos los dispositivos comerciales disponibles en el mercado.

1.8.6. Protocolo IEEE 802.11ax

Diseñado para operar en los espectros de 2.4 y 5 GHz. Además de utilizar MIMO y MU-MIMO, es usualmente conocido como Wi-Fi 6, este nuevo estándar introduce OFDMA ofreciendo un mayor rendimiento. Es compatible con los protocolos anteriores y puede alcanzar una velocidad máxima teórica de 10 Gbps.

1.9. Tipos de seguridad de conexiones WIFI

La seguridad del Wi-Fi es variable y ha ido evolucionando a lo largo del tiempo, manejando el cifrado que se aplica a las comunicaciones entre el router y los adaptadores inalámbricos. El objetivo de los protocolos de seguridad es que los datos que se transmiten a través de una red Wi-Fi sean privados y no puedan ser interceptados por una tercera persona. Existen varias opciones, y se pueden dividir entre seguras y no seguras por sus características técnicas.:

1.9.1. WEP (Wired Equivalent Privacy)

Fue el primero que apareció para solucionar los problemas generados por las redes abiertas en el año 1999. Este sistema de cifrado está basado en el algoritmo de cifrado RC4, el cual utiliza claves de 64 o de 128 bits. En su llegada se descubrieron muchos agujeros y fallos de seguridad, lo que hizo que su uso fuera disminuyendo con el paso de los años. Aunque los principales agujeros y fallos fueron solucionados, lo cierto es que se trata de un cifrado con agujeros y poco fiable.

1.9.2. WPA (Wi-Fi Protected Access)

WPA fue la respuesta a las vulnerabilidades y los principales fallos de WEP. A diferencia de las claves de seguridad usados por WEP que son de 128 bits, las claves usadas por WPA suben a 256 bits, aunque no es la única mejora ya que incorpora la comprobación de contenidos

e integridad de mensajes para evitar que puedan ser interceptados y el uso del protocolo de clave temporal TKIP, lo cual ayuda a que un router no pueda ser atacado fácilmente como ocurría con WEP.

1.9.3. WPA2

WPA2 soluciona los problemas de vulnerabilidad detectados en la primera versión (WPA). La principal diferencia es el reemplazo del algoritmo RC4 por el algoritmo AES, uno de los más seguros actualmente, el cual realiza un cifrado por bloques para permitir claves más largas, seguras y la implementación del CCMP que se trata de un protocolo mejorado de encriptación que sustituye a TKIP, con 128 bits.

1.9.4. WPA3

Incorpora el cifrado de 192 bits en vez de 128 bits, lo que hace que el cifrado sea más seguro y difícil de romper. El protocolo WPA3 implementa como una de sus características deshabilitar protocolos anteriores, lo que provoca que los dispositivos WPA2 no se podrán conectar a puntos de acceso exclusivos de WPA3 que no tengan habilitado un modo de transición especial. Cuenta también con un nuevo modo de conectar y configurar a la red dispositivos sin la necesidad de que tengan pantalla ni botones físicos, etc. E incluso una misma contraseña que podría ser vulnerada en WPA2 no sería posible con WPA3.

1.10. Tecnología LoRa y protocolo LoRaWAN

LoRa es una tecnología que define la conectividad a nivel de capa 1 en el modelo OSI y es propiedad del fabricante Semtech. Esta tecnología inalámbrica está diseñada para permitir la interconexión de dispositivos de recursos muy reducidos (memoria, procesador y energía) y a

velocidades muy bajas, y hasta un máximo de 37,5 Kbps. El nombre LoRa deriva de las palabras “Long Range” que en español significa largo alcance.

Por otro lado, LoRaWAN es un protocolo de capa 2 promovido por LoRa Alliance, una asociación sin fines de lucro con la finalidad de promover y desarrollar la tecnología de manera interoperable y estandarizada. Esta asociación está compuesta por más de 500 empresas que desarrollan activamente soluciones y equipamiento relacionado. Según (Mayobre, 2018) la arquitectura básica de una solución que hace uso de una red LoRaWAN se implementa en una topología estrella, en la cual se hace uso de 3 tipos de equipos: nodos, gateways y servidor de aplicación. Es posible implementar redes tipo mesh o mall, en la cual los nodos retransmiten mensajes de otros nodos.

Gateways o antenas: son los encargados de recibir y enviar información a los nodos.

Nodos o dispositivos: son los dispositivos finales que envían y reciben información hacia el gateway.

Características de LoRa:

- Utiliza bandas ISM, bandas de radio industriales, científicas y médicas.
- La arquitectura de red que utiliza es “one-hop star topology”, es decir, desde los nodos se envía información a un gateway y éste la envía a un servidor de aplicaciones.
- Se puede conseguir arquitecturas mesh apoyándose en protocolos que lo implementen en capa 2 y superior.
- El bitrate es configurable, desde 11 bps y logrando alcanzar un máximo de 37.5 kbps.
- No hay restricciones de la cantidad de información a enviar y recibir.
- Largo alcance 10 a 20 km

- Puede manipular características de la modulación, las cuales varían el consumo de energía, el rango de alcance obtenido y el bitrate, entre otros parámetros.
- Libertad para construir sistemas a medida.
- Los circuitos integrados solo son fabricados por Semtech o empresas licenciadas por esta compañía.
- Frecuencias de trabajo: 433 Mhz en Asia, 868 Mhz en Europa y 915 Mhz en América.

Los parámetros de interés en LoRaWAN son 3 los cuales definen el desempeño del enlace de comunicación, estos son: ancho de banda (BW, bandwidth), factor de dispersión (SF, Spreading factor) y Coding Rate (CR).

El Ancho de banda es la longitud de la extensión de frecuencias en la que se encuentra la mayor potencia de señal. Si se requiere una transmisión rápida, la opción de 500MHz es la mejor, pero si se necesita transmitir la información a distancias más largas el módulo debe ser configurado con el valor de 125kHz. Mientras menor sea el ancho de banda más tiempo en el aire permanece la transmisión, pero a su vez la sensibilidad aumenta, por lo que la comunicación tiene un mayor alcance (González, 2018).

El Coding Rate (CR) o tasa de codificación puede tomar unos de estos cuatro valores: 4/5, 4/6, 4/7 y 4/8. El primer bit indica cuantos bits son de información (payload) y el segundo los bits enviados. Cuanto menor CR, mayor es el tiempo en el aire de la transmisión, esto hace que la tarea de recepción sea más rápida ya que el receptor puede demodular los paquetes con una menor potencia de recepción, pero también se tarda más tiempo en terminar la transmisión del paquete, esto quiere decir que el consumo de energía aumenta.

El Factor de dispersión (SF) determina cuántos símbolos se utilizan para codificar los datos del usuario. Esto se especifica para la modulación LoRa de 6 a 12. Por ejemplo, se utilizan 512 símbolos para los símbolos 9 y 4096 para los 12 para codificar los datos de usuario idénticos. Elevando el bit (2) al número de factor de dispersión.

Estos valores también afectan a la sensibilidad del receptor, se debe recordar que la sensibilidad se define como la potencia mínima que una señal transmitida debe tener para poder ser captada por el receptor para que pueda ser decodificada. La modificación de los parámetros anteriormente mencionados influye en la sensibilidad del decodificador, además del tiempo en el aire de la transmisión. Al aumentar el Ancho de Banda se disminuye la sensibilidad del receptor, al aumentar el valor del factor de dispersión se logra un aumento en la sensibilidad y al disminuir el valor del Coding Rate se reduce la ratio de paquetes con error en presencia de pequeñas ráfagas de interferencia, pero el bitrate efectivo disminuye.

Hay tres clases de dispositivos de nodo final, los cuales permiten la comunicación bidireccional y puede iniciar una subida de información a los servidores a través de una puerta de enlace (Pickering, 2017), estos son: Dispositivos Clase A, Clase B y Clase C.

Clase A una vez que realizan una transmisión en el canal de subida abren dos ventanas para la escucha del canal de bajada por un corto período de tiempo.

Clase B se añaden funcionalidades en el momento en el que el dispositivo envía información por el canal de subida, como la sincronización con los concentradores, lo que permite a los servidores de red saber cuándo el dispositivo espera una respuesta por parte de ellos y los dispositivos LoRa.

Clase C el cual a diferencia de las dos clases anteriores tiene la ventana para la recepción de información desde el canal de bajada siempre abierto, que se cierra únicamente cuando el dispositivo tiene información para mandar al concentrador.

1.11. Protocolos de comunicación IoT

Los protocolos de comunicación para IoT son la forma que se envían los paquetes de datos del emisor al receptor, lo que asegura que pueda haber una comunicación efectiva entre las partes, es análogo a la comunicación en las personas, la tecnología sería las ondas electromagnéticas con el estándar WiFi y el protocolo de comunicación sería el lenguaje con el que se comunican por lo que deben de hablar el mismo lenguaje para entenderse.

A continuación, veremos los más importantes:

1.12. HTTP (REST/JSON)

Llamado así por Hypertext Transfer Protocolo o Protocolo de transferencia de hipertexto, el cual fue creado por fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force. Es un protocolo cliente-servidor, muy accesible por ser de código abierto, además de poseer numerosas librerías. Es efectivo para enviar grandes cantidades de información, como por ejemplo lecturas de sensores con intervalos de algunos segundos, minuto a minuto o cada hora; aunque no es adecuado ni para enviar actualizaciones en periodos de tiempo del orden de milisegundos tampoco para enviar información de video. El protocolo HTTPS aplica el protocolo criptográfico SSL/TLS sobre HTTP. No obstante, el método más seguro no consiste en un servidor HTTP en el dispositivo IoT sino solo un cliente HTTP, de manera que el dispositivo IoT pueda iniciar conexiones a un servidor web, pero no es capaz de recibir solicitudes de conexión.

1.13. MQTT

El MQTT (Message Queuing Telemetry Transport) es un protocolo de tipo publicación-suscripción de nivel de aplicación con una versión para redes no basadas en TCP/IP, como podría ser Zigbee. Este protocolo ha sido implementado en múltiples aplicaciones como, por ejemplo, MS Azure IoT hub, para entornos de generación de electricidad mediante fuentes renovables o el Messenger de Facebook. Puesto que MQTT envía credenciales de conexión en claro (no incluye en su diseño medidas de seguridad) sin autenticación o cifrado, es recomendable usarlo con TLS para asegurar las comunicaciones en su versión sobre TCP.

1.14. CoAP

Es un protocolo cliente/servidor, es similar a HTTP, pero usa UDP/multicast en lugar de TCP. Llamado así por Constrained Application Protocol fue creado por IETF (Internet Engineering Task Force) para proveer la compatibilidad de HTTP con una mínima carga, además de simplificar el encabezado reduce el tamaño de cada requerimiento. Su seguridad es DTLS (Datagram Transport Layer Security) basado en el protocolo TLS y proporciona garantías de seguridad equivalentes y básicamente consiste en aplicar seguridad en la capa de transporte para proteger las comunicaciones.

Capítulo II

2. METODOLOGÍA Y DISEÑO

2.1. METODOLOGÍA

Se ha seguido una metodología basada en la investigación de las soluciones que se han planteado al problema, así mismo como si existen dispositivos comerciales que cumplan con la necesidad, con su respectivo análisis de sus ventajas y desventajas.

Como se ha hecho en el capítulo 1 se realiza el estudio de los sistemas de riego automáticos existentes, de los sistemas que utilizan el internet de las cosas para operar y las diferentes tecnologías aplicadas. Para aquello se revisa los trabajos en esta dirección que ha desarrollado la Universidad Técnica del Norte, los sistemas de riego automatizados comerciales local y globalmente, para proceder a buscar trabajos de tesis que potencialmente resuelven el problema a nivel global y las posibles soluciones comerciales que están desarrollándose entorno a la temática de sistemas de irrigación con tecnología IoT.

Para tener un criterio lógico, coherente y objetivo se plantea empaparse de las tecnologías existentes para el IoT o Internet de las cosas, para lo cual se desarrolla una tabla comparativa tanto en sus tecnologías como en sus protocolos, determinándose dos tecnologías probables para la solución del problema las cuales se estudia con mayor detalle técnico y se elige una solución que satisfaga el problema tomando muy en cuenta la coyuntura actual por la pandemia.

Para el diseño del prototipo se plantea la selección de tecnologías a utilizar partiendo de los requisitos del sitio de trabajo, determinando el área de trabajo y delimitando la zona a intervenir, se debe tomar en cuenta que es un prototipo por lo que el estudio y la

implementación se realiza para el funcionamiento del sistema, sin considerar los parámetros variables propios del área agronómica.

En tanto a la selección de los elementos de hardware y software se determinan por las características necesarias, la disponibilidad, el nivel de desarrollo para las tecnologías seleccionadas, madurez, compatibilidad, nivel de aprendizaje, etc. En el caso de software hemos decidido utilizar software libre por las ventajas que este ofrece.

En el diseño propio del prototipado se realiza el levantamiento del plano eléctrico requerido para el funcionamiento del sistema, el diseño electrónico se realiza con los elementos representativos del hardware seleccionado y acoplado sus entradas y salidas correctamente. En cuanto al diseño de la interfaz web se ha diseñado para que sea fácil e intuitiva para el público en general dentro del nivel de modificación que permite la plataforma que se ha elegido para trabajar.

Para la construcción del prototipo se realiza la instalación de controladores y librerías necesarias para el desempeño deseable del sistema, así como las actualizaciones disponibles, para después entrar en la programación del sistema aplicando la ingeniería recurrente de acuerdo con los problemas que se presenten al ir añadiendo funciones. Finalmente, la implementación se la realiza en el sitio piloto de pruebas del sistema donde queda funcionando para comprobar su validación.

2.2. Requisitos del prototipo para sistemas de irrigación piloto

El lugar para la prueba piloto del prototipo se encuentra ubicado en Tomas, Sosoranga Loja Ecuador

- Poder controlar el riego a distancia
- Conocer el porcentaje de humedad del suelo
- Tener acceso vía internet
- Poder tener un programa automático
- Que los instrumentos funcionen con la energía de red 110-120V
- Interfaz fácil de manejar, intuitiva
- De fácil mantenimiento
- El control del riego debe de ser desde una tubería de agua convencional

2.2.1. Requerimientos

- Sistema de riego automático
- Funcione con energía eléctrica
- Se conecte a internet
- Pueda controlar el riego desde cualquier lugar con la confiabilidad de que se ejecute el riego cuándo debe regarse
- La opción de poder regar manualmente

2.2.2. Restricciones

- Debe funcionar con energía eléctrica de la casa 110 voltios
- El suministro de agua es de la llave
- El riego es riego por goteo

2.2.3. Deseables

- La interfaz sea fácil de usar
- Que el riego se realice al atardecer o al amanecer
- Que el sensor sea resistente a condiciones externas
- Los datos puedan enviarse a grandes distancias

2.3. Análisis del sitio de trabajo

El sitio de trabajo es una finca ubicada en el sector llamado Tomás de la parroquia Sosoranga, es un sitio alejado donde no existe la señal de celular, pero si se obtiene internet. Es un ambiente tropical húmedo donde se da diferente tipo de vegetación como el maíz, guineo, yuca, mangos, limones, tomate entre otras. La necesidad que tiene el dueño de esta finca es que no pasa todo el tiempo en ella y a su vez desea obtener cierto tipo de vegetación que necesita de un riego casi constante para conveniencia de este trabajo el sitio donde tiene la huerta queda cerca de la zona residencial por lo que tenemos acceso a wifi, la fuente de agua para el suministro se la puede obtener de la llave y la electricidad se puede derivar de la casa.

2.4. Tipo de control

Se realiza un sistema de control de lazo cerrado en el cual se recibe información del entorno de la variable de humedad del suelo, lo que permite al actuador realizar el riego de las plantas hasta que el valor del sensor sea igual o mayor del valor seteado como máximo porcentaje de humedad permitida. En la figura 4 podemos observar como sería el ingreso del valor de la señal del sensor.

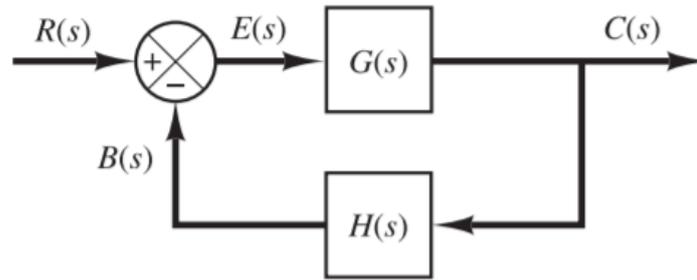
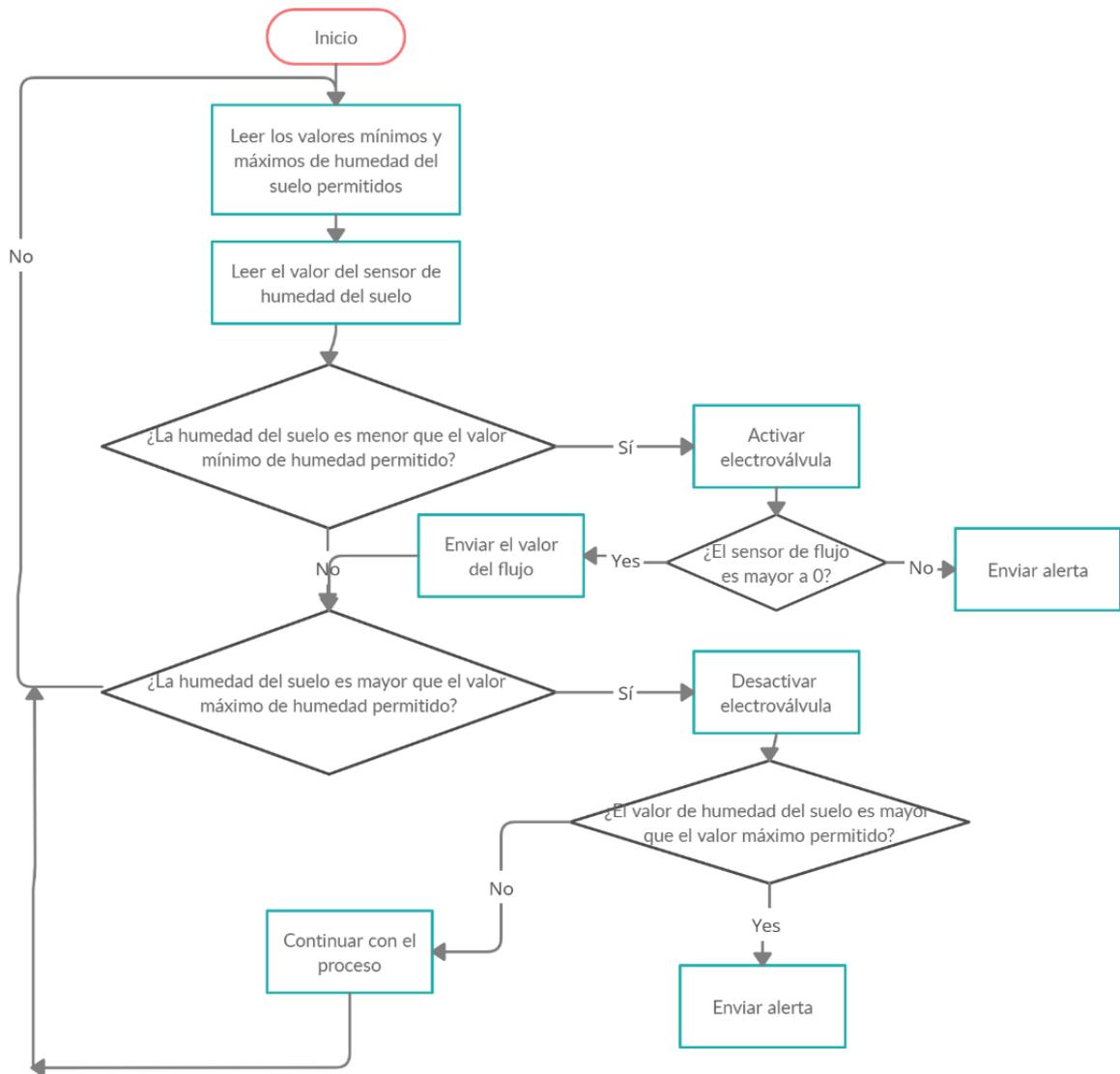


Figura 4. Sistema de control en lazo cerrado(Ogata, 2010)

El diagrama de flujo del programa se lo puede asociar como se muestra en la figura 5, tomando en consideración que al ser un sistema descentralizado se puede agregar diferentes tópicos con diferentes sensores y de acuerdo con los valores que reciba tomar decisiones.



Autor Rogger Román

Figura 5. Diagrama de flujo del prototipo (Autoría propia)

2.5. Selección de elementos

Para la selección de los elementos se hizo un estudio donde se determina la diferente tecnología a utilizar entre otros aspectos se detalla el estudio realizado y luego una tabla de la selección de cada elemento.

Tabla 1
Características de las tecnologías para IoT

Tecnología	Consumo	Alcance	Madurez	Disponibilidad	Seguridad	Usabilidad	Tasa de datos
BLE	Muy bajo	Muy bajo	Alto	Medio	Baja	Media	Baja
WiFi	Alto	Bajo	Muy alta	Muy alto	Alta	Alta	Media
ZigBee	Medio	Bajo	Medio	Muy bajo	Alta	Baja	Baja
SigFox	Bajo	Medio	Baj	Bajo	Media	Baja	Muy baja
LoRa	Muy Bajo	Alto	Bajo	Bajo	Medio	Bajo	Muy baja
GSM/GPRS	Muy alto	Alto	Muy alta	Bajo	Alta	Media	Alta

Nota. Elaborado en base con la información de los fabricantes. (Autoría propia)

De acuerdo con la tabla de características presenta anteriormente, los requerimientos, restricciones y los deseables, se procedió a calificar a las tecnologías con las que se puede trabajar. Como se muestra en la siguiente tabla:

Tabla 2
Selección de la tecnología

Tecnología	Consumo	Alcance	Madurez	Disponibilidad	Seguridad	Usabilidad	Tasa de datos	total
BLE	5	1	4	3	2	3	4	22
WiFi	2	2	5	5	4	4	3	25
ZigBee	3	2	3	1	4	2	4	19
SigFox	2	3	2	2	3	2	5	19
LoRa	5	4	2	2	3	2	5	23
GSM/GPRS	1	4	5	2	4	3	2	21

Autoría propia

Siendo la tecnología WiFi como la que mejor se adapta a nuestro proyecto, con mención especial a la tecnología LoRa ya que presenta condiciones similares de usabilidad a la seleccionada. El protocolo de conexión WiFi que se utilizará es el protocolo 802.11n, porque permite trabajar con Multiple-Input y Multiple-Output (MIMO), y la conexión de seguridad que se realiza es WPA2 haciendo uso del AES, que realiza un cifrado por bloques para permitir claves más largas y seguras. Siendo uno de los estándares más seguros y utilizados a nivel mundial.

A continuación, se realiza una búsqueda de protocolos de IoT que funciones con WiFi para la comunicación con la red, donde se estudiaron los más importantes y sus características son:

Tabla 3
Protocolos de comunicación para IoT

Protocolo	Pros	Contras
HTTP	- Muchas bibliotecas	- Consumo de energía
MQTT	- Lo mejor para pub / sub - Compatible con IoT	- Lecturas no admitidas
CoAP	- Ligero: compatible con IoT	- SSL no compatible (todavía)

De los protocolos estudiados y disponibles se elige el que mejor se adapta a nuestras necesidades en este caso el protocolo MQTT. Parte de la solución seleccionada se debe a que el servidor puede iniciar la conversación con el cliente, cosa que carecen otros protocolos.

2.5.1. Selección de hardware

Para la selección del hardware se realiza un estudio de las soluciones que se ofrecen en el mercado, que puedan cumplir nuestros requerimientos y trabajen con las tecnologías seleccionadas. A nivel mundial existen muchos dispositivos creados para trabajar con IoT que se adaptan a nuestras necesidades. En los estudios de trabajos de tesis que utilizan IoT para plantear sus soluciones se pueden evidenciar una gran variedad de dispositivos, pero a nivel

nacional no son conocidos, y no existen en el mercado. Un ejemplo de esto es el Arduino MKR WIFI 1010. Sin embargo, si existen diferentes soluciones compatibles con nuestras necesidades como son el Node MCU, Wemos D1, LoRa, Modulo Wroom Arduino, wifi, etc.

En la investigación de cada uno de los elementos encontramos que elementos como el módulo Wroom Arduino estaban quedando obsoletos porque ya existían soluciones oficiales de Arduino que venían con el módulo integrado y trabajaba mejor, y otros recientes que no tenían mucho desarrollo de código o librerías para trabajar con los diferentes elementos de hardware que la aplicación requería.

Para esta aplicación se requiere de dos módulos separados uno que regule el paso del agua y otro módulo que sirva de sensor donde lea y envíe la información.

2.5.1.1. Módulo principal

Por ser de código abierto, por la compatibilidad y desarrollo de librerías con los diferentes elementos a trabajar se selecciona el Node MCU con el esp8266 para el módulo principal, el cual tiene como funciones activar la electroválvula, leer el sensor de flujo y realizar la conexión con la aplicación. Adicionalmente contiene led indicador del estado de funcionamiento.

2.5.1.1.1. Node MCU

Es dispositivo IoT de código abierto. Incluye el firmware Wi-Fi ESP8266 de Espressif Systems que se ejecuta en el System on chip mayormente conocido como SoC. El hardware se basa en el módulo ESP-12. El término "NodeMCU" se refiere al firmware en lugar de a los kits de desarrollo. Es compatible con bibliotecas MQTT de algunas plataformas como Electronic cats, Contiki, etc. (Ver Apéndice A)

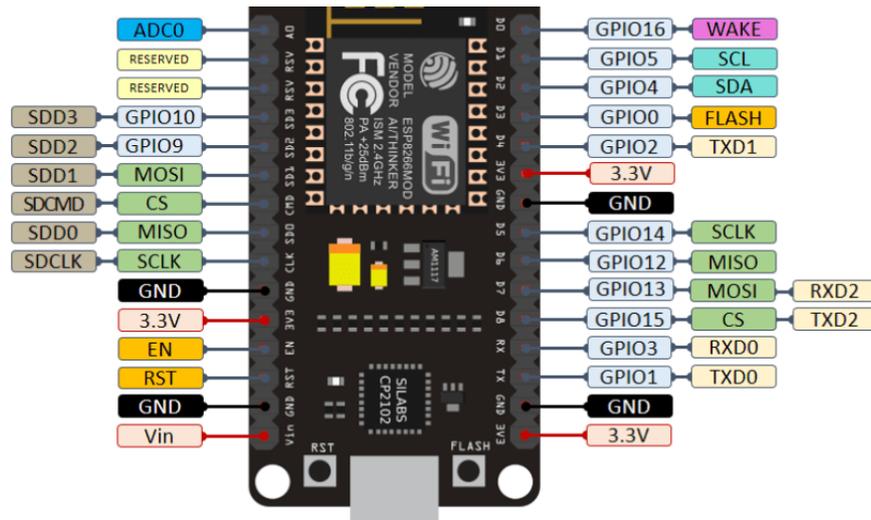


Figura 6. Tarjeta NodeMCU 8266 y su configuración de pines (Electronilab, 2020)

2.5.1.1.2. Electroválvula solenoide

Es una válvula electromecánica de ½ pulgada, diseñada para controlar el paso del agua, esta válvula se mueve mediante una bobina solenoide, el cual convierte la energía eléctrica, mediante magnetismo, en energía mecánica la cual puede abrir o cerrar la válvula. Tiempo de respuesta < a 1 segundo y funciona a 12 V DC.

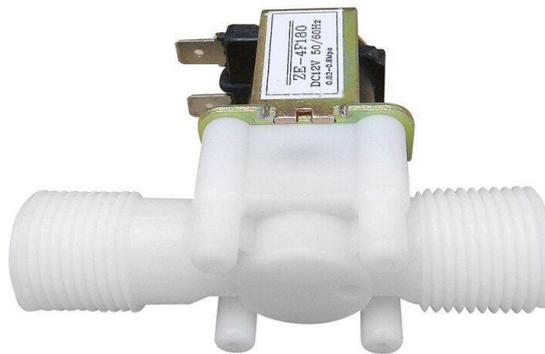


Figura 7. Electroválvula solenoide 12V (Valparaíso, n.d.)

2.5.1.1.3. Sensor de flujo YF-S201

El sensor de flujo o también llamado caudalímetro es un dispositivo que, instalado en la línea de tubería, (en este caso de 1/2 pulgada) mide el valor de flujo del agua en litros por minuto. El sensor de flujo de agua de 1/2" YF-S201 sirve para determinar en que momento se encuentra circulando líquido por la línea, es un elemento de monitoreo que permite conocer si el módulo principal, tiene las condiciones adecuadas de funcionamiento, como por ejemplo si después de activar la electroválvula el sistema cuenta con la circulación del líquido en este caso agua, debido a que es un sistema remoto debemos de asegurarnos que el suministro de agua se encuentra disponible y funcionando. Funciona a 5 V, y tiene una salida de pulsos del sensor de efecto Hall, la cual envía pulsos cada vez que gira, permitiendo conocer el aproximado de la cantidad de agua que pasa por el sensor.



Figura 8. Sensor de flujo de agua de 1/2" YF-S201 (ElectronicsComp, 2021)

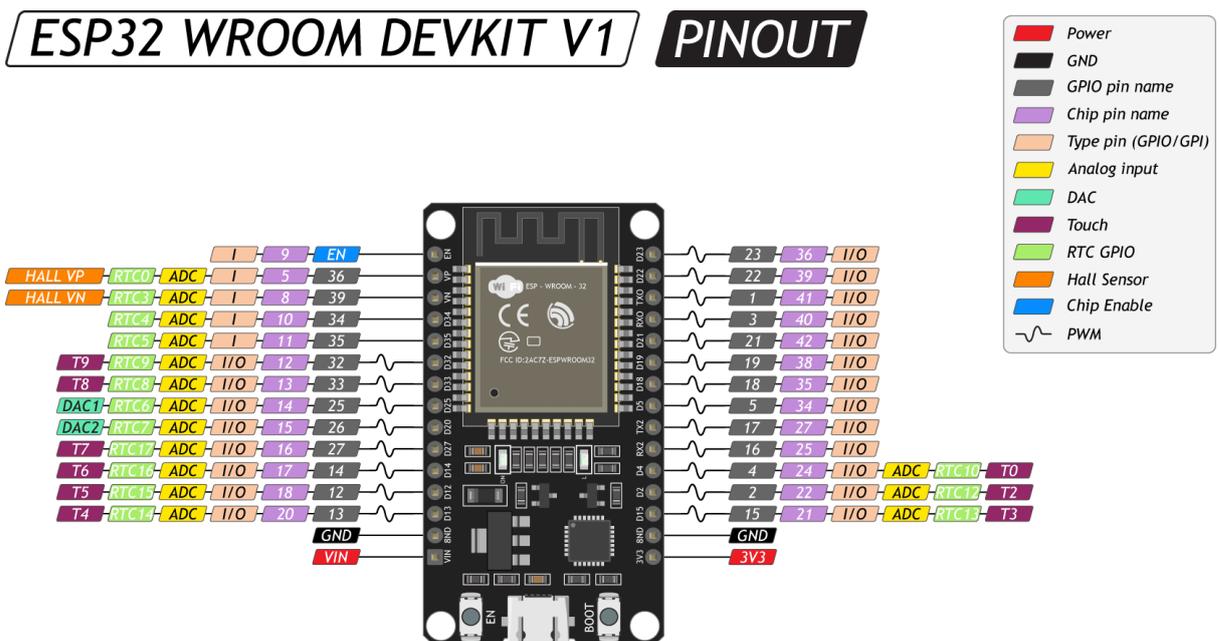
2.5.1.2. Módulo sensor

Para este este módulo, se requiere que obtenga los valores de humedad del suelo y los publique en la aplicación, al igual debe de ser compatible con las tecnologías seleccionadas y estar dispuesto a actuar de acuerdo lo pida la aplicación. La mejor elección por lo antes

mencionado es el ESP32 Devkit, que cuenta entre sus características con una mejorada versión de comunicación wifi.

2.5.1.2.1. ESP32 Devkit

Es una placa con doble núcleo, lo que significa que tiene dos procesadores, cuenta con wifi 2.4 GHz que puede transmitir hasta 150Mbps/s. Es compatible con MQTT y puede ejecutar programas de 32 bits, tiene una amplia variedad de periféricos disponibles, como: táctil capacitiva, ADC, SPI, I2C, DAC, UART, entre otros. Cuenta con sensor interno de temperatura y sensor de efecto Hall incorporado. Cuenta con interfaz USB a UART, para que pueda programarlo fácilmente usando Arduino IDE u otros entornos de desarrollo. (Ver Apéndice B).



corriente a través del suelo, lee la resistencia y obtienen el valor de humedad. A mayor presencia de agua hace que la tierra conduzca electricidad más fácil (Menor resistencia), mientras que un suelo seco es un conductor pobre de la electricidad (Mayor resistencia).

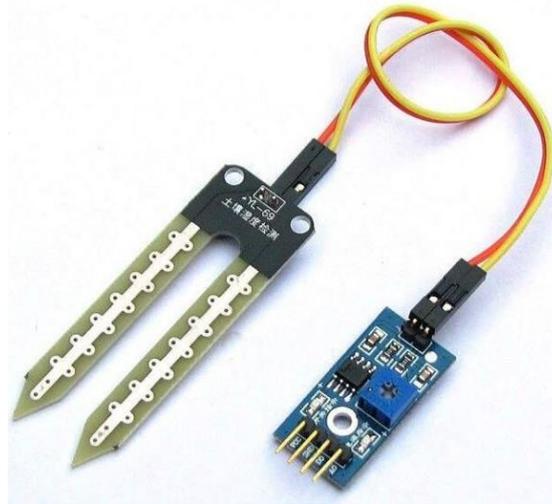


Figura 10. Sensor de humedad YL-69 (Caldas, 2021)

2.5.2. Selección de software

Una de las ventajas de elegir hardware libre es que la comunidad desarrolla herramientas que permiten trabajar con software libre como es el caso de las placas seleccionadas son compatibles para trabajar con el IDE de Arduino y con el lenguaje algunos lenguajes de programación, para este proyecto y por las librerías compatibles se selecciona el lenguaje C++.

Después de seleccionar la tecnología y el protocolo a utilizar, se puede continuar con la selección del software, pero no hay que reinventar la rueda y ya existen múltiples compañías que se dedican a facilitar el manejo de aplicaciones IoT desde la parte de la nube, por lo que se procede a realizar una investigación de los posibles aliados a trabajar.

Donde hay que distinguir de las propuestas comerciales que requieren licencia para trabajar y las que puedes desarrollar tu proyecto, utilizarlas siempre y cuando no excedas los límites mensuales. A continuación, una tabla resumen de las compañías de pago.

Tabla 4
Comparación de plataformas IoT de pago

	Azure IoT Hub	AWS IoT	Plataforma Watson IoT	Plataforma Sofia2 IoT
Protocolos	HTTP, AMQP, MQTT y protocolos a medida usando Protocol Gateway	HTTP, MQTT	HTTP, MQTT	HTTP, MQTT, AMQP, JMS, OPC y protocolos a medida vía Gateway Plugin
Patrones de comunicación	Telemetry, Command	Telemetry, Command	Telemetry, Command	Telemetry, Command
Plataformas certificadas	Intel, Raspberry Pi, Freescale, Texas Instruments, MinnowBoard, BeagleBoard, Seeed, resin.io	Broadcom Marvell, Reneses, Texas Instruments, Microchip, Intel, Mediatek, Qualcomm, Seeed, BeagleBoard	ARM mbed, Texas Instruments, Intel, Raspberry Pi, Arduino Uno	Intel, ARM, Raspberry Pi, Arduino, Texas Instruments, Seeed, Cisco, BeagleBoard, Texas Instruments
SDK/Lenguaje	Net y UWP, Java, C, NodeJS	Javascript, Arduino, Java, Python, iOS, Android	C#, C, Python, Java, NodeJS	Java, Javascript, C, NodeJS, iOS, Android, Python, .Net, Arduino
Seguridad	TLS (Server Authentication)	TLS (Mutual Authentication Certificado X.509 con autenticación cliente, IAM Service, Cognito Service	TLS	TLS
Autenticación	Por dispositivos con SAS token	Pago por tráfico en millones de mensajes (publicados de dispositivos y entregados a dispositivos)	Per-device con token	Per-device/system con token, certificado X509, user/password
Precio	Pago por unidad IoT Hub según número de dispositivos y mensajes por días		Pago por número de dispositivos, tráfico de datos y almacenamiento	Modo SaaS: pago por tráfico de datos y almacenamiento Modo PaaS: pago por VMs

Elaborado en base de información de las plataformas. Autoría propia

En cambio, las opciones que no exigen pago tenemos las plataformas de ThingSpeak y Ubidots, cada cual, con sus pro y contras, que las hacen herramientas poderosas para el internet

de las cosas, ambas ya llevan algunos años en el mercado, aportando soluciones para proyectos alrededor del mundo, por lo que su desarrollo es superior al de otras plataformas.

ThingSpeak es una aplicación y API de Internet de las cosas (IoT) de código abierto para almacenar y recuperar datos de cosas, compatible con MQTT. Permite la creación de sensores aplicaciones de registro, aplicaciones de rastreo de ubicación y una red social de cosas con actualizaciones de estado. Su característica más importante es que tiene soporte integrado del software de computación numérica MATLAB de MathWorks, permitiendo visualizar y analizar datos cargados usando Matlab sin requerir la compra de una licencia de Matlab de Mathworks. Sin embargo, no permite la integración de widgets de control que sean amigables con el usuario final, por lo que el usuario de la aplicación no podrá realizar cambios desde la interfaz de usuario, impidiendo el control de la electroválvula.

Ubidots es una empresa de visualización y análisis de datos de Internet de las cosas (IoT), diseñada para permitir crear prototipos y escalar los proyectos de IoT a producción, mientras mejora y economiza el mundo físico que nos rodea, con ayuda de los datos de los sensores y actuadores, los cuales los convierte en información importante para la toma de decisiones, las interacciones de máquina a máquinas, etc. El valor agregado de Ubidots es que el tiempo de desarrollo se reduce notablemente, cuenta con una arquitectura centrada en la eficiencia de los datos y una UX (experiencia de usuario) atractiva, los usuarios pueden conectarse, crear e implementar aplicaciones de IoT en la nube con facilidad, dejando a Ubidots para manejar la infraestructura de UX en la nube y del usuario final. Esta plataforma empodera a más de 60.000 estudiantes, creadores e investigadores para explorar y desarrollar tecnología y soluciones, dando a todos los usuarios sus primeros 3 dispositivos gratis siempre. Y se puede enviar hasta 40000 datos diariamente.

La razón fundamental es porque brindan una conexión simple y segura, para enviar y recuperar datos hacia y desde nuestro servicio en la nube en tiempo real. Permitiendo el intercambio de información desde cualquier dispositivo que esté vinculado.

2.5.3. Diseño de la interfaz

El diseño de la interfaz de usuario se la diseña lo más amigable con el usuario, como sea posible dentro de los parámetros de personalización que proporciona Ubidots. Se divide en 2 dashboards la primera llamada “Control”, en la cual tiene los elementos de control y visualización de los valores de los sensores y de cómo evolucionan en el tiempo. Como se muestra a continuación.

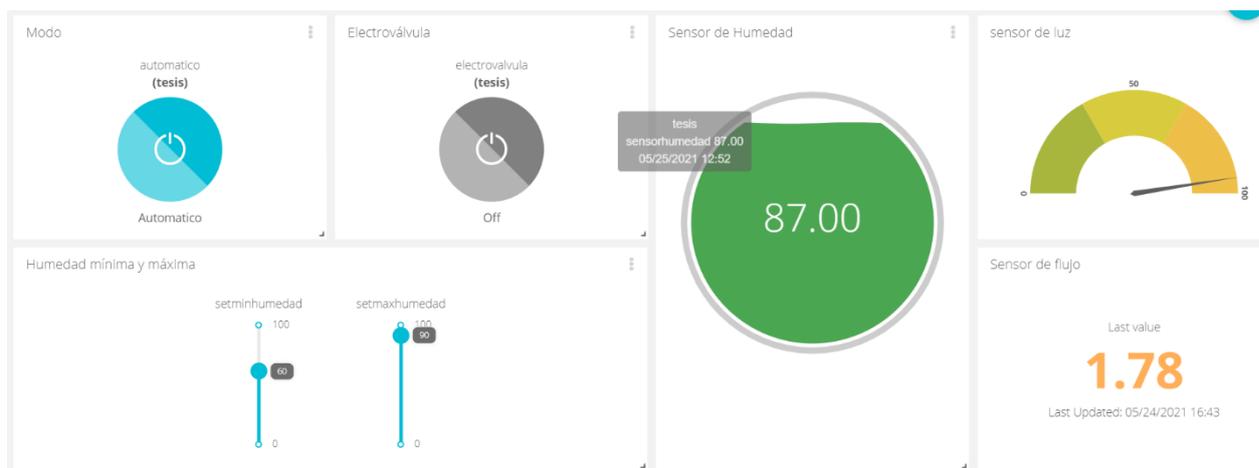


Figura 11. Diseño de la interfaz del proyecto (Autoría propia)

Este a su vez se divide en 3 partes, la parte de control donde se puede activar o desactivar le modo manual y automático, así también el encendido y apagado de la electroválvula, y los rangos de humedad que se desea para el sistema, tanto la humedad mínima que se permitirá antes de que la electroválvula se encienda y la humedad máxima permitida en la que la electroválvula se apaga.

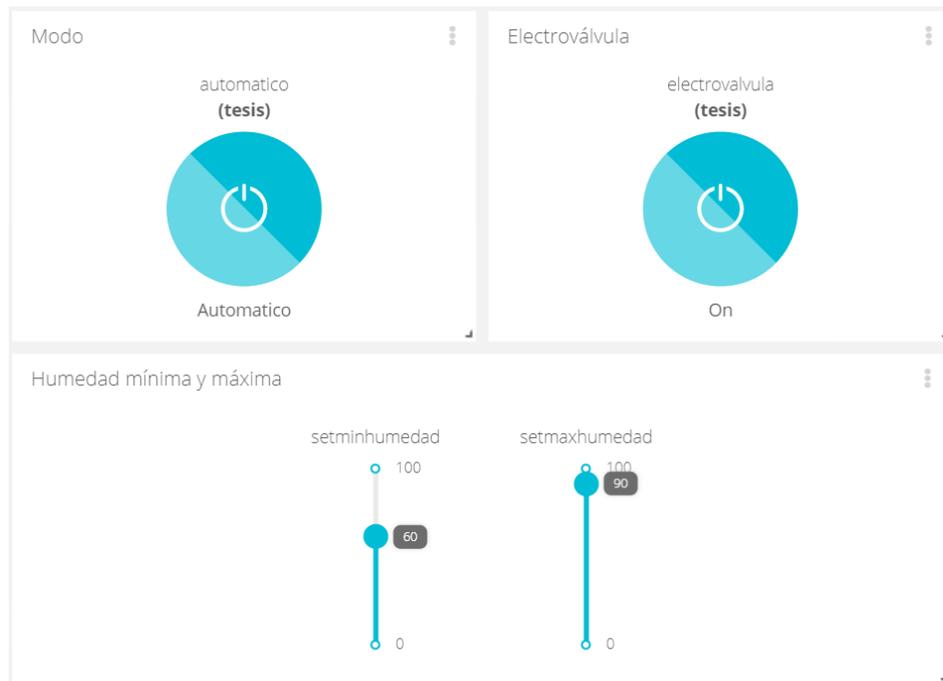


Figura 12. Sección de control (Autoría propia)

La siguiente subdivisión son los últimos valores registrados de los sensores y cuando se registró, tratando de hacer los gráficos más interactivos para una fácil comprensión.

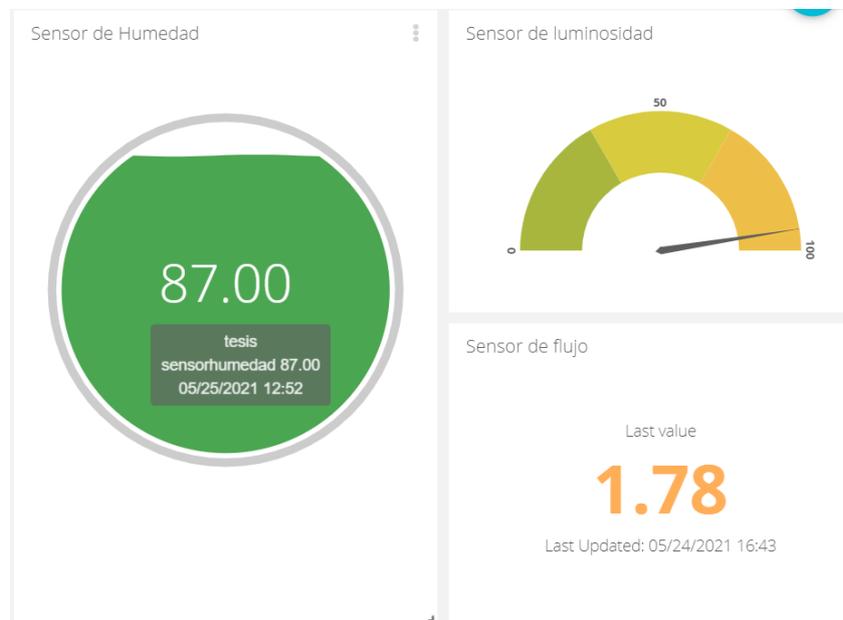


Figura 13. Sección de visualización (Autoría propia)

En la tercera subdivisión tenemos una gráfica interactiva donde se muestran los valores obtenidos a lo largo del tiempo, tanto de los sensores como de los mínimos y máximos

permitidos, cabe destacar que las gráficas se encuentran vinculadas con el tiempo de la dashboard, lo que permite visualizar las gráficas de acuerdo con el tiempo que se desee solo cambiando el intervalo de tiempo de la dashboard. Así puede visualizar lo que a ocurrido el último día o la última semana, el último mes, etc.



Figura 14. Evolución de los datos en el tiempo (Autoría propia)

La siguiente dashboard llamada “Datos” permite visualizar todos los datos o puntos que han llegado a Ubidots, en la fecha y hora recibidos, así como cualquier cambio hecho en la plataforma, todo queda registrado. Como se muestra a continuación.

Values Table		Values Table		Values Table	
DATE	AUTOMATICO (TESIS) LAST VALUE	DATE	ELECTROVALVULA (TESIS) LAST VALUE	DATE	SENSORFLUJO (TESIS) LAST VALUE
04/26/2021 20:50	0.00	04/29/2021 03:54	0.00	04/26/2021 20:50	0.00
04/26/2021 20:49	1.00	04/29/2021 01:09	0.00	04/26/2021 20:49	0.00
04/25/2021 13:34	0.00	04/29/2021 00:52	0.00	04/25/2021 13:33	0.00
		04/29/2021 00:49	0.00	04/25/2021 13:32	0.00
		04/28/2021 23:44	0.00	04/25/2021 13:32	0.00
		04/28/2021 23:40	0.00	04/25/2021 13:31	0.00
		04/28/2021 13:32	0.00	04/25/2021 13:30	0.00
		04/28/2021 13:31	0.00	04/25/2021 13:29	0.00
		04/28/2021 03:54	0.00	04/25/2021 13:28	0.00
		04/27/2021 23:22	0.00	04/25/2021 13:27	0.00

Figura 15. Sección de los datos o puntos recibidos en el bróker (Autoría propia)

Values Table		Values Table	
DATE	SENSORLUZ (TESIS) LAST VALUE	DATE	SENSORHUMEDAD (TESIS) LAST VALUE
04/29/2021 12:24	74.00	04/29/2021 12:25	0.00
04/29/2021 12:09	63.00	04/29/2021 11:55	0.00
04/29/2021 11:54	61.00	04/29/2021 11:25	0.00
04/29/2021 11:39	60.00	04/29/2021 10:55	0.00
04/29/2021 11:24	63.00	04/29/2021 10:25	0.00
04/29/2021 11:09	58.00	04/29/2021 09:55	0.00
04/29/2021 10:54	59.00	04/29/2021 09:24	0.00
04/29/2021 10:39	59.00	04/29/2021 08:54	0.00
04/29/2021 10:24	61.00	04/29/2021 08:24	0.00
04/29/2021 10:09	57.00	04/29/2021 07:54	0.00

Figura 16. Visualización de datos de sensores (Autoría propia)

Capítulo III

3. Construcción e implementación

3.1. Desarrollo del prototipo

La función básica de este prototipo es entregar el líquido vital a las plantas cuando estas la necesiten, entonces para saber cuándo una planta necesita agua se necesita conocer si el nivel de humedad de agua presente en el suelo es adecuado para que este ser vivo pueda tomarla sin mayor esfuerzo y en niveles adecuados, para la investigación de la presente tesis se obtuvo mayor conocimiento del sistema integral que debe tener este proceso, en los cuales los niveles de humedad que el suelo necesita tener es variable y depende del tipo de cultivo, época del año del tamaño de la planta, de la edad que tiene, entre otros factores para que el riego sea óptimo. Resolver estos detalles se encuentran fuera del alcance del presente trabajo. Pero lo que se realiza el proyecto es permitir seleccionar manualmente los niveles de humedad máximos y mínimos que el usuario cree convenientes, haciendo de este un sistema versátil.

Las placas de desarrollo son compatibles con software libre como el IDE de Arduino, por lo que para poder programar en esta plataforma se debe de instalar primero los drivers o controladores para que la PC pueda reconocerlos en tu puerto USB, desde los siguientes enlaces:

- http://www.wch.cn/downloads/CH341SER_EXE.html (para el ESP8266)
- <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers> (para el ESP32)

Los drivers para poder trabajar con las placas desde el IDE de Arduino podemos agregarlos desde dentro de la aplicación en la pestaña de “file”, luego “preferences”, en la pestaña de “settings, en el área de “Additional Boards Manager URLs” debemos de pegar los siguientes enlaces:

- http://arduino.esp8266.com/stable/package_esp8266com_index.json (ESP8266)
- https://dl.espressif.com/dl/package_esp32_index.json (ESP32).

Estos enlaces permiten configurar la aplicación para posteriormente poder seleccionar las placas de desarrollo en las que se trabaja. Otra guía más detallada se la puede encontrar en su repositorio de GIT oficial en el siguiente enlace:

<https://github.com/espressif/arduino-esp32/blob/master/docs/arduino-ide/windows.md>

Para comprobar si funcionan los drivers se puede conectar la placa y en la pestaña de “file”, “examples”, seleccionas algún ejemplo de las opciones de las placas de desarrollo por ejemplo el programa Blink. Con mención especial para la placa ESP32 en el momento que se encuentra cargando el archivo a la placa debemos de tener presionado el pulsador “Boot” de la placa para que el programa pueda escribirse de forma correcta, caso contrario no se graba.

Una vez comprobado la correcta comunicación se precede a instalar las librerías necesarias para trabajar con la tecnología, protocolo de comunicación, y elementos adicionales en el proyecto. Se puede realizar desde la plataforma de Arduino o descargar el archivo comprimido de las librerías en formato .ZIP y luego agregarlas.

Después de la selección del sensor de humedad YL-69, por ser ampliamente utilizado con estas placas de desarrollo y porque permite una lectura muy confiable, se procede a realizar las primeras lecturas por la entrada del pin analógico de la placa ESP32, los datos del sensor de humedad de suelo YL-69 resistivo cuando se encuentra en el aire libre su valor es de 4095 esto se debe a que la resolución del pin analógico del ESP32 es de 2 elevado a la 12 por los que teóricamente de 0 a 4095, mostrando su valor mínimo cuando está totalmente sumergido en el agua es de 1650 -1800 aproximadamente. Para diferentes tipos de suelo se recomienda realizar

la calibración de acuerdo con las condiciones que va a trabajar, con el suelo y el agua que van a ser utilizados en la aplicación. En el código se implementa la función “map” para hacer la conversión de los máximos y mínimos valores a los valores de 0 a 100 % de humedad, obteniendo el valor del porcentaje de humedad del suelo. Es recomendable realizar la activación del sensor mediante un pin digital para que solo se encienda cuando vamos a tomar la medida, esto porque al ser un sensor resistivo suele deteriorarse o corroerse la parte metálica del sensor.

Para el módulo de control se realiza la conexión de la electroválvula seleccionada, tomando en cuenta que la tarjeta de desarrollo NodeMCU con el ESP8266 solo entregan a la salida de los pines digitales 3.3V se debe realizar un circuito acoplador con un transistor 2N2222 (Ver Apéndice C) y su respectivo relé de 5V, como se muestra en la figura.

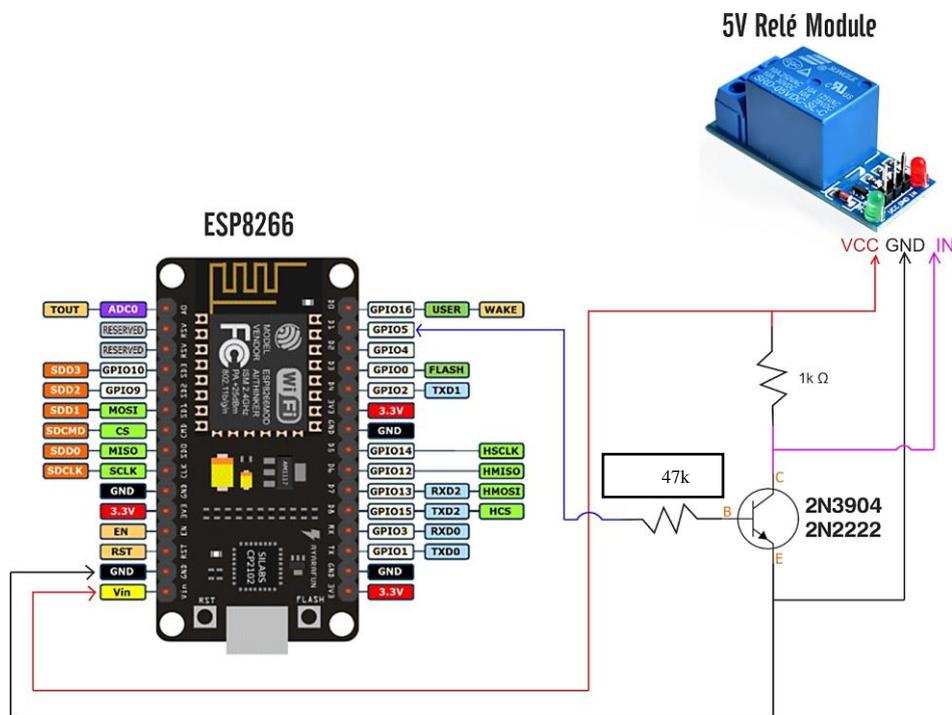


Figura 17. Circuito acoplador para relé de 5V (Alex, 2020)

Para la conexión de la placa con Ubidots se debe de conectar a internet mediante la tecnología seleccionada en este caso Wi-Fi, después de crear una cuenta en la plataforma de

Ubidots se procede a obtener el token, que es la llave para que la placa se conecte a nuestra cuenta, debemos de configurar en el sitio web las variables y widgets como se muestra en las imágenes.

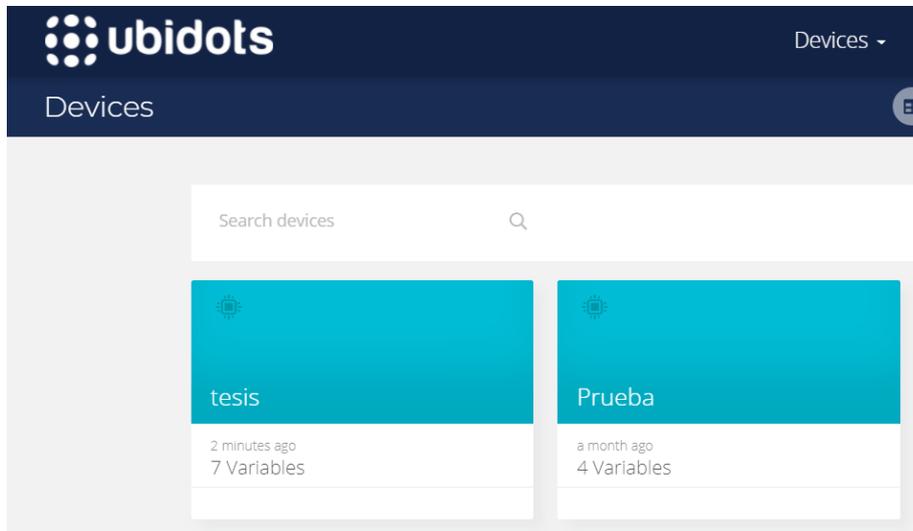


Figura 18. Visualización del proyecto en Ubidots (Autoría propia)

Añadimos el nombre de nuestro proyecto en este caso tesis

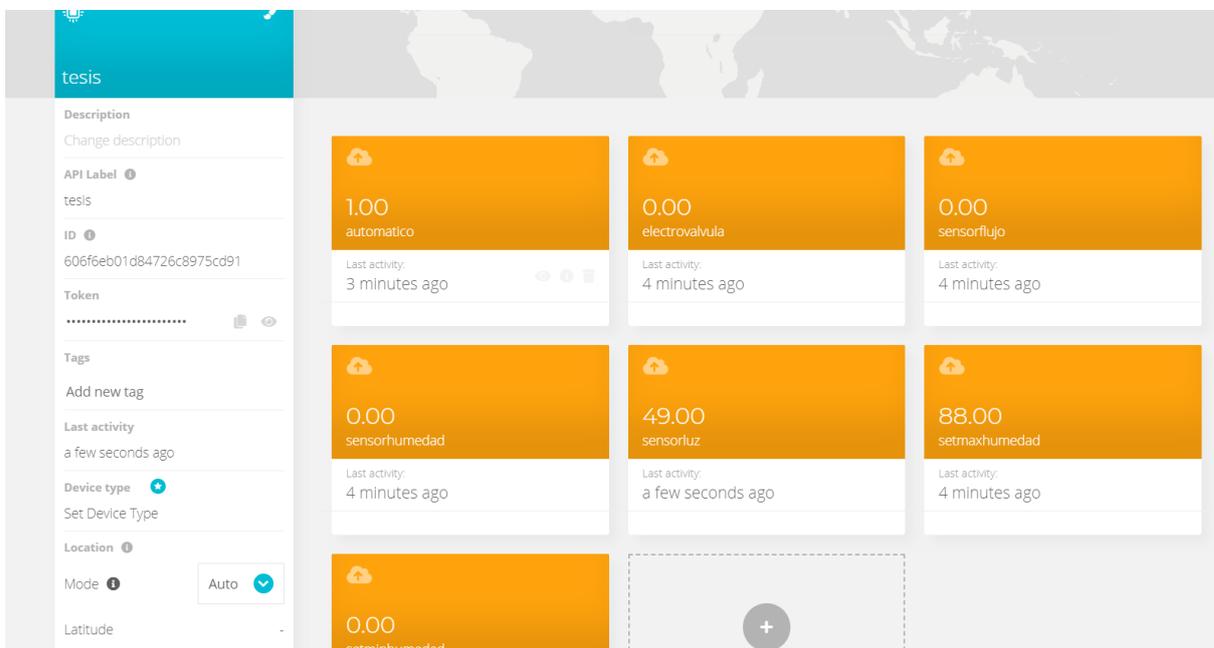


Figura 19. Variables del proyecto (Autoría propia)

Dentro del proyecto añadimos las variables de nuestro proyecto.

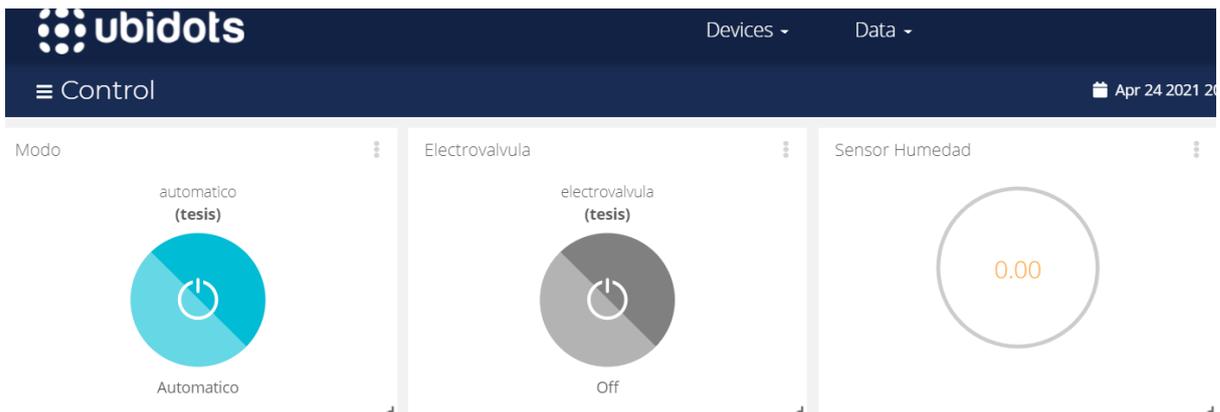


Figura 20. Configuración de Widgets (Autoría propia)

En la pestaña de Data, en dashboard creamos nuestro Widgets y los vinculamos a la respectiva variable. Elaboramos el diseño de la interfaz.

En el siguiente enlace podemos observar su diseño y funcionamiento:
<https://stem.ubidots.com/app/dashboards/public/dashboard/tePq4ky8aYTFcPifaDuPLBnUNFB8cRQbT4XRYA5A2ZU>

Datos	
Values Table	Values Table
DATE	AUTOMATICO (TESIS) LAST VALUE
DATE	ELECTROVALVULA (TESIS) LAST VALUE
05/01/2021 20:40	1.00
05/01/2021 20:40	0.00
05/01/2021 20:37	1.00
05/01/2021 19:43	0.00
04/29/2021 22:23	0.00
05/01/2021 19:36	0.00
04/29/2021 22:22	1.00
05/01/2021 19:34	0.00
04/26/2021 20:50	0.00
05/01/2021 19:33	0.00

Figura 21. Configuración de la visualización de los datos (Autoría propia)

Creamos una dashboard con los datos del proyecto, como podemos apreciar en el siguiente enlace:

https://stem.ubidots.com/app/dashboards/public/dashboard/16DZ0_HsUfShD3bxJqAPJDhBroRDB28t4OBpHRWPcM4

Después de tener lista la conexión MQTT y que el sensor de humedad de suelo detecte que se encuentra por debajo del umbral de humedad permitido, se debe de activar la electroválvula, para comprobar que la electroválvula se encuentra funcionando correctamente se puede fijar si el valor de humedad de suelo se encuentra aumentando o para asegurar que por la tubería se encuentra circulando agua se hace uso de un sensor de flujo, por lo que se podría asegurar un funcionamiento satisfactorio del sistema. Para fines prácticos de esta tesis se utiliza este sensor solo para saber si se encuentra fluyendo líquido por la tubería luego de abrir la electroválvula y poder comprobar que se está efectuando el riego desde cualquier lugar del mundo gracias a la tecnología IoT. Adicionalmente se agrega un sensor de luz, para dar información del contexto del lugar al usuario.

Finalmente, para informar al usuario de alguna anomalía en el sistema se realiza en la pestaña “eventos” de Ubidots, las alarmas al correo electrónico del cliente, por lo que si el sensor de humedad detecta valores fuera de los rangos configurados se envían las alertas instantáneamente. Cabe mencionar que cuando se desee modificar los valores mínimos y máximos de las alarmas, se debe de tomar en cuenta que estos son independientes del programa esto debido a que por ejemplo si deseo activar el riego cuando el valor de humedad del suelo este por debajo de 60% se activa la electroválvula y riega, en cambio sí activo la electroválvula pero ha habido alguna fuga o daño en el sistema entonces los valores de humedad pueden seguir

bajando por ejemplo hasta 50 % entonces manda una alerta al correo para notificar este uso inusual.



Figura 22. Máximos y mínimos de la función de riego. (Autoría propia)

Si desea cambiar los valores seteados de envío de notificación por correo se debe de ir a la pestaña de eventos y modificar.

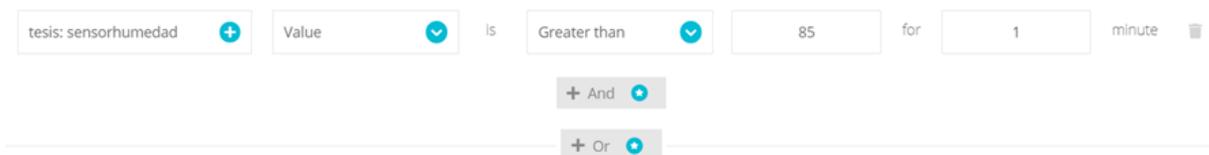


Figura 23. Mínimos y máximos de las alertas (Autoría propia)

Una vez configuradas las variables en la plataforma Ubidots, se realiza la programación en las tarjetas seleccionadas, la programación se muestra en los anexos 1 y 2. En ambas tarjetas se debe de configurar el token de la cuenta de Ubidots, su respectiva red wifi a la que se conectarán los dispositivos, su frecuencia de trabajo, las librerías correspondientes, se debe de agregar las variables con el mismo nombre que fueron declaradas en la plataforma, inicializar los tópicos a los que se va a subscribir cada dispositivo.

Debido a que el protocolo de comunicación es MQTT el programa no sigue un solo flujo de trabajo sino más bien se conecta al bróker y dependiendo de los valores de las variables que estas van obteniendo el sistema responde. Por ejemplo, el módulo del sensor lee el valor de la

humedad del suelo cada 1800 segundos (valor modificable por código) o si la válvula está activada y han pasado 30 segundos (valor modificable por código), para luego publicarlo en el bróker. (Ver Anexo 1)

Para el módulo principal tenemos que, si el modo automático esta activado y el valor de humedad de suelo es menor que el valor mínimo y también se debe de cumplir que el valor de la electroválvula esta apagada, entonces activa la electroválvula. (Ver anexo 2)

Al igual que si el módulo recibe las señales del bróker que dice que se encuentra activo el modo automático y el valor de humedad del suelo es mayor que el valor de humedad máximo permitido y la electroválvula continúa abierta, entonces se enviará la señal de cerrar el flujo de agua. Ofrece también la opción de configurar cada que tiempo se lea el valor de los sensores.

Finalmente, si desea regar los cultivos a su criterio puede desactivar el modo automático lo que le da la libertad de poder activar o desactivar la electroválvula, permitiendo suministrar la cantidad de agua que el agricultor crea conveniente.

3.2. Construcción del prototipo

Los pines en los gráficos son los utilizados para este proyecto y corresponden a la programación de los anexos, así puede dar mantenimiento en caso de alguna desconexión.

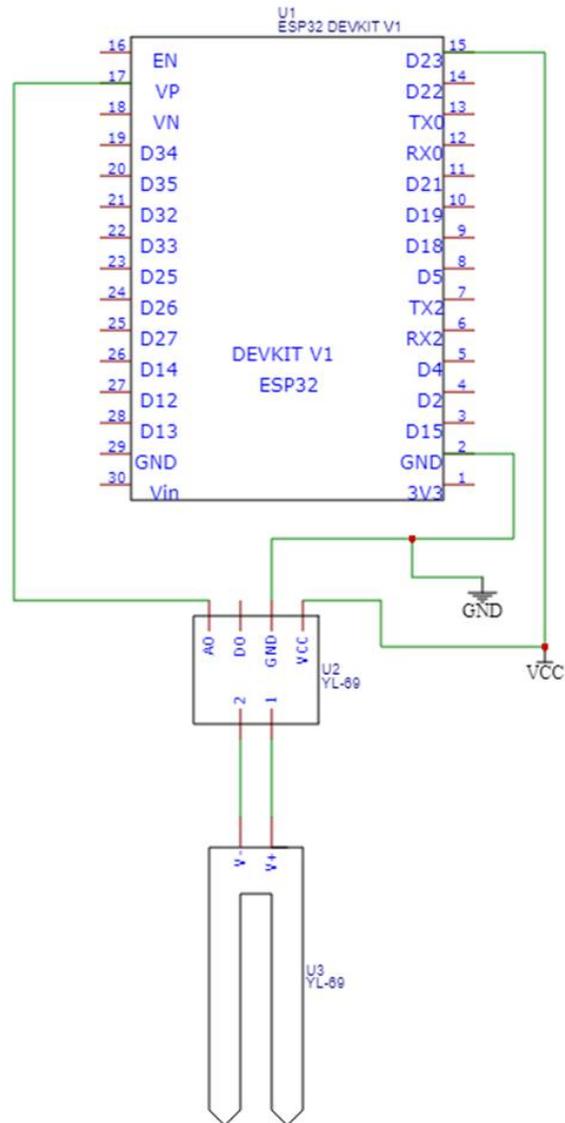


Figura 24. Diagrama de conexión del módulo sensor (Autoría propia)

Para el módulo de control se realizaron las siguientes conexiones, donde se puede apreciar leds indicadores, el led naranja muestra si se encuentra activo el modo automático, y el led rojo que la electroválvula se encuentra abierta. Las entradas de los sensores el sensor de flujo en el GPIO 2 y la fotorresistencia en el pin analógico A0.

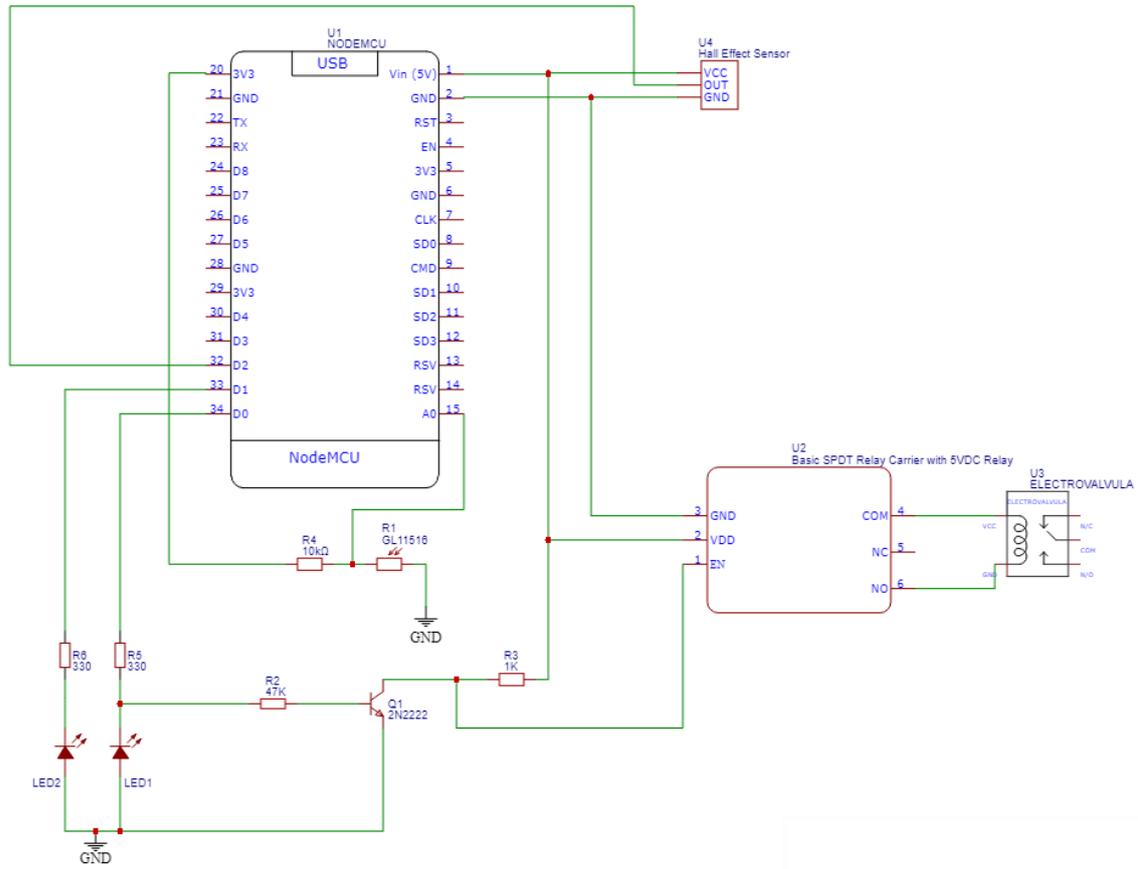


Figura 25. Conexión de módulo de control (Autoría propia)

3.3. Implementación del prototipo

Se hace la respectiva instalación del prototipo en el lugar seleccionado, añadiendo también la etapa de suministro de energía eléctrica. Se debe tener precaución de proteger los componentes de las adversidades del clima para su correcto funcionamiento, como podemos apreciar en las figuras. (Ver Anexo 3)

3.4. Validación

El Sistema funciona correctamente, permitiendo suministrar la cantidad de agua necesaria a las plantas en el momento que ellas lo requieren, automatizando la tarea del riego manual y permitiendo poder hacer cualquier modificación sobre los valores mínimos y máximos de

humedad desde cualquier lugar conectados a internet y en caso de requerir activar la electroválvula remotamente. Como podemos apreciar en la figura 26.

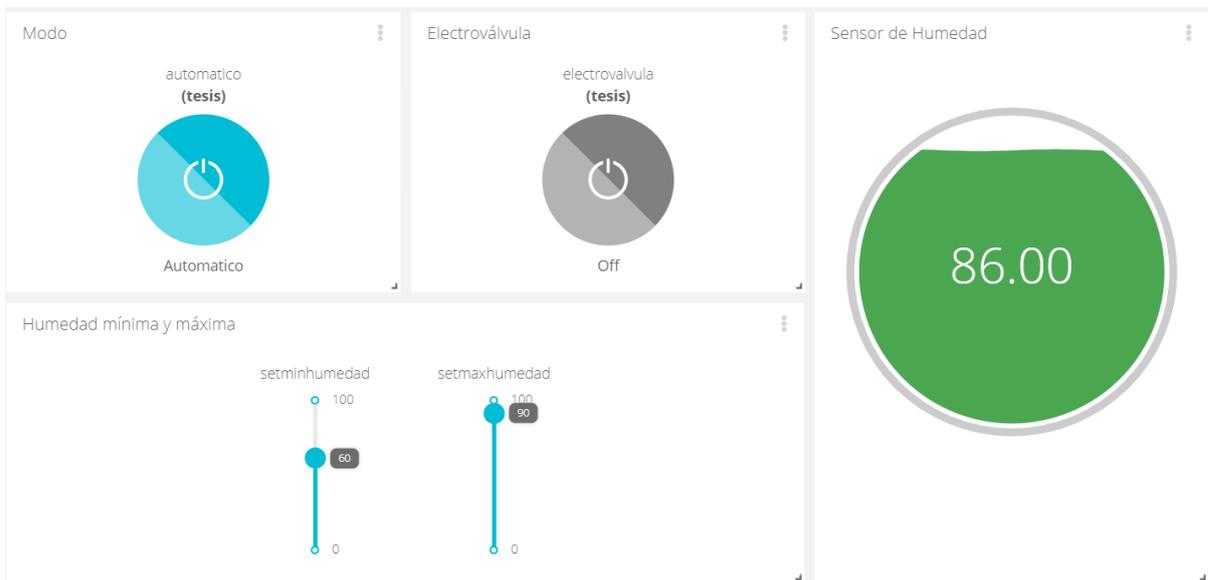


Figura 26. Sistema de monitoreo y control de irrigación

El valor de humedad se mantiene dentro de los límites que el usuario configure manteniéndose en el rango del mínimos y máximo permitidos. Cualquier sea el valor de humedad máxima (representada por la línea roja) y valor de humedad mínima (representada por la línea azul), la humedad del suelo (representado por la línea amarilla) se mantiene entre estos valores, otorgando un desarrollo óptimo en las plantas.



Figura 27. Validación de la humedad del suelo

Capítulo IV

4. Conclusiones y recomendaciones

4.1. Conclusiones

En este trabajo se construyó un sistema de riego inteligente que permite el monitoreo y su control utilizando tecnologías del internet de las cosas (IOT). La característica más importante del sistema de riego es su capacidad para ser controlado remotamente desde cualquier lugar con acceso a internet porque representa un avance en el monitoreo y control de los sistemas de riego automáticos. Lo que más ayudó al desarrollo del sistema de riego automático con tecnología IoT fue la plataforma Ubidots porque gestiona la información en la nube y ayuda a presentar la aplicación web responsiva al usuario. Parte de lo complejo para la elaboración del sistema de riego fue encontrar disponibilidad de las placas compatibles con tecnologías IoT por la coyuntura que se vive por la pandemia.

Se identificó con ayuda del propietario de la finca que el método de irrigación al aire libre que se utilizaría en este prototipo sería el riego por goteo debido a sus necesidades.

Se diseñó el sistema de riego que cumple con los requerimientos y restricciones del usuario, satisfaciendo la necesidad del riego automático. Lo más destacable del diseño del sistema de riego automático es su fácil interfaz de usuario porque permite la utilización de la aplicación web por personas con mínimos conocimientos de tecnología. Para generar el diseño del sistema de riego fue de gran ayuda la utilización de hardware y software libre porque existen librerías que logran conectar los diferentes componentes con relativa facilidad. Entre lo que causó mayor dificultad para el diseño del sistema de riego inteligente fue limitar las funciones que tendría debido a que se debe de cumplir con los requerimientos y restricciones del usuario. Además, se debe de considerar que las placas de desarrollo tienen relativamente poca memoria.

En esta tesis se construyó e implementó el prototipo de irrigación en una finca del barrio Thomas de Sosoranga-Loja-Ecuador. Parte fundamental en este proceso fue la apertura de los dueños de la finca para su instalación porque permite hacer las pruebas necesarias y corregir los percances del proceso.

Finalmente se evaluó el sistema automático de riego con tecnología IOT en un espacio dispuesto para el riego por goteo. Donde se comprobó el correcto funcionamiento del prototipo debido a que logró mantener los valores apropiados de humedad del suelo para que la planta tenga un desarrollo idóneo y a la vez se ahorre agua en el proceso. Las gráficas que se pueden obtener en la plataforma de Ubidots ayudaron en este proceso porque se puede identificar claramente el funcionamiento de los componentes a través del tiempo.

4.2. Recomendaciones

Se debe de tener en cuenta que los sensores de tipo resistivo tienden a sufrir de corrosión electrolítica por lo que se recomienda alimentar el módulo electrónico automáticamente solo al momento de tomar las mediciones, apagándolo inmediatamente al terminar y así minimizarla. También se puede intercambiar las conexiones del elemento sensor periódicamente para que los dos electrodos roten de polaridad. Aumentando la vida útil del mismo.

Para proyectos más grandes o comerciales hacer uso de la versión de pago de Ubidots para poder manejar mayor cantidad de datos, lo que permite aumentar la cantidad de sensores, en caso de que la aplicación lo requiera, o en su defecto hacer usos de las plataformas pago de IoT que se mencionan en el trabajo.

Bibliografía

- Bongiovanni, R., Mantovani, E., Best, S., & Roel, Á. (2006). *Agricultura de precisión: integrando conocimientos para una agricultura moderna y sustentable*. Procisur/IICA.
- Das, S. (2020). Modeling and Simulation of Mechatronic Systems using Simscape. In *Modeling and Simulation of Mechatronic Systems using Simscape*. Morgan & Claypool.
<https://ieeexplore.ieee.org/document/9036320>
- De Silva, C. W. (2010). *Mechatronics: a foundation course*. CRC press.
- García, L., Parra, L., Jimenez, J. M., Lloret, J., & Lorenz, P. (2020). IoT-Based Smart Irrigation Systems: An Overview on the Recent Trends on Sensors and IoT Systems for Irrigation in Precision Agriculture. *Sensors*, 20(4), 1042.
- Jensen, M. E., Burman, R. D., & Allen, R. G. (1990). *Evapotranspiration and irrigation water requirements*.
- Jones, H. G. (2004). Irrigation scheduling: advantages and pitfalls of plant-based methods. *Journal of Experimental Botany*, 55(407), 2427–2436.
- Karnopp, D. C., Margolis, D. L., & Rosenberg, R. C. (2012). *System dynamics: modeling, simulation, and control of mechatronic systems*. John Wiley & Sons.
- Lennox, R. J., Crook, D. A., Moyle, P. B., Struthers, D. P., & Cooke, S. J. (2019). Toward a better understanding of freshwater fish responses to an increasingly drought-stricken world. *Reviews in Fish Biology and Fisheries*, 29(1), 71–92.
- Ogata, K. (2010). *Ingeniería de control moderna Quinta Edición*. Pearson, México.
- Pereira, L. S., Oweis, T., & Zairi, A. (2002). Irrigation management under water scarcity. *Agricultural Water Management*, 57(3), 175–206.
- Rajalakshmi, P., & Mahalakshmi, S. D. (2016). IOT based crop-field monitoring and irrigation automation. *2016 10th International Conference on Intelligent Systems and Control*

(ISCO), 1–6.

Vermeulen, S., Zougmore, R., Wollenberg, E., Thornton, P., Nelson, G., Kristjanson, P., Kinyangi, J., Jarvis, A., Hansen, J., & Challinor, A. (2012). Climate change, agriculture and food security: a global partnership to link research and action for low-income agricultural producers and consumers. *Current Opinion in Environmental Sustainability*, 4(1), 128–133.

Linkografía

Alex. (31 de 03 de 2020). *Alex7 Tutoriales*. Obtenido de <https://alex7tutoriales.blogspot.com/2020/03/rele-5v-con-esp8266-y-firebase-conexion.html>

Bishop, R. H. (2018). *The Mechatronics Handbook*. CRC Press.

Caldas. (2021). *Electronicos*. Obtenido de Caldas: <https://www.electronicoscaldas.com/es/sensores-de-humedad-lluvia-inundacion/461-sensor-de-humedad-en-suelo-yl-69.html>

ElectronicsComp. (2021). *ElectronicsComp*. Obtenido de <https://www.electronicscamp.com/0.5-inch-water-flow-sensor-yf-s201>

Electronilab. (2020). *Electronilab*. Obtenido de <https://cdn.electronilab.co/wp-content/uploads/2016/02/NodeMCU-%E2%80%93-Board-de-desarrollo-con-m%C3%B3dulo-ESP8266-WiFi-y-Lua-4.jpg>: <https://electronilab.co/tienda/nodemcu-board-de-desarrollo-con-esp8266-wifi-y-lua/>

Garriga Domínguez, A. (2016). *Nuevos retos para la protección de datos personales. En la era del Big Data y de la computación ubicua*. Dykinson.

Gozzer, S. (27 de abril de 2020). *BBC News Mundo*. Recuperado el 09 de julio de 2020, de <https://www.bbc.com/mundo/noticias-america-latina-52413808>

Klimov, A. (2020). *developer*. Obtenido de [alexanderklimov.ru: http://developer.alexanderklimov.ru/arduino/esp32/](http://developer.alexanderklimov.ru/arduino/esp32/)

Lopez, M. (14 de 05 de 2020). *iotconsulting*. Obtenido de https://www.youtube.com/watch?v=bu5pYdLsLc0&ab_channel=IoTConsulting: <https://iotconsulting.tech/blog/>

Valparaíso, T. (s.f.). *Maxelectronica*. Obtenido de <https://maxelectronica.cl/prototipo/328-electrovalvula-solenoide-12vdc-12-pulgada-normalmente-cerrada.html>

Anexos

Anexo 1. Programación del módulo principal o de control

```

/*****
* Incluimos Librerías
*****/
#include <PubSubClient.h>
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>

/*****
* Definimos constantes
*****/
#define WIFISSID "SUPERNEXUS-Alvarado" // Poner tu WifiSSID aquí
#define PASSWORD "****" // Poner la contraseña de wifi

#define TOKEN "BBFF-p7rZf5MR8k7yZNHd0ueTGT93Yqrck5" // TOKEN de ubidots
rogger09roman

#define VARIABLE_LABEL1 "sensorhumedad" // asignamos las etiquetas de las variables
para publicar y suscribir
#define VARIABLE_LABEL2 "automatico"
#define VARIABLE_LABEL3 "electrovalvula"
#define VARIABLE_LABEL4 "setminhumedad"
#define VARIABLE_LABEL5 "setmaxhumedad"
#define VARIABLE_LABEL6 "sensorflujo"
#define VARIABLE_LABEL7 "sensorluz"

#define DEVICE_LABEL "tesis" // asignamos el nombre del dispositivo a suscribir
#define MQTT_CLIENT_NAME "roggerroma" // nombre del cliente MQTT, put a Random
ASCII

#define electrovalvula 16 //definimos los gpio a utilizar
#define maautomatico 5
#define SENSOR 4

/*****
* Definimos variables
*****/
char mqttBroker[] = "industrial.api.ubidots.com";// nombre del broker MQTT

int modo = 1; //Automático = 1 o manual = 0
float modoval = 0; //Válvula encendida = 1 o válvula apagada = 0
float modomax = 80.0; //seteo por defecto del valor máximo de humedad

```

```

float modomin = 40.0; //seteo por defecto del valor mínimo de humedad
float modohum = 40.0; // humedad

int tamtopic=0; //tamaño del topic
String topico;
String valor1;
int valor;

int times; // tiempos para trabajar
int times2;
int times3;
int times4;

char str_val1[6]; // cadenas de caracteres para enviar
char str_val2[6];
char str_val3[6];
char str_val4[6];
char topic[150];
char payload[400];

float value; // variable para valores de entrada de los diferentes topics

long currentMillis = 0; //variables del sensor de flujo
long previousMillis = 0;
int interval = 1000;
float calibrationFactor = 4.5;
volatile byte pulseCount;
byte pulse1Sec = 0;
float flowRate;
unsigned long flowMilliLitres;
unsigned int totalMilliLitres;
float flowLitres;
float totalLitres;

/*****
 * Inicializamos los constructores para los objetos
 *****/
ESP8266WiFiMulti WiFiMulti;
WiFiClient ubidots;
PubSubClient client(ubidots);

/*****
 * Funciones auxiliares
 *****/

//funcion que permite recibir los cambios en los topics suscritos
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Mensaje recibido en [");

```

```

Serial.print(topic);
Serial.print("] ");
value = btOf(payload, length); // llama a la función para obtener los valores
//Serial.println(value); // valores recibidos

for (int i=0;i<length;i++) { //almacena en un char los valores recibidos
Serial.print((char)payload[i]);
valor1=valor1+(char)payload[i];

}
Serial.println();
//Serial.print(valor1);

tamtopic=strlen(topic); //determina el tamaño del topic recibido

for (int i=0;i<tamtopic;i++) { // acomoda como string al topic recibido
//Serial.print((char)topic[i]); // des comentar para ver el topic recibido como string
topico=topico+(String)topic[i];
}

//Selección de los topics entrantes para modificar variables internas
if (topico == "/v1.6/devices/tesis/automatico/lv"){
  if (valor1=="0.0"){
    digitalWrite(mautomatico, LOW);
    modo = 0;
  }else{
    digitalWrite(mautomatico, HIGH);
    modo = 1;
  }
}

if (topico == "/v1.6/devices/tesis/electrovalvula/lv"){
  if (valor1=="0.0"){
    digitalWrite(electrovalvula, LOW);
    modoval = 0.00;
    delay(1);
    totalMilliLitres =0;
    totalLitres =0;
  }else{
    digitalWrite(electrovalvula, HIGH);
    modoval = 1.00;
  }
}

if (topico == "/v1.6/devices/tesis/setmaxhumedad/lv"){
  modomax=value;
  Serial.print("El maximo de humedad se ha establecido en ");
  Serial.println(modomax);
}

```

```

if (topico == "/v1.6/devices/tesis/setminhumedad/lv"){
  modomin=value;
  Serial.print("El minimo de humedad se ha establecido en ");
  Serial.println(modomin);
}
if (topico == "/v1.6/devices/tesis/sensorhumedad/lv"){
  modohum=value;
  Serial.print("La humedad es de: ");
  Serial.println(modohum);
}
topico=""; // ponemos las variables como vacías
valor1="";
}

// Función de reconexión
void reconnect() {
  // Bucle hasta que volvamos a conectarnos
  while (!client.connected()) {
    Serial.println("Intentando la conexion MQTT...");

    // intentando la conexión
    if (client.connect(MQTT_CLIENT_NAME, TOKEN,"")) {
      Serial.println("Conectado a cliente MQTT");
      modoval = 0.00;
      dtostrf(modoval, 4, 2, str_val2);

      sprintf(topic, "%s", ""); // Limpiamos el contenido del topic
      sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE_LABEL); // agregamos el dispositivo
a utilizar
      sprintf(payload, "%s", ""); // Cleans the payload content
      sprintf(payload, "{\"%s\":\"", VARIABLE_LABEL3); // agregamos el nombre de la
variable
      sprintf(payload, "%s {\"value\": %s", payload, str_val2); // agregamos el valor
      sprintf(payload, "%s } }", payload); // cerramos el formato para publicar
      Serial.println("Electrovalvula apagado por seguridad"); //Apagamos la válvula por
seguridad
      client.publish(topic, payload); // publicamos los datos
      modoval = 0.00;
      delay(1000);
    } else {
      Serial.print("Falla, reconectando=");
      Serial.print(client.state());
      Serial.println(" intentandolo de nuevo en 2 segundos");
      delay(2000);
    }
  }
}

//Contador de pulsos del sensor de flujo
void IRAM_ATTR pulseCounter()
{

```

```

pulseCount++;
}

// convertir desde una matriz los caracteres a flotantes
float btof(byte * payload, unsigned int length) {
  char * demo_ = (char *) malloc(sizeof(char) * 10);
  for (int i = 0; i <length; i++) {
    demo_[i] = payload[i];
  }
  return atof(demo_);
}

/*****
* Función principal
*****/

void setup() {
  Serial.begin(115200);// inicializamos los baudios
  //defimos el modo de los pines
  //pinMode(A0, INPUT);
  pinMode(electrovalvula, OUTPUT);
  pinMode(mautomatico, OUTPUT);
  pinMode(SENSOR, INPUT_PULLUP);

  //conexion wifi
  WiFiMulti.addAP(WIFISSID, PASSWORD);
  Serial.println();
  Serial.print("Esperando la conexion WIFI... ");

  while(WiFiMulti.run() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  Serial.println("");
  Serial.println("WiFi conectado");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());// direccion ip del dispositivo
  client.setServer(mqttBroker, 1883); //broker y puerto a conectarse
  client.setCallback(callback); //llamada al subscritor

  pulseCount = 0;
  flowRate = 0.0;
  flowMilliLitres = 0;
  totalMilliLitres = 0;
  previousMillis = 0;

  attachInterrupt(digitalPinToInterrupt(SENSOR), pulseCounter, FALLING);//interrupción
del contador de pulsos
}

```

```

void loop() {
  //intentando conectarse al cliente MQTT
  if (!client.connected()) {
    reconnect();

    // Se suscribe para obtener el valor de las diferentes variables
    char topicToSubscribe[200];
    sprintf(topicToSubscribe, "%s", "");
    sprintf(topicToSubscribe, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
    sprintf(topicToSubscribe, "%s/%s/lv", topicToSubscribe, VARIABLE_LABEL1);
  //sensor de humedad
    Serial.println("subscribing to topic:");
    Serial.println(topicToSubscribe);
    client.subscribe(topicToSubscribe);

    sprintf(topicToSubscribe, "%s", ""); // Cleans the content of the char
    sprintf(topicToSubscribe, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
    sprintf(topicToSubscribe, "%s/%s/lv", topicToSubscribe,
VARIABLE_LABEL2); //automatico
    Serial.println(topicToSubscribe);
    client.subscribe(topicToSubscribe);

    sprintf(topicToSubscribe, "%s", ""); // Cleans the content of the char
    sprintf(topicToSubscribe, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
    sprintf(topicToSubscribe, "%s/%s/lv", topicToSubscribe,
VARIABLE_LABEL3); //electrovalvula
    Serial.println(topicToSubscribe);
    client.subscribe(topicToSubscribe);

    sprintf(topicToSubscribe, "%s", ""); // Cleans the content of the char
    sprintf(topicToSubscribe, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
    sprintf(topicToSubscribe, "%s/%s/lv", topicToSubscribe,
VARIABLE_LABEL4); //minimo de humedad
    Serial.println(topicToSubscribe);
    client.subscribe(topicToSubscribe);

    sprintf(topicToSubscribe, "%s", ""); // Cleans the content of the char
    sprintf(topicToSubscribe, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
    sprintf(topicToSubscribe, "%s/%s/lv", topicToSubscribe,
VARIABLE_LABEL5); //maximo de humedad
    Serial.println(topicToSubscribe);
    client.subscribe(topicToSubscribe);

    float sensorlu = analogRead(0);
    int sensorf = map(sensorlu, 1024, 0, 0, 100);
    dtostrf(sensorf, 4, 2, str_val1);

    sprintf(topic, "%s", ""); // Cleans the topic content

```

```

sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
sprintf(payload, "%s", ""); // Cleans the payload content
sprintf(payload, "{\"%s\":", VARIABLE_LABEL7); // Adds the variable label
sprintf(payload, "%s {\"value\": %s", payload, str_val1); // Adds the value
sprintf(payload, "%s } }", payload); // Closes the dictionary brackets
Serial.println("publicando sensor de luz");
client.publish(topic, payload);
times=0;
Serial.println("fin de subscripcion");
}

if ( times == 900){// times es el tiempo en segundos que se repite esta acción
float sensorlu = analogRead(0);// lectura del sensor de luminosidad
int sensorf = map(sensorlu, 1024, 0, 0, 100);
dtostrf(sensorf, 4, 2, str_val1);

sprintf(topic, "%s", ""); // Cleans the topic content
sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
sprintf(payload, "%s", ""); // Cleans the payload content
sprintf(payload, "{\"%s\":", VARIABLE_LABEL7); // Adds the variable label
sprintf(payload, "%s {\"value\": %s", payload, str_val1); // Adds the value
sprintf(payload, "%s } }", payload); // Closes the dictionary brackets
Serial.println("publicando sensor de luz");
client.publish(topic, payload);
times=0;
}
times++;

if ((modo==1) && (modohum<modomin)&&(modoval==0.00)){

if (( times2 == 5)&&(modoval==0.00)){
digitalWrite(electrovalvula, HIGH);
modoval = 1.00;
dtostrf(modoval, 4, 2, str_val2);

sprintf(topic, "%s", ""); // Cleans the topic content
sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
sprintf(payload, "%s", ""); // Cleans the payload content
sprintf(payload, "{\"%s\":", VARIABLE_LABEL3); // Adds the variable label
sprintf(payload, "%s {\"value\": %s", payload, str_val2); // Adds the value

sprintf(payload, "%s } }", payload); // Closes the dictionary brackets
Serial.println("Publicando la activacion de la electrovalvula");
client.publish(topic, payload);
times2=0;
delay(1000);
}
times2++;
}
}

```

```

if ((modo==1) && (modohum>modomax) && (modoval==1.00)){
  //Serial.println(modo);
  //Serial.println(modomax);
  //Serial.println(modohum);
  times3++;
  if (( times3 == 5)&&(modoval==1.00)){
    digitalWrite(electrovalvula, LOW);
    //Serial.println(modoval);
    modoval = 0.00;
    //Serial.println(modoval);
    dtostrf(modoval, 4, 2, str_val3);

    sprintf(topic, "%s", ""); // Cleans the topic content
    sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE_LABEL);

    sprintf(payload, "%s", ""); // Cleans the payload content
    sprintf(payload, "{\"%s\":", VARIABLE_LABEL3); // Adds the variable label
    sprintf(payload, "%s {\"value\": %s", payload, str_val3); // Adds the value

    sprintf(payload, "%s } }", payload); // Closes the dictionary brackets
    Serial.println("Publicando la desactivacion de la electrovalvula");
    client.publish(topic, payload);
    times3=0;
    delay(1000);

  }

}

// lector de pulsos del sensor de flujo
currentMillis = millis();
if (modoval==1.00){
  if (currentMillis - previousMillis > interval)
  {

    pulse1Sec = pulseCount;
    pulseCount = 0;

    // Because this loop may not complete in exactly 1 second intervals, we calculate
    // the number of milliseconds that have passed since the last execution and use
    // that to scale the output. We also apply the calibrationFactor to scale the output
    // based on the number of pulses per second per units of measure (litres/minute in
    // this case) coming from the sensor.
    flowRate = ((1000.0 / (millis() - previousMillis)) * pulse1Sec) / calibrationFactor;
    previousMillis = millis();

    // Divide the flow rate in litres/minute by 60 to determine how many litres have
    // passed through the sensor in this 1 second interval, then multiply by 1000 to
    // convert to millilitres.
    flowMilliLitres = (flowRate / 60) * 1000;
  }
}

```

```

flowLitres = (flowRate / 60);

// Add the millilitres passed in this second to the cumulative total
totalMilliLitres += flowMilliLitres;
totalLitres += flowLitres;

// Print the flow rate for this second in litres / minute

//float sensorfl = flowRate;
times4++;
}}
//publicacion del sensor de flujo
if ( times4 == 30){

    Serial.print("Tasa de flujo: ");
    Serial.print(float(flowRate)); // Print the integer part of the variable
    Serial.print("L/min");
    Serial.print("\t");    // Print tab space

// Print the cumulative total of litres flowed since starting
    Serial.print("Cantidad de líquido de salida: ");
    Serial.print(totalMilliLitres);
    Serial.print("mL / ");
    Serial.print(totalLitres);
    Serial.println("L");
    dtostrf(flowRate, 4, 2, str_val4);

    sprintf(topic, "%s", ""); // Cleans the topic content
    sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
    sprintf(payload, "%s", ""); // Cleans the payload content
    sprintf(payload, "{\"%s\":", VARIABLE_LABEL6); // Adds the variable label
    sprintf(payload, "%s {\"value\": %s", payload, str_val4); // Adds the value
    sprintf(payload, "%s } }", payload); // Closes the dictionary brackets
    Serial.println("publicando sensor de flujo");
    client.publish(topic, payload);
    times4=0;
}
client.loop();
delay(1000);
}

```

Anexo 2. Programación del módulo sensor

```

#include <WiFi.h>
#include <PubSubClient.h>

#define WIFISSID "SUPERNEXUS-Alvarado" // Poner tu WifiSSID aquí
#define PASSWORD "*****" // Poner la contraseña de wifi

// #define TOKEN "BBFF-KEZPtMgqKckBccJqIolYAYcInzNnUf" //TOKEN
RALVARADO
#define TOKEN "BBFF-p7rZf5MR8k7yZNHd0ueTGT93Yqrck5" //TOKEN de ubidots
rogger09roman
#define MQTT_CLIENT_NAME "roggerrom4n" // MQTT client Name, nombre del cliente
MQTT, please enter your own 8-12 alphanumeric character ASCII string;
//it should be a random and unique ascii string and different from all other devices

/*****
 * Define Constantes
 *****/
#define VARIABLE_LABEL "sensorhumedad" // asignamos las etiquetas de las variables
para publicar y subscribir
#define VARIABLE_LABEL2 "setminhumedad"
#define VARIABLE_LABEL3 "setmaxhumedad"
#define VARIABLE_LABEL4 "automatico"
#define VARIABLE_LABEL5 "electrovalvula"
#define DEVICE_LABEL "tesis" // asignamos el nombre del dispositivo

#define led 23 // definimos los gpio a utilizar

long lastMsg=0;
char mqttBroker[] = "things.ubidots.com"; // funciona mejor con este broker
//char mqttBroker[] = "industrial.api.ubidots.com";
char payload[100];
char topic[150];
//char topicSubscribe[150];
char str_sensor[10];

int times = 0; //tiempos para trabajar
int timeselec = 0;
int tamtopic=0;
String topico;

float modomax = 80.0; //seteo de valores predeterminados
float modomin = 60.0;
float modo;
float modoelec;
float value;

/*****

```

```

* Auxiliar Functions
*****/
WiFiClient ubidots;
PubSubClient client(ubidots);
//funcion reconectar
void reconnect() {
  // Loop para reconexión
  while (!client.connected()) {
    Serial.println("Intentando la conexion MQTT...");

    // Intentando conectarte
    if (client.connect(MQTT_CLIENT_NAME, TOKEN, "")) {
      Serial.println("Conectado");

    } else {
      Serial.print("Falla, rc=");
      Serial.print(client.state());
      Serial.println("Intentandolo de nuevo en 2 segundos");
      delay(2000);
    }
  }
}
//función para lectura de los tópicos recibidos
void callback(char* topic, byte* payload, unsigned int length) {
  //digitalWrite(led, HIGH);
  Serial.print("Mensaje recibido en [");
  Serial.print(topic);
  Serial.print("] ");

  value = btOf(payload, length);

  tamtopic=strlen(topic);
  for (int i=0;i<tamtopic;i++) {
    //Serial.print((char)topic[i]);// des comentar para ver el topic recibido como string
    topico=topico+(String)topic[i];
  }
  // selecciond e topics recibidos
  if (topico == "/v1.6/devices/tesis/setminhumedad/lv"){
    modomin=value;
    //Serial.println(modomin);
  }
  if (topico == "/v1.6/devices/tesis/setmaxhumedad/lv"){
    modomax=value;
    //Serial.println(modomin);
  }

  if (topico == "/v1.6/devices/tesis/automatico/lv"){
    modo=value;
    //Serial.println(modo);
  }
}

```

```

if (topico == "/v1.6/devices/tesis/electrovalvula/lv"){
  modoelec=value;
  //Serial.println(modoelec);
  timeselec=0;
}

//

Serial.write(payload, length);// para publicar el archivo byte como string
Serial.println();
//digitalWrite(led, LOW);
topico="";
value=0;
}

// funcion para emitir desde una matriz de caracteres a valor flotante.
float btof(byte * payload, unsigned int length) {
  char * demo_ = (char *) malloc(sizeof(char) * 10);
  for (int i = 0; i <length; i++) {
    demo_[i] = payload[i];
  }
  return atof(demo_);
}

/*****
* Main Functions
*****/

void setup() {
  Serial.begin(115200);
  WiFi.begin(WIFISSID, PASSWORD);
  //definimos los pines
  pinMode(led, OUTPUT);

  Serial.println();
  Serial.print("Esperando la conexion de WiFi...");

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  Serial.println("");
  Serial.println("WiFi Connectado");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  client.setServer(mqttBroker, 1883);
  client.setCallback(callback);

  //times = 1800;
}

```

```

void loop() {
  if (!client.connected()) {
    reconnect();
    char topicToSubscribe[200];
    //subscripcion de topicos
    Serial.println("Subscribiendo topicos:");
    sprintf(topicToSubscribe, "%s", ""); // Cleans the content of the char
    sprintf(topicToSubscribe, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
    sprintf(topicToSubscribe, "%s/%s/lv", topicToSubscribe, VARIABLE_LABEL2);
    Serial.println(topicToSubscribe);
    client.subscribe(topicToSubscribe);

    sprintf(topicToSubscribe, "%s", ""); // Cleans the content of the char
    sprintf(topicToSubscribe, "/v1.6/devices/%s/%s/lv", DEVICE_LABEL,
VARIABLE_LABEL3);
    Serial.println(topicToSubscribe);
    client.subscribe(topicToSubscribe);

    sprintf(topicToSubscribe, "%s", ""); // Cleans the content of the char
    sprintf(topicToSubscribe, "/v1.6/devices/%s/%s/lv", DEVICE_LABEL,
VARIABLE_LABEL4);
    Serial.println(topicToSubscribe);
    client.subscribe(topicToSubscribe);

    sprintf(topicToSubscribe, "%s", ""); // Cleans the content of the char
    sprintf(topicToSubscribe, "/v1.6/devices/%s/%s/lv", DEVICE_LABEL,
VARIABLE_LABEL5);
    Serial.println(topicToSubscribe);
    client.subscribe(topicToSubscribe);

    sprintf(topic, "%s", ""); // Cleans the topic content
    sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
    sprintf(payload, "%s", ""); // Cleans the payload
    sprintf(payload, "{\"%s\"":", VARIABLE_LABEL); // Adds the variable label
    digitalWrite(led, HIGH);
    delay(5);
    int sensorh = analogRead(36);

    digitalWrite(led, LOW);
    delay(5);
    Serial.println(sensorh);

    int sensor = map(sensorh, 4095, 700, 0, 100);
    Serial.print("El porcentaje de humedad es: ");
    Serial.println(sensor);
    dtostrf(sensor, 4, 2, str_sensor);
    sprintf(payload, "%s {\"value\": %s}", payload, str_sensor); // Adds the value
    Serial.println("Publicando la humedad del suelo");
    client.publish(topic, payload);
  }
}

```

```

    //fin de la subscripcion y publicacion
}
//lectura del sensor
if (( times == 1800)||((modoelec==1.0)&(timeselec==30))){

    sprintf(topic, "%s", ""); // Cleans the topic content
    sprintf(topic, "%s%s", "/v1.6/devices/", DEVICE_LABEL);
    sprintf(payload, "%s", ""); // Cleans the payload
    sprintf(payload, "{\"%s\":", VARIABLE_LABEL); // Adds the variable label
    digitalWrite(led, HIGH);
    delay(50);
    int sensorh = analogRead(36);
    digitalWrite(led, LOW);
    Serial.println(sensorh);

    int sensor = map(sensorh, 4095, 700, 0, 100);
    Serial.print("El porcentaje de humedad es: ");Serial.println(sensor);
    dtostrf(sensor, 4, 2, str_sensor);
    sprintf(payload, "%s {\"value\": %s}", payload, str_sensor); // Adds the value
    Serial.println("Publicando la humedad del suelo");
    client.publish(topic, payload);

    times=0;
    timeselec=0;
}

timeselec++;
times++;
client.loop();
delay(1000);
}

```

Anexo 3. Instalación



Figura 28. Derivación del suministro de agua



Figura 29. Instalación del módulo de control



Figura 30. Instalación de la electroválvula y sensor de flujo



Figura 31. Instalación del sistema para riego por goteo



Figura 32. Módulo sensor



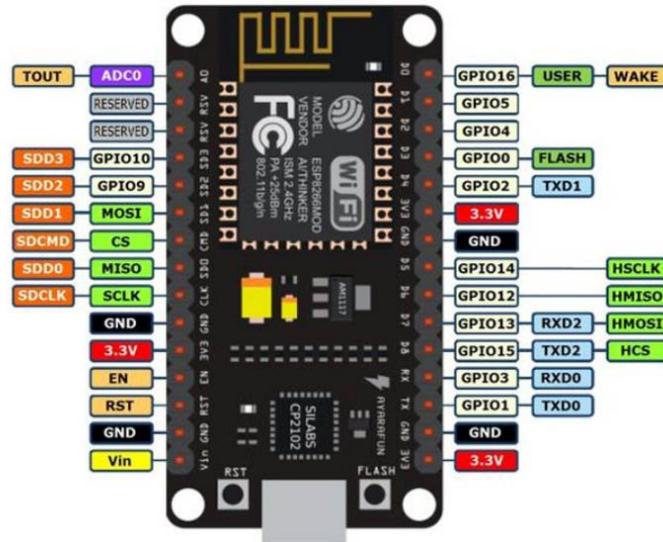
Figura 33. Sensor de humedad junto a los goteros



Figura 34. Sistema automático de monitoreo y control de irrigación con IoT

Apéndices

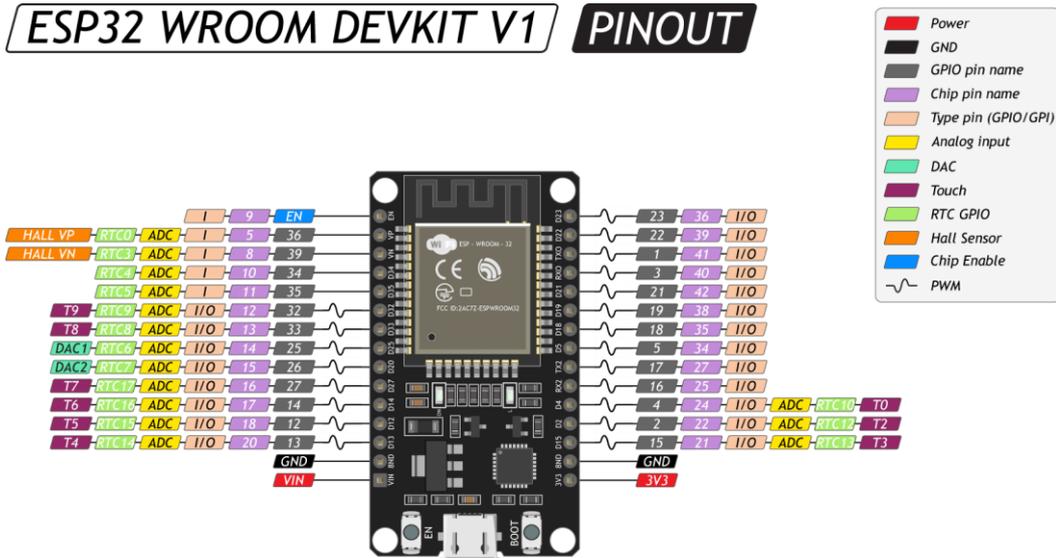
Apéndice A. Datasheet de NodeMCU ESP8266



Pin Category	Name	Description
Power	Micro-USB, 3.3V, GND, Vin	<p>Micro-USB: NodeMCU can be powered through the USB port</p> <p>3.3V: Regulated 3.3V can be supplied to this pin to power the board</p> <p>GND: Ground pins</p> <p>Vin: External Power Supply</p>
Control Pins	EN, RST	The pin and the button resets the microcontroller
Analog Pin	A0	Used to measure analog voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to GPIO16	NodeMCU has 16 general purpose input-output pins on its board
SPI Pins	SD1, CMD, SDO, CLK	NodeMCU has four pins available for SPI communication.
UART Pins	TXD0, RXD0, TXD2, RXD2	NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C.

Apéndice B. Datasheet de ESP32 – DevKit

ESP32 WROOM DEVKIT V1 PINOUT



👁 Interface on all GPIO pins: 3×SPI, 3×UART, 2×I²C and 2×I²S

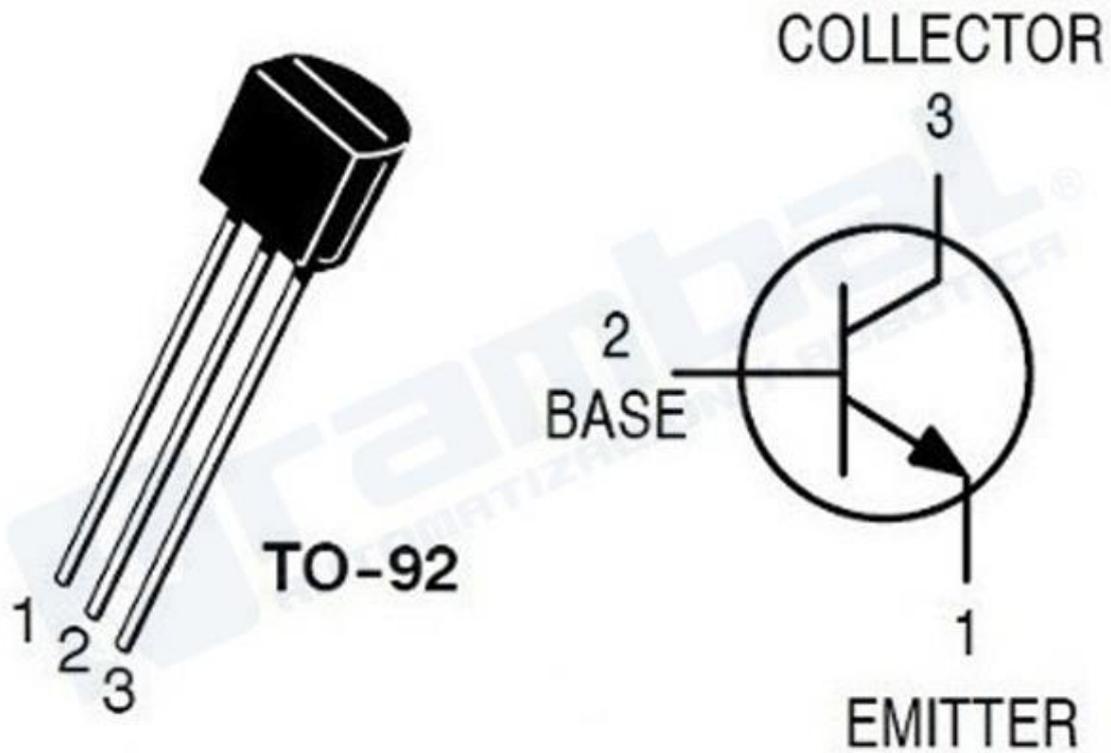
ESP32 Technical Specifications

Microprocessor	Tensilica Xtensa LX6
Maximum Operating Frequency	240MHz
Operating Voltage	3.3V
Analog Input Pins	12-bit, 18 Channel
DAC Pins	8-bit, 2 Channel
Digital I/O Pins	39 (of which 34 is normal GPIO pin)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
SRAM	520 KB
Communication	SPI(4), I2C(2), I2S(2), CAN, UART(3)
Wi-Fi	802.11 b/g/n
Bluetooth	V4.2 – Supports BLE and Classic Bluetooth

Pin Configuración

Pin Category	Pin Name	Details
Power	Micro-USB, 3.3V, 5V, GND	Micro-USB: ESP32 can be powered through USB port 5V: Regulated 5V can be supplied to this pin which is we be again regulated to 3.3V by on board regulator, to power the board. 3.3V: Regulated 3.3V can be supplied to this pin to power the board. GND: Ground pins.
Enable	En	The pin and the button resets the microcontroller.
Analog Pins	ADC1_0 to ADC1_5 and ADC2_0 to ADC2_9	Used to measure analog voltage in the range of 0-3.3V. 12-bit 18 Channel ADC
DAC pins	DAC1 and DAC2	Used for Digital to analog Conversion
Input/Output Pins	GPIO0 to GPIO39	Totally 39 GPIO pins, can be used as input or output pins. 0V (low) and 3.3V (high). But pins 34 to 39 can be used as input only
Capacitive Touch pins	T0 to T9	These 10 pins can be used a touch pins normally used for capacitive pads
RTC GPIO pins	RTCIO0 to RTCIO17	These 18 GPIO pins can be used to wake up the ESP32 from deep sleep mode.
Serial	Rx, Tx	Used to receive and transmit TTL serial data.
External Interrupts	All GPIO	Any GPIO can be use to trigger an interrupt.
PWM	All GPIO	16 independent channel is available for PWM any GPIO can be made to work as PWM though software
VSPI	GPIO23 (MOSI), GPIO19(MISO), GPIO18(CLK) and GPIO5 (CS)	Used for SPI-1 communication.
HSPI	GPIO13 (MOSI), GPIO12(MISO), GPIO14(CLK) and GPIO15 (CS)	Used for SPI-2 communication.
IIC	GPIO21(SDA), GPIO22(SCL)	Used for I2C communication.
AREF	AREF	To provide reference voltage for input voltage.

Apéndice C. Datasheet 2n2222

**ELECTRICAL CHARACTERISTICS** ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
OFF CHARACTERISTICS				
Collector-Emitter Breakdown Voltage ($I_C = 10 \text{ mA dc}$, $I_B = 0$)	$V_{(BR)CEO}$	40	-	Vdc
Collector-Base Breakdown Voltage ($I_C = 10 \mu\text{A dc}$, $I_E = 0$)	$V_{(BR)CBO}$	75	-	Vdc
Emitter-Base Breakdown Voltage ($I_E = 10 \mu\text{A dc}$, $I_C = 0$)	$V_{(BR)EBO}$	6.0	-	Vdc
Collector Cutoff Current ($V_{CE} = 60 \text{ Vdc}$, $V_{EB(off)} = 3.0 \text{ Vdc}$)	I_{CEX}	-	10	nA dc
Collector Cutoff Current ($V_{CB} = 60 \text{ Vdc}$, $I_E = 0$) ($V_{CB} = 60 \text{ Vdc}$, $I_E = 0$, $T_A = 150^\circ\text{C}$)	I_{CBO}	-	0.01 10	$\mu\text{A dc}$
Emitter Cutoff Current ($V_{EB} = 3.0 \text{ Vdc}$, $I_C = 0$)	I_{EBO}	-	10	nA dc
Collector Cutoff Current ($V_{CE} = 10 \text{ V}$)	I_{CEO}	-	10	nA dc
Base Cutoff Current ($V_{CE} = 60 \text{ Vdc}$, $V_{EB(off)} = 3.0 \text{ Vdc}$)	I_{BEX}	-	20	nA dc

ON CHARACTERISTICS

DC Current Gain ($I_C = 0.1 \text{ mA}$, $V_{CE} = 10 \text{ Vdc}$) ($I_C = 1.0 \text{ mA}$, $V_{CE} = 10 \text{ Vdc}$) ($I_C = 10 \text{ mA}$, $V_{CE} = 10 \text{ Vdc}$) ($I_C = 10 \text{ mA}$, $V_{CE} = 10 \text{ Vdc}$, $T_A = -55^\circ\text{C}$) ($I_C = 150 \text{ mA}$, $V_{CE} = 10 \text{ Vdc}$) (Note 1) ($I_C = 150 \text{ mA}$, $V_{CE} = 1.0 \text{ Vdc}$) (Note 1) ($I_C = 500 \text{ mA}$, $V_{CE} = 10 \text{ Vdc}$) (Note 1)	h_{FE}	35 50 75 35 100 50 40	- - - - 300 - -	-
Collector – Emitter Saturation Voltage (Note 1) ($I_C = 150 \text{ mA}$, $I_B = 15 \text{ mA}$) ($I_C = 500 \text{ mA}$, $I_B = 50 \text{ mA}$)	$V_{CE(sat)}$	- -	0.3 1.0	Vdc
Base – Emitter Saturation Voltage (Note 1) ($I_C = 150 \text{ mA}$, $I_B = 15 \text{ mA}$) ($I_C = 500 \text{ mA}$, $I_B = 50 \text{ mA}$)	$V_{BE(sat)}$	0.6 -	1.2 2.0	Vdc

SMALL – SIGNAL CHARACTERISTICS

Current – Gain – Bandwidth Product (Note 2) ($I_C = 20 \text{ mA}$, $V_{CE} = 20 \text{ Vdc}$, $f = 100 \text{ MHz}$)	f_T	300	-	MHz
Output Capacitance ($V_{CB} = 10 \text{ Vdc}$, $I_E = 0$, $f = 1.0 \text{ MHz}$)	C_{obo}	-	8.0	pF
Input Capacitance ($V_{EB} = 0.5 \text{ Vdc}$, $I_C = 0$, $f = 1.0 \text{ MHz}$)	C_{ibo}	-	25	pF
Input Impedance ($I_C = 1.0 \text{ mA}$, $V_{CE} = 10 \text{ Vdc}$, $f = 1.0 \text{ kHz}$) ($I_C = 10 \text{ mA}$, $V_{CE} = 10 \text{ Vdc}$, $f = 1.0 \text{ kHz}$)	h_{ie}	2.0 0.25	8.0 1.25	k Ω
Voltage Feedback Ratio ($I_C = 1.0 \text{ mA}$, $V_{CE} = 10 \text{ Vdc}$, $f = 1.0 \text{ kHz}$) ($I_C = 10 \text{ mA}$, $V_{CE} = 10 \text{ Vdc}$, $f = 1.0 \text{ kHz}$)	h_{re}	- -	8.0 4.0	$\times 10^{-4}$
Small–Signal Current Gain ($I_C = 1.0 \text{ mA}$, $V_{CE} = 10 \text{ Vdc}$, $f = 1.0 \text{ kHz}$) ($I_C = 10 \text{ mA}$, $V_{CE} = 10 \text{ Vdc}$, $f = 1.0 \text{ kHz}$)	h_{fe}	50 75	300 375	-
Output Admittance ($I_C = 1.0 \text{ mA}$, $V_{CE} = 10 \text{ Vdc}$, $f = 1.0 \text{ kHz}$) ($I_C = 10 \text{ mA}$, $V_{CE} = 10 \text{ Vdc}$, $f = 1.0 \text{ kHz}$)	h_{oe}	5.0 25	35 200	μMhos
Collector Base Time Constant ($I_E = 20 \text{ mA}$, $V_{CB} = 20 \text{ Vdc}$, $f = 31.8 \text{ MHz}$)	$rb'C_c$	-	150	ps
Noise Figure ($I_C = 100 \mu\text{A}$, $V_{CE} = 10 \text{ Vdc}$, $R_S = 1.0 \text{ k}\Omega$, $f = 1.0 \text{ kHz}$)	N_F	-	4.0	dB