



**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE GUAYAQUIL  
CARRERA DE INGENIERÍA ELECTRÓNICA**

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO IOT PARA EL  
MONITOREO DE PARÁMETROS AMBIENTALES APLICADOS AL  
CULTIVO DE ARROZ UTILIZANDO ESP32 Y THINGSPEAK.**

Trabajo de titulación previo a la obtención del  
Título de Ingeniero Electrónico

**AUTORES: LEANDRO MAURICIO MOSQUERA MELÉNDREZ  
CÉSAR DANILO CEVALLOS ROJAS  
TUTOR: ING. ORLANDO BARCIA AYALA MSC.**

Guayaquil – Ecuador

2022

## CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Leandro Mauricio Mosquera Meléndrez con cédula de identidad N° 0921719597 y César Danilo Cevallos Rojas con cédula de identidad N° 0931726400; manifestamos que:

Somos los autores y responsables de este trabajo de titulación; por lo cual autorizamos a la Universidad Politécnica Salesiana para que sin fines de lucro pueda utilizar, distribuir, reproducir o publicar total o parcialmente este proyecto.

Guayaquil, 5 de mayo del año 2022

Atentamente,



---

Leandro Mauricio Mosquera Meléndrez  
0921719597



---

César Danilo Cevallos Rojas  
0931726400

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN  
A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Leandro Mauricio Mosquera Meléndrez con cédula de identidad N° 0921719597 y César Danilo Cevallos Rojas con cédula de identidad N° 0931726400, por medio del presente documento en calidad de autores, manifestamos que se otorga a la Universidad Politécnica Salesiana los derechos patrimoniales del trabajo de investigación: “Diseño e implementación de un prototipo IoT para el monitoreo de parámetros ambientales aplicados al cultivo de arroz utilizando ESP32 y ThingSpeak”, el mismo que se ha elaborado para aspirar al título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, dejando claro que la Universidad está autorizada para ejercer en su totalidad los derechos traspasados.

En conformidad con lo anteriormente expuesto, se suscribe este documento en el instante en que se realiza la entrega del proyecto de titulación en formato digital a la Biblioteca que forma parte de la Universidad Politécnica Salesiana.

Guayaquil, 5 de mayo del año 2022

Atentamente,



Leandro Mauricio Mosquera Meléndrez  
0921719597



César Danilo Cevallos Rojas  
0931726400

## CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Orlando Barcia Ayala con documento de identificación N° 1309445714, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO IOT PARA EL MONITOREO DE PARÁMETROS AMBIENTALES APLICADOS AL CULTIVO DE ARROZ UTILIZANDO ESP32 Y THINGSPEAK, realizado por Leandro Mauricio Mosquera Meléndrez con documento de identificación N° 0921719597 y César Danilo Cevallos Rojas con documento de identificación N° 0931726400, obteniendo como resultado final el trabajo de titulación bajo la opción tesis que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 5 de mayo del año 2022

Atentamente,



Ing. Orlando Barcia Ayala, MSc.

1309445714

## **AGRADECIMIENTO**

Agradezco en primera instancia a mi Dios, por brindarme la oportunidad de culminar mis estudios universitarios, además de otorgarme salud, sabiduría y entendimiento para alcanzar esta tan anhelada meta.

A mis padres, por ser ejemplo de trabajo y sacrificio, porque siempre creyeron en mí y estuvieron cuando más los necesitaba, y un agradecimiento muy especial a mi querida y amada madre, ya que esto no hubiera sido posible sin su incondicional amor y apoyo.

A mi esposa Nathalia, por su apoyo, comprensión, paciencia, cariño y por creer en mis capacidades durante este largo y arduo camino.

A mi adorada y amada hija Romina, por ser mi impulso y fuente de motivación para la culminación de mi carrera profesional, y es por ella que me esfuerzo cada día para poder brindarle un futuro mejor.

A mis maestros, que con dedicación y empeño impartieron sus conocimientos, a mi tutor Ing. Orlando Barcia por guiarme con su experiencia para la culminación de este proyecto.

**LEANDRO MAURICIO MOSQUERA MELÉNDREZ**

## **AGRADECIMIENTO**

Primero doy las gracias a Dios por darme la vida, y por ser luz guía en mi camino y además de brindarme la sabiduría y fortaleza necesarias para alcanzar mis objetivos y metas. Por permitirme tener buenas experiencias tanto dentro como fuera de la universidad.

A mi madre Nelly por siempre brindarme su apoyo e impulsarme para seguir adelante hasta poder cumplir mis sueños y anhelos. Por ser un ejemplo de esfuerzo y trabajo duro que me motivo para poder alcanzar esta meta.

A mis maestros por guiarme a ser una mejor persona y profesional, mis especiales agradecimientos a mi tutor el Ing. Orlando Barcia por guiarme para poder culminar este proyecto.

**CÉSAR DANILO CEVALLOS ROJAS**

## RESUMEN DEL PROYECTO

<b>Año</b>	<b>Alumnos</b>	<b>Tutor de Proyecto de titulación</b>	<b>Proyecto de titulación</b>
2022	Leandro Mauricio Mosquera Meléndrez y César Danilo Cevallos Rojas	MSc. Orlando Barcia	Diseño e implementación de un prototipo IoT para el monitoreo de parámetros ambientales aplicados al cultivo de arroz utilizando ESP32 y ThingSpeak.

Este proyecto de titulación cuenta con un objetivo principal el cual es la implementación y el diseño de un prototipo IoT para monitorear los parámetros ambientales aplicados al cultivo de arroz utilizando ESP32 y ThingSpeak, el cual se aplica en un diseño de maqueta que servirá para realizar prácticas en el laboratorio de electrónica de la Universidad Politécnica Salesiana con sede en la ciudad de Guayaquil. Con la elaboración de este tipo de proyectos se da pie a que se desarrollen futuras investigaciones enfocadas en la agricultura, específicamente en el cultivo de arroz de la costa ecuatoriana, estas pueden ser llevadas a cabo por estudiantes de ingeniería, personal docente e investigadores de la facultad de ingeniería en electrónica, en la cual se puede realizar diversas aplicaciones de IoT.

Se realiza el desarrollo un prototipo IoT que consiste en varios sensores que tienen como función monitorear los factores ambientales como son la humedad y temperatura del suelo, la temperatura y humedad relativa, el nivel de agua y la radiación ultravioleta que se aplicarán para llevar un control del cultivo de arroz, estos parámetros son importantes para una correcta tecnificación en el desarrollo de las plantaciones de arroz y su posterior cultivo.

La propuesta del prototipo IoT se realiza mediante el dispositivo ESP32, que será el encargado de procesar toda la información recolectada por medio de los sensores y a su vez, serán subidos por medio de una red inalámbrica al internet, estos datos son guardados en servidores web de la plataforma ThingSpeak. Se ejecuta la evaluación y estudio de la información obtenida por los sensores, y mediante un sistema vía mails de ThingSpeak y tweets de Twitter alertar de acuerdo con los umbrales de cada sensor. Adicionalmente se realiza el control vía web mediante ThingHTTP, en el cual se puede realizar el encendido y

apagado automático de motor DC para el riego de agua del cultivo de arroz en maqueta de acuerdo con los umbrales establecidos de nivel de agua, estos umbrales serán definidos por expertos en el área del cultivo de arroz.

Para concluir, los estudiantes de ingeniería, personal docente y el departamento de investigadores de la facultad de ingeniería en electrónica de la Universidad Politécnica Salesiana con sede en la ciudad de Guayaquil, serán favorecidos con este proyecto de prototipo IoT, ya que podrán desarrollar prácticas en los laboratorios o en plantaciones de arroz, aplicando así, todo lo aprendido con lo que respecta al internet de las cosas en beneficio a la ingeniería.



## ABSTRACT

Year	Students	Degree Project Tutor	Technical Degree Project
2022	Leandro Mauricio Mosquera Meléndrez y César Danilo Cevallos Rojas	Ing. Orlando Barcia Ayala MSc.	Design and implementation of an IoT prototype for monitoring environmental parameters applied to rice cultivation using ESP32 and ThingSpeak.

This titling project has a main objective which is the implementation and design of an IoT prototype to monitor the environmental parameters applied to rice cultivation using ESP32 and ThingSpeak, which is applied in a mock-up design that will serve to carry out practices in the electronics laboratory of the Salesian Polytechnic University based in the city of Guayaquil. With the development of this type of projects, it is possible to develop future research focused on agriculture, specifically on the cultivation of rice on the Ecuadorian coast, these can be carried out by engineering students, teaching staff and researchers of the faculty of electronic engineering, in which various IoT applications can be carried out.

An IoT prototype is developed consisting of several sensors whose function is to monitor environmental parameters such as soil humidity and temperature, temperature and relative humidity, water level and ultraviolet radiation that will be applied to control of rice cultivation, these parameters are important for a correct technification in the development of rice plantations and their subsequent cultivation.

The proposal of the IoT prototype is carried out through the ESP32 device, which will be in charge of processing all the information collected through the sensors and, in turn, will be uploaded through a wireless network to the internet, these data are stored in web servers of the ThingSpeak platform. The evaluation and study of the information obtained by the sensors is carried out, and through a system via ThingSpeak emails and Twitter tweets, alerts are made according to the thresholds of each sensor. Additionally, control is carried out via the web through ThingHTTP, in which the automatic turning on and off of the DC motor can be carried out for the water irrigation of the rice crop in a model according to the established water level thresholds, these thresholds will be defined by experts in the area of rice cultivation.

To conclude, engineering students, teaching staff and the research department of the electronic engineering faculty of the Salesian Polytechnic University based in the city of Guayaquil, will be favored with this IoT prototype project, since they will be able to develop practices in laboratories or in rice plantations, thus applying everything learned with regard to the internet of things for the benefit of engineering.

## ÍNDICE GENERAL

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN .....	II
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN .....	IV
AGRADECIMIENTO.....	V
AGRADECIMIENTO.....	VI
RESUMEN DEL PROYECTO.....	VII
ABSTRACT .....	IX
ÍNDICE GENERAL.....	XI
ÍNDICE DE FIGURAS .....	XIV
INTRODUCCIÓN .....	XVII
1. El problema .....	1
1.1. Descripción del problema .....	1
1.2. Antecedentes .....	1
1.3. Importancia y alcance .....	2
1.4. Delimitación.....	2
1.4.1. Delimitación temporal.....	2
1.4.2. Delimitación espacial .....	2
1.4.3. Delimitación académica .....	3
1.5. Beneficiarios de la propuesta .....	3
1.6. Propuesta de solución .....	3
1.7. Innovación e impacto del proyecto .....	4
1.8. Objetivos .....	5
1.8.1. Objetivo general .....	5
1.8.2. Objetivos específicos .....	5
2. Fundamentos teóricos.....	6
2.1. Características principales del arroz .....	6
2.2. El cultivo de arroz en el Ecuador.....	6
2.3. Parámetros ambientales adecuados para el cultivo de arroz.....	7
2.4. Tecnificación del cultivo del arroz.....	8
2.5. IoT .....	9
2.6. IoT en la agricultura .....	10
2.7. Cultivos aplicando la tecnología IOT.....	11
2.8. IoT aplicado al cultivo del arroz.....	11

2.9.	Smart Farming ThingSpeak.....	12
2.10.	Dispositivos para prototipos IoT .....	13
2.10.1.	Módulo Wifi, Bluetooth y Microcontrolador ESP32.....	13
2.10.2.	Sensor de Humedad Relativa, Ambiente y Temperatura DTH22.....	15
2.10.3.	Sensor de Humedad de la tierra y Temperatura SHT-10.....	15
2.10.4.	Sensor Ultrasonido US-100.....	17
2.10.5.	Sensor de luz ultravioleta UV UVM-30A.....	18
2.10.6.	Convertidor Voltaje DC-DC Step-Down 2A MP1584EN.....	19
2.11.	ThingSpeak.....	20
3.	Marco metodológico.....	21
3.1.	Tipo de investigación .....	21
3.2.	Diseño de investigación.....	21
3.3.	Enfoque de la investigación.....	21
3.4.	Metodología de investigación .....	22
3.5.	Proyectos de investigación vinculados.....	23
3.5.	Título de la propuesta .....	23
3.6.	Descripción de la propuesta .....	23
4.	Resultados.....	25
4.1.	Maqueta .....	25
4.2.	Diseño electrónico del prototipo .....	27
4.3.	Programación del prototipo.....	32
4.4.	Montaje y pruebas de prototipo .....	38
4.5.	Configuraciones en ThignSpeak .....	46
5.	Análisis de resultados .....	57
5.1.	Mails de alertas recibidos .....	60
5.2.	Tweets de alertas recibidos .....	61
5.3.	Resultados de encendido de la bomba.....	63
5.4.	Resultados de los datos obtenidos por los sensores.....	65
5.4.1	Análisis de datos del Sensor Temperatura Ambiente.....	65
5.4.2	Análisis de datos del Sensor Humedad Relativa.....	66
5.4.3	Análisis de datos del Sensor Humedad del Suelo.....	67
5.4.4	Análisis de datos del Sensor Temperatura del Suelo.....	68
5.4.5	Análisis de datos del Sensor del Nivel del Agua.....	69
5.4.6	Análisis de datos del Sensor del Índice UV.....	70

5.4.7	Análisis de los Estados de la Bomba.....	71
5.5.	Visualización de resultados por medio de aplicación móvil ThingView.....	72
6.	Conclusiones.....	73
7.	Recomendaciones.....	74
	Bibliografía.....	75
	Anexos.....	77

## ÍNDICE DE FIGURAS

Figura 1 Cosecha y Post cosecha del arroz [5].....	7
Figura 2 Tecnificación arroceras [2] .....	9
Figura 3 IoT Aplicado al cultivo del arroz [4].....	12
Figura 4 Cultivo de arroz tecnificado [7] .....	13
Figura 5 ESP32-WROOM-32D [12].....	14
Figura 6 WROOM-32U [13].....	14
Figura 7 Sensor DHT22 [14].....	15
Figura 8 Sensor SHT-10 [14] .....	16
Figura 9 Sensor US-100 [14] .....	17
Figura 10 Sensor UV UVM-30A [14].....	18
Figura 11 Convertidor de voltaje MP-1584EN [14].....	19
Figura 12 Plataforma ThingSpeak [15].....	20
Figura 13 Esquema del proceso.....	22
Figura 14 Prototipo IoT para cultivo de arroz .....	24
Figura 15 Maqueta .....	26
Figura 16 Sistema de sensores .....	26
Figura 17 Conexiones del ESP32 placa principal .....	27
Figura 18 Conexión USB UART Bridge.....	28
Figura 19 Diseño del power supply .....	28
Figura 20 Conexiones de los sensores .....	29
Figura 21 Conexiones GPIO del ESP32.....	29
Figura 22 CP2102 .....	30
Figura 23 Diseño del MP1584 .....	30
Figura 24 UVM-30A.....	31
Figura 25 Conexiones del ESP32 placa secundaria .....	31
Figura 26 Placa de prototipo en 3D .....	32
Figura 27 Código para conectarse a la red WiFi.....	33
Figura 28 Código que comunica los módulos ESP32 con ThingSpeak. ....	33
Figura 29 Código para sensor de temperatura ambiente y humedad relativa. ....	34
Figura 30 Código para sensor de temperatura y humedad del suelo.....	34
Figura 31 Código para sensor ultrasónico US100. ....	35
Figura 32 Código para sensor ultrasónico US100. ....	35

Figura 33 Código para subir los datos a la plataforma ThingSpeak.....	36
Figura 34 Código placa secundaria.....	37
Figura 35 Montaje de estructura de aluminio.....	38
Figura 36 Montaje de sensores.....	38
Figura 37 Montaje del prototipo en maqueta de cultivo.....	39
Figura 38 Alineación de prototipo IoT.....	39
Figura 39 Prototipo IoT, sensor UV.....	40
Figura 40 Bomba de agua.....	40
Figura 41 Sensor de proximidad.....	41
Figura 42 Sensor de humedad y temperatura de suelo.....	41
Figura 43 Sensor de temperatura y ambiente.....	42
Figura 44 Sensores desmontados.....	42
Figura 45 Preparado de tierra de cultivo.....	43
Figura 46 Colocación de tierra en maqueta de cultivo.....	43
Figura 47 Maqueta con cultivo de arroz.....	44
Figura 48 Prototipo en cultivo de arroz.....	44
Figura 49 Testeo de prototipo en maqueta.....	45
Figura 50 Puesta en marcha de prototipo en maqueta.....	45
Figura 51 ThingSpeak.....	46
Figura 52 Cuenta de ThingSpeak.....	47
Figura 53 Canal de ThingSpeak.....	47
Figura 54 Configuraciones de canal.....	48
Figura 55 Configuraciones de Fields.....	49
Figura 56 Configuraciones de localidad.....	50
Figura 57 API Keys.....	51
Figura 58 Sección de importar y exportar.....	52
Figura 59 Configuraciones de APP.....	53
Figura 60 APP para envío de mails.....	54
Figura 61 Configuración de React.....	55
Figura 62 Configuración de acción.....	56
Figura 63 Configuración de acción y react.....	56
Figura 64 Dashboards de sensores.....	57
Figura 65 Dashboard de ubicación.....	58
Figura 66 Datos de nivel de agua y UV.....	58

Figura 67 Datos de temperatura ambiente y humedad relativa .....	59
Figura 68 Datos de Humedad y Temperatura del suelo .....	59
Figura 69 Mails recibidos de ThingSpeak .....	60
Figura 70 Alerta recibida por mail .....	61
Figura 71 Cuenta de Twitter del proyecto .....	61
Figura 72 Configuraciones de Twitter en ThingSpeak .....	62
Figura 73 Configuraciones de Tweet alert React .....	62
Figura 74 Configuraciones de React para apagado de bomba .....	63
Figura 75 Configuraciones de React para encendido de bomba.....	63
Figura 76 Configuraciones de ThingHTTP para apagado de bomba .....	64
Figura 77 Configuraciones de ThingHTTP para encendido de bomba .....	64
Figura 78 Datos de temperatura ambiente .....	65
Figura 79 Datos de Humedad relativa .....	66
Figura 80 Datos de Humedad de suelo .....	67
Figura 81 Datos de temperatura de suelo .....	68
Figura 82 Datos de nivel de agua .....	69
Figura 83 Datos de índice UV .....	70
Figura 84 Dashboards de encendido de bomba .....	71
Figura 85 Aplicación móvil ThingView .....	72



## INTRODUCCIÓN

*Oryza sativa*, nombre científico del arroz, es un cereal y alimento importante dentro de la dieta que consumen los ecuatorianos, por tal motivo cada año se consume bastantes cantidades de arroz en el país, este producto se da específicamente en la zona costera del Ecuador, el clima adecuado para el cultivo se desarrolla en climas tropicales y subtropicales, con precipitaciones entre 0.200 a 0.300 metros de lluvia por cada mes, este grano debe estar a una temperatura óptima para el desarrollo del cultivo entre 24 grados Celsius y 35 grados Celsius, humedad relativa mayor al 80%. Luminosidad eficiente no menos de 12 horas de luz natural, pH del suelo óptimo de 6.6 ya que con esta estimación la liberación microbiana de elementos como el nitrógeno y el fósforo de la materia orgánica y su disponibilidad es muy elevada. Estos parámetros de carácter ambiental para el crecimiento de la planta de arroz en los cultivos, son de mucha importancia para el agricultor ya que depende de un correcto monitoreo de estas variables para que su producción de arroz sea de la manera óptima [1] [2].

Hay arroceros en el Ecuador que utilizan herramientas tecnológicas con eficiencia económica y buena producción. Sin embargo, según el Ministerio de Agricultura, hay gente que gasta menos en sembrar para no perder ingresos. La baja producción de arroz de 2 a 3 toneladas/ha (t/ha) es una de las razones por las que algunos agricultores abandonan sus negocios [3].

Con la tecnificación de las hectáreas de arroz de los productores de algunos sectores costeros del país se ha logrado aumentar sus ganancias, sin embargo, se estima que solo el 20% de agricultores tecnifican sus cultivos, por los altos costos generados. Con el desarrollo de sistemas basados en IoT que utilizan sistemas de sensores para monitorear los parámetros ambientales utilizados en la agricultura y el cultivo de arroz, será posible realizar una tecnificación adecuada y además aumentar los rendimientos, minimizando los daños causados por plagas, sequías e inundaciones de las plantaciones de arroz, etc. [4].

Para ese entorno, se plantea utilizar un prototipo IoT para monitorear parámetros ambientales empleados al cultivo de arroz como propósito de prácticas y pruebas para los próximos ingenieros electrónicos en la carrera de ingeniería electrónica de la Universidad Politécnica Salesianas de la ciudad de Guayaquil, con el fin de implementar métodos para aplicaciones IoT. Así como también se apertura la opción de la aplicación del prototipo IoT en un cultivo real siempre y cuando haya cobertura de red para la transmisión de datos.

El proyecto de investigación está ordenado con la siguiente forma, donde el primer apartado describe el problema, sus antecedentes, la importancia y el alcance, su delimitación, los beneficiarios de la propuesta, su propuesta de solución y los objetivos. En la segunda sección se dan a conocer los argumentos de la teoría del cultivo del arroz. El tercer apartado relata el marco metodológico, y los tipos de investigación. El cuarto título explica sobre los frutos de la investigación. En el último apartado, que viene a ser el quinto, se describe el estudio de los resultados obtenidos, las conclusiones y recomendaciones para el buen uso del prototipo IoT.

## **1. El problema**

### **1.1. Descripción del problema**

Uno de los problemas que principalmente afectan al cultivo de arroz en agricultores de la costa ecuatoriana es la poca tecnificación del cultivo de la planta de arroz. El control adecuado de los parámetros ambientales como la radiación ultravioleta, la temperatura y humedad del ambiente, y los niveles de temperatura y humedad del suelo durante el crecimiento de la plantación de arroz es importante para el agricultor de las zonas costeras del Ecuador, ya que le permitirá tomar decisiones oportunas para evitar plagas, enfermedades, e incluso la pérdida total del cultivo arroz.

La poca accesibilidad de plataformas o prototipos IoT aplicados a la agricultura dentro de la Universidad Politécnica Salesiana de la ciudad de Guayaquil es uno de los problemas en el desarrollo técnico práctico del estudiante de la facultad de ingeniería electrónica.

### **1.2. Antecedentes**

La necesidad de tecnificar el cultivo del arroz en los pequeños productores agrícolas para una mejor producción y mejores rendimientos económicos incentiva a la ejecución de este sistema IoT de bajo costo, el cual mediante sensores y electrónica se pueden monitorizar los parámetros ambientales de forma idónea para las plantaciones de arroz.

Mediante la elaboración de este trabajo de investigación, no solo se va a lograr el objetivo de modernizar la agricultura en las zonas rurales de Ecuador, mejorando las condiciones técnicas de los pequeños productores de arroz, sino también ser fundamental en la formación de ingenieros electrónicos y de telecomunicaciones en el futuro de la Universidad Politécnica Salesiana, con sede en Guayaquil.

### **1.3. Importancia y alcance**

Con el aporte de este proyecto, el gremio de agricultores que producen arroz espera beneficiarse de los estudios y prototipos agrícolas diseñados por parte de estudiantes e investigadores de la Universidad Politécnica Salesiana de Guayaquil, así mismo, los estudiantes de la carrera de ingeniería en electrónica y telecomunicaciones podrán realizar prácticas y emplear los conocimientos adquiridos en temas relacionados con IoT.

Con lo mencionado anteriormente, se expone este proyecto de titulación para acceder a la aplicación de nuevas técnicas para el cultivo del arroz por medio de aplicaciones del IoT, dispositivos tecnológicos y además el uso de servidores web en el internet para el monitorear los diferentes factores ambientales como son la temperatura y humedad del ambiente, temperatura y humedad del suelo, nivel de agua y los rayos ultravioletas.

### **1.4. Delimitación**

#### **1.4.1. Delimitación temporal**

Para la realización del diseño, estructuración y ensayos de funcionamiento del proyecto de tesis es de aproximadamente 8 meses, conforme al cronograma estipulado desde el mes de junio del año 2021 hasta el mes de febrero del año 2022.

#### **1.4.2. Delimitación espacial**

Por motivos de la amenaza a la salud pública relacionada con la pandemia a nivel mundial del COVID-19 y de acuerdo con las directrices establecidas por la Universidad Politécnica Salesiana de la ciudad de Guayaquil, se realizarán los ensayos del sistema IoT de monitoreo de factores ambientales para el cultivo de arroz en la casa del tesista Leandro Mosquera, responsable del trabajo de titulación, ubicada en la ciudad de Guayaquil, con el uso de una maqueta elaborada a escala, la cual será llevada al Laboratorio de Sensores y Actuadores de la Universidad Politécnica Salesiana con sede en Guayaquil para beneficio de los estudiantes de la facultad de electrónica y automatización.

Las pruebas e interpretaciones de los resultados obtenidos se realizarán en directo de acuerdo con el diseño empleado en el prototipo IoT, los datos obtenidos por los sensores implementados en el prototipo se gestionarán a través de internet en el servidor de la plataforma de ThingSpeak y se controlarán por medio de acceso remoto vía web.

#### **1.4.3. Delimitación académica**

El proyecto de titulación el cual es un prototipo IoT práctico y útil será un instrumento clave para el crecimiento de las aptitudes y habilidades de los próximos ingenieros electrónicos de la Universidad Politécnica Salesiana de la ciudad de Guayaquil. Gracias a este proyecto se lleva a cabo los conocimientos y destrezas que fueron obtenidos en las asignaturas tales como redes inalámbricas, comunicaciones digitales, redes de computadora, electrónica digital, etc., que ofrece la carrera de ingeniería en electrónica y telecomunicaciones.

#### **1.5. Beneficiarios de la propuesta**

Los beneficiarios directos de este trabajo de investigación serán los pequeños productores de arroz en las zonas rurales de la costa de Ecuador, lo que dará resultados positivos para la aplicación de medidas de control de arroz más eficientes y fáciles después de implementar el sistema IOT.

Además, los estudiantes de la facultad de ingeniería en electrónica y telecomunicaciones de la Universidad Politécnica Salesiana de Guayaquil serán favorecidos ya que se entregará información e instrucciones prácticas sobre el prototipo IoT y se proporcionará una maqueta con el sistema de sensores para la realización de las prácticas en laboratorio o en el campo.

#### **1.6. Propuesta de solución**

Se presenta un prototipo IoT de monitoreo de factores ambientales como temperatura y humedad relativa, temperatura y humedad del suelo, nivel de agua y radiación UV en el cultivo

de arroz, parámetros importantes para una correcta tecnificación en el crecimiento de las plantas de arroz.

La propuesta del prototipo IoT se realiza mediante el dispositivo ESP32, que será el encargado de procesar toda la información recolectada por medio de los sensores y a su vez, serán subidos por medio de una red inalámbrica al internet, estos datos son guardados en servidores web de la plataforma ThingSpeak.

Se observa también que el monitoreo puede realizarse vía web y vía APP mediante un celular, además hay configuraciones de alerta vía web desde la plataforma de ThingSpeak el cual enviará emails de alerta si algún parámetro monitoreado sobrepasa o disminuye el umbral de monitoreo establecido.

La data almacenada y tratada por ThingSpeak puede ser descargada y en lo posterior analizada para una toma de decisiones futuras en el cultivo del arroz.

Los valores adquiridos por el sistema de sensores permitirán la automatización de los actuadores que van a encender un motor DC, para el llenado de agua en las plantaciones de arroz.

Para el diseño de este proyecto se realizará un esquema de tarjeta PCB cuya función será el control y administración del sistema de sensores que estarán en la maqueta. La maqueta que se va a utilizar será a escala en tamaño de 2x2 metros para la simulación de un terreno de cultivo de arroz.

### **1.7. Innovación e impacto del proyecto**

El diseño y elaboración de este trabajo de titulación posee un notable grado de innovación, ya que la configuración, codificación y recolección de datos se realiza con plataformas IoT diseñadas y adecuadas para la monitorización de parámetros ambientales en el cultivo del arroz, además, el impacto del proyecto es excelente, permitiendo consolidar los conocimientos básicos del prototipo para los próximos análisis e investigaciones en el área del IoT para el monitoreo de los factores ambientales en cultivos de arroz.

## **1.8. Objetivos**

### **1.8.1. Objetivo general**

Diseñar e implementar un prototipo IoT para el monitoreo de parámetros ambientales aplicados al cultivo de arroz utilizando ESP32 y ThingSpeak.

### **1.8.2. Objetivos específicos**

- Diseñar un sistema de red de los sensores de temperatura ambiente, humedad relativa, humedad de suelo, temperatura del suelo, nivel de agua, y radiación UV para dispositivo IoT de monitoreo de parámetros ambientales para el cultivo de arroz.
- Implementar servicios en la nube para el almacenamiento y monitoreo web de la información obtenida por los sensores mediante ThingSpeak, además monitorear mediante una APP los datos obtenidos de los cultivos de arroz.
- Evaluar y analizar los diferentes datos obtenidos de los sensores y mediante un sistema vía mails de ThingSpeak y tweets de Twitter alertar de acuerdo con los umbrales de cada sensor.
- Controlar vía web mediante ThingHTTP el encendido y apagado automático de motor DC para el riego de agua del cultivo de arroz en maqueta de acuerdo con los umbrales establecidos de nivel de agua, estos umbrales serán definidos por expertos en el área del cultivo de arroz.

## **2. Fundamentos teóricos**

### **2.1. Características principales del arroz**

Al igual que los cereales, el arroz es rico en carbohidratos y es una importante fuente de energía para el cuerpo humano. Así que el arroz integral proporcionará energía de forma pausada y gradual porque tiene un mayor contenido de fibra que el arroz blanco y ralentiza la asimilación de nutrientes. Por otro lado, este grano también contiene vitaminas del grupo B como son: la tiamina (B1), el niacina (B3), la piridoxina (B6) y el ácido fólico (B9) donde este último es muy importante para las mujeres embarazadas. Otro factor importante es que también contiene minerales esenciales como son el fósforo y el potasio.

Esta gramínea se cultiva en las zonas de la costa ecuatoriana, sobre todo en la provincia del Guayas y como en la provincia de Los Ríos. Los factores climáticos de las áreas productoras de arroz en todo el país tienen un amplio rango de distribución, desde tropical húmedo a tropical seco, temperatura aproximada de 20 °C ~ 35 °C, precipitación anual máxima de 2500 mm, y una mínima de 500 mm, elevada humedad y un índice UV relativamente alto [3].

### **2.2. El cultivo de arroz en el Ecuador**

Se cosecha cuando las espigas alcanzan el 95% madurez, es decir, cuando la mayor parte de la espiga es de color pajizo, el resto es amarillo como se observa en la Figura 1.

Un indicador importante es su cantidad de agua, por lo general es el 25% aproximadamente, en el período de cosecha muy temprano, cuando todavía hay muchos granos inmaduros, el rendimiento es muy bajo, pero si la cosecha es tardía, disminuye el rendimiento debido a pérdidas cereales para alojamiento, descascarado, roedores y aves. En cualquiera de los dos escenarios es perjudicial.





**Figura 1** Cosecha y Post cosecha del arroz [5]

Después de la cosecha, es importante secar el arroz hasta que la humedad máxima se reduzca aproximadamente al 12%. Una vez que los granos de arroz se secan hasta alcanzar el valor estimado de 12% de humedad, se puede almacenar sin riesgo al ataque de enfermedades. [6].

### **2.3. Parámetros ambientales adecuados para el cultivo de arroz**

Los factores ambientales adecuados en los que resulta óptimo el desarrollo del cultivo de arroz son:

- a) **La temperatura:** El arroz requiere al menos de 10°C a 13°C para germinar, considerándose óptima una temperatura entre 30°C y 35°C. A temperaturas superiores a 40°C no se produce la germinación. El crecimiento mínimo del tallo, hojas y raíces es de 7°C, donde 23°C se considera óptimo. A temperaturas más altas, las plantas crecen más rápido, pero los tejidos se vuelven demasiado blandos siendo muy susceptibles a enfermedades.
  
- b) **La radiación solar:** Los requisitos de radiación solar para el cultivo de arroz dependen de las diferentes etapas de crecimiento de la planta. La baja irradiación solar durante la fase vegetativa tiene poco efecto sobre su productividad y composición, mientras que el número de semillas se reduce significativamente durante la fase vegetativa. En

cambio, en el período que va desde el llenado hasta la maduración de las semillas, el rendimiento disminuye drásticamente debido a una disminución en la tasa de llenado de las semillas.

- c) **La altitud:** El arroz se puede cultivar en lugares que se encuentran en el nivel del mar hasta zonas que están a aproximadamente hasta los 2500 metros de altura.
- d) **Las precipitaciones:** Determinan los sistemas y métodos de cultivo, especialmente cuando se cultivan en zonas montañosas, donde se ven más afectados por su variabilidad.
- e) **El suelo:** Las condiciones edáficas a las cuales el cultivo de arroz se adapta son las siguientes:
  - a) **La textura del suelo:** Pueden ser arenosas o compuestas de arcilla, donde poseen una fina textura con un drenaje óptimo.
  - b) **El pH:** El valor adecuado es de 6.6.

#### **2.4. Tecnificación del cultivo del arroz**

En el cantón Daule, específicamente en Santa Lucía, se utilizan siete drones para fumigar las plantaciones de arroz. Los agricultores cercanos se han acostumbrado al sonido de los aparatos eléctricos. La máquina ocupa una hectárea en siete minutos y puede funcionar durante dos veces al día. Puede llevar de cuatro a cinco horas rociar manualmente una hectárea de arroz, tal como se muestra en la Figura 2.



**Figura 2** *Tecnificación arrocera [2]*

El trabajo manual se observa en la mayoría de las plantaciones en este cantón, y el estado tiene 40,000 residentes que dependen del cultivo de esta gramínea. Con 18.000 hectáreas plantadas, es la segunda área de siembra del Guayas, después de Daule, y es considerada la "capital arrocera del Ecuador". Sin embargo, se estima que solo el 5% de su producción es tecnológica, lo que es similar a lo que ocurre a nivel nacional.

Hoy se ha tecnificado toda la cosecha, desde hace dos años se promueve el uso de tecnología para reducir el costo de 130 días del ciclo productivo, mejorar la calidad de los alimentos y competir con otros países [7].

## **2.5. IoT**

El IoT, también conocido como Internet de las cosas son las conexiones de tecnologías integradas que contienen dispositivos físicos y se utilizan para la comunicación e interacción en áreas internas y externas. En lugar de la comunicación de persona a persona, IoT hace hincapié en la comunicación de máquina a máquina.

Hoy en día, en todo el mundo, la tecnología IoT se encuentra entre las 5 principales tecnologías según el gráfico de Gartner. Eso significa que es muy utilizado en diferentes sectores en diferentes roles, ya sea en hogares inteligentes o seguimiento de vehículos,

monitoreo de niños y personas mayores o trabajo de rutina diaria. Sin embargo, en la actualidad la realidad es que estos segmentos contratan varios dispositivos habilitadores de IoT, y el futuro ya se está fragmentando con la nueva revolución.

Finalmente, el futuro de IoT se convierte en un valor, pero cantidades masivas de datos incrementaron su complejidad en detección, comunicaciones, controlador y en generar conciencia, pero su crecimiento aumentará día a día. Aunque el futuro de IoT será predecible para integrarse, todos en uno, y ubicuo. La organización de servicios debe estar incluida en un conjunto de normas. Por lo tanto, como sistema inteligente, los avances de IoT se pueden decidir con la cooperación de interoperabilidad, conciencia, habilidad, trabajo en equipo, sostenibilidad energética, privacidad, confianza, confidencialidad y seguridad [8].

En los próximos años, mejorar estos desafíos será un paso poderoso y audaz para las redes y la comunicación en el área comercial, industrial y académica.

## **2.6. IoT en la agricultura**

Emplear IoT en la agricultura se describe como una tecnología que tiene como objetivo organizar varios sensores en una red a través de la cual se puede recopilar información de la tierra apta para la agricultura y los resultados se pueden analizar y transmitir a los agricultores en tiempo real para que decidan los cambios o mejoras para el cultivo. Son variados los desafíos de implementar IoT en la agricultura los cuales se tiene: bajos costos operativos, escalabilidad, necesidad de soluciones de bajo costo, desarrollo de tierras pequeñas y desiguales, facilidad de uso y permisividad al error.

A nivel técnico, el uso de IoT en la agricultura ha encontrado poderosos aliados para el procesamiento de datos en las siguientes aplicaciones a través de la computación en la nube: uso efectivo de fertilizantes y de algunos pesticidas, disminución de precios, control eficaz de la ganadería, agricultura de interior, invernaderos, piscicultura, supervisión de almacenamiento en depósitos de agua, reservorios de combustible, asignación de recursos, actualización en el backend, desarrollo simple y rápido, también se incluye la colaboración con ciertos sistemas en la web. Ciertas ventajas de utilizar Internet de las cosas y la red WSN en la agricultura son: alta escalabilidad, bajo costo, larga vida, sistema reconfigurable, margen

de error, seguridad de la información, administración de terrenos agrícolas, calidad de servicio (QoS) y administración dinámica [9] [10].

## **2.7. Cultivos aplicando la tecnología IOT**

IoT en la agricultura tiene como principio determinar el grado de empleo de la tecnología en la agricultura; por otro lado, el enfoque es buscar el desarrollo de aplicaciones en América Latina para determinar el predominio de este tipo de tecnología en diversas zonas de la región.

Al elaborar un sistema que puede recopilar datos de factores relacionados con los cultivos en su crecimiento y en su desarrollo, se utilizan para observar el sistema de IoT para cultivos vigilados. Los datos obtenidos se envían al servidor para su procesamiento y a su vez se envían al usuario final a través del protocolo determinado.

El objetivo es obtener información en tiempo real para su estudio y a su vez autorizar al propio sistema, así como a los agricultores, la toma de resoluciones. El usuario final puede comunicarse de forma remota con la plataforma además de recibir alertas y ordenes específicas. Los primeros resultados muestran que el sistema brinda información extensa acerca el estado de estos factores, lo cual es primordialmente útil para el cuidado de estos cultivos.

## **2.8. IoT aplicado al cultivo del arroz**

Debido a la comunicación a través de Internet, los beneficios de diferentes tecnologías como los sistemas GPS y los sensores para los agricultores se han multiplicado. Por un lado, esta conectividad permite que las tecnologías involucradas trabajen junto con el backend de información para así mejorar la toma de las decisiones en vivo o en tiempo real. Además, les da la opción hablar entre ellos y trabajar como un único sistema integrado, dándose retroalimentación entre ellos. En la Figura 3, se puede apreciar un modelo de la comunicación en un cultivo.



**Figura 3** IoT Aplicado al cultivo del arroz [4]

Del mismo modo, gracias a la entrada de IoT (Internet of Things), la tecnología de agricultura de precisión se ha vuelto más inteligente. Por lo tanto, no es necesaria la intervención manual en el proceso agrícola todos los días. Las máquinas y tecnologías involucradas son más autónomas que nunca, capaces de tomar decisiones en unos pocos segundos, y la precisión es inimaginable hasta ahora.

## **2.9. Smart Farming ThingSpeak**

Smart Farming es una gestión agrícola que utiliza tecnologías de datos para aumentar la calidad y eficacia de la producción agrícola. ThingSpeak proporciona gestión y almacenamiento de datos para sus sensores y controles agrícolas. Luego, puede usarse MATLAB en ThingSpeak para crear los algoritmos que hacen que la granja sea más inteligente.

Ver de manera instantánea la información publicada por sus dispositivos, como datos de humedad y temperatura del suelo en vivo o información sobre la salud de los cultivos, como indica la Figura 4.

Realizar análisis y procesamiento de datos en línea con datos en vivo y ver visualizaciones automáticas. Crear sistemas de Internet de las cosas (IoT) en su ubicación sin configurar servidores ni desarrollar software web [11].



**Figura 4** Cultivo de arroz tecnificado [7]

## **2.10. Dispositivos para prototipos IoT**

### **2.10.1. Módulo Wifi, Bluetooth y Microcontrolador ESP32.**

Se trata de un dispositivo WiFi / Bluetooth completo, integrado y certificado que va mucho más allá de proporcionar radio. Su procesador en realidad posee dos núcleos de procesamiento, y la frecuencia de operación se controla de manera independiente en un rango de 80 megahercios (MHz) y los 240 MHz. Los elementos periféricos del procesador se pueden conectar fácilmente a varias interfaces del exterior, entre esas se tiene:

- Interfaz Periférica Serial
- I2C
- Transmisor receptor asíncrono universal (UART)
- I2S
- Ethernet
- tarjeta SD

Los desarrolladores pueden elegir una variedad de módulos ESP32 diferentes de acuerdo con los requisitos de la aplicación. El primer módulo es ESP32-WROOM-32D, este es el más popular, que posee una frecuencia de operación de hasta 240 MHz como se muestra en la Figura 5.



Figura 5 ESP32-WROOM-32D [12]

El dispositivo tiene una antena de rastreo de PCB para simplificar la implementación, como se muestra en la Figura 6.

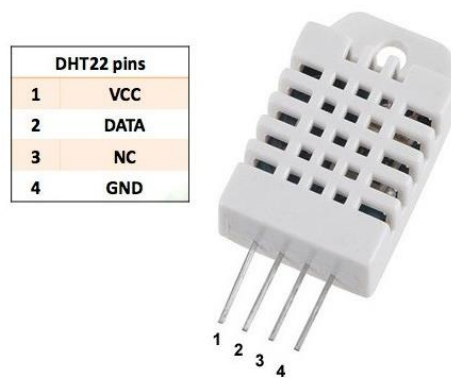


Figura 6 WROOM-32U [13]



### 2.10.2. Sensor de Humedad Relativa, Ambiente y Temperatura DHT22.

DHT22 (AM2302) es un sensor digital de temperatura y humedad relativa de alto rendimiento y de un costo bajo. Posee un sensor de humedad capacitivo y además un termistor para medir el aire que circula alrededor, muestra los datos a través de una señal digital en el pin de datos. Es utilizado en el control automático de temperatura, en los aires acondicionados y para el control del entorno agrícola, como indica la Figura 7.



**Figura 7** Sensor DHT22 [14]

### 2.10.3. Sensor de Humedad de la tierra y Temperatura SHT-10.

El SHT10 es un sensor fuerte y resistente a la oxidación, este puede ser usado tanto en la industria como a nivel empresarial, o bien en diversos proyectos: estaciones de control, cajas de monitoreo electrónico, en bodegas, cuartos de maquinaria, en viveros, granjas de animales, en el sector de la medicina, etc.

Está elaborado a base de metal, como se aprecia en la Figura 8, esto le permite trabajar en lugares de mucho polvo, puede ser colgado de una pared lo que lo hace muy fácil de instalar en cualquier lugar.

La malla metálica que cubre el sensor está impregnada con cobre, haciéndolo más resistente a elevadas temperaturas, presión y desgaste mecánico. Su regulación de voltaje interna, y la interfaz serial de dos hilos permiten una integración sencilla para la elaboración de proyectos.



**Figura 8** Sensor SHT-10 [14]

Este tipo de sensores tiene las siguientes especificaciones:

- El consumo eléctrico: 0,15mW (Promedio).
- El tipo de interfaz: interfaz two wire (Digital).
- Los rangos de medición 0-100% RH.
- Los rangos de temperatura -10°C – 80°C.
- La precisión de humedad:  $\pm 5.0\%$  RH.
- La precisión de Temperatura:  $\pm 0.5^\circ\text{C}$ .
- El tamaño: 49mm x 14mm.

#### 2.10.4. Sensor Ultrasonido US-100.

Es un sensor de distancia económico que utiliza ultrasonido para medir la distancia de un objeto en un rango de longitud de 2 a 450 cm. Sobresale por ser un dispositivo de tamaño pequeño, consume poca energía y es preciso. Se lo utiliza en proyectos de robótica, y en diversos tipos de proyectos de automatización. Posee dos transductores: un emisor y un receptor, como se muestra en la Figura 9.



**Figura 9** Sensor US-100 [14]

Las especificaciones técnicas más relevantes del sensor ultrasonido US-100 son las siguientes:

- Voltaje de Operación: 5V DC
- Corriente de trabajo: 15mA
- Rango de detección: 2cm-450cm
- Precisión: 3mm
- Ángulo de detección: 15°
- Frecuencia de trabajo: 40KHz
- Dimensiones: 45mm x 20mm x 15mm

### 2.10.5. Sensor de luz ultravioleta UV UVM-30A.

Este sensor es usado para la detección del índice de intensidad ultravioleta. Este tipo radiación electromagnética posee longitudes de onda más cortas que la radiación que se puede ver y pueden ser detectadas sin inconvenientes. Este módulo se basa en el UVM-30A, que posee una amplia gama espectral de 200 nm hasta 370 nm\*, como se detalla en la Figura 10.

Su señal eléctrica de salida es analógica, que puede variar respecto a la intensidad de los rayos ultravioletas.



**Figura 10** Sensor UV UVM-30A [14]

Entre las principales características del sensor ultravioleta, se tiene:

- Adecuado para medir la radiación UV del sol o de otras fuentes de luz.
- Conforme al Estándar de clasificación del Índice UV de la Organización Mundial de la Salud.
- Rango de detección UV: 200-370 nm.
- Voltaje de operación: 3 V a 5 V.
- Voltaje de salida: 0V a 1V (0-10 Índice UV).
- Exactitud: +/- 1 Índice UV.
- Temperatura de trabajo: -20°C a +85°C
- Largo, tamaño: 28mm X 12mm X 10mm de ancho.

### 2.10.6. Convertidor Voltaje DC-DC Step-Down 2A MP1584EN.

Tiene como función proporcionar un voltaje bajo de salida constante en relación con el voltaje de entrada frente a fluctuaciones del voltaje de entrada. Es capaz de soportar corrientes de salida de hasta 2A, y voltajes de entrada entre 4.5V a 28V además de voltaje de salida entre 0.8V a 20V.

Este dispositivo es un regulador de tipo (Step-Down o Buck) con una muy elevada eficacia al momento de realizar la conversión, eficiente en la regulación de línea, en la Figura 11 se detalla su composición.

El módulo minimiza el uso de componentes externos para simplificar el diseño de la fuente de alimentación. Le permite obtener el voltaje del regulador de una fuente de mayor voltaje, por ejemplo, obtener 5 V, 3,3 V, 1,8 V de una fuente o batería de 12 V. Para que funcione correctamente, el nivel de voltaje de entrada debe ser al menos 1,5 V más alto que el nivel de voltaje de salida, o habrá problemas de rendimiento. Capaz de soportar cargas de hasta 2A, con una corriente superior a 1A, se recomienda instalar un disipador de calor.



**Figura 11** Convertidor de voltaje MP-1584EN [14]

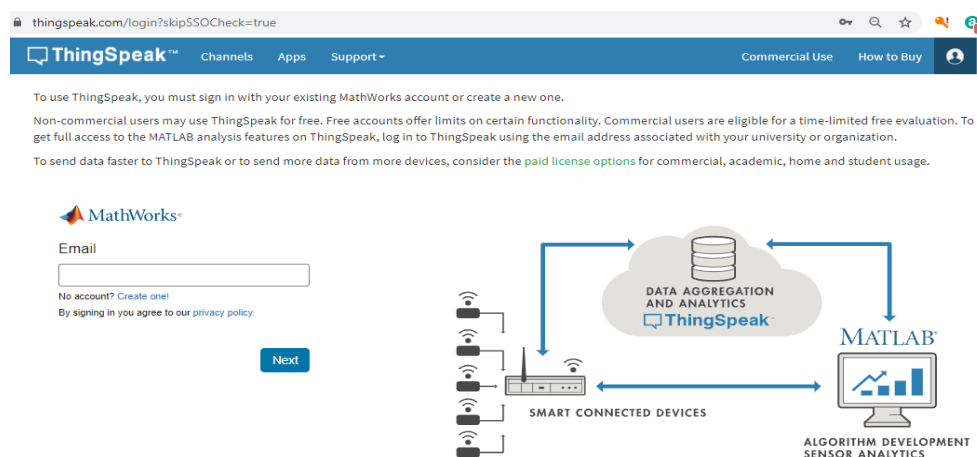
## 2.11. ThingSpeak

Es una plataforma de análisis de IoT que le permite recopilar, observar y analizar flujos de información en tiempo real en la nube y proporcionar visualizaciones instantáneas de la data enviada por sus dispositivos a ThingSpeak.

Puede analizar y procesar datos en línea tan pronto como estén disponibles. ThingSpeak se usa comúnmente para crear prototipos y validar conceptos de sistemas de IoT que requieren análisis.

ThingSpeak es una plataforma de aplicaciones abierta que permite a las personas comunicarse con los objetos. Se caracteriza por ser una plataforma de código abierto con una API para almacenar y recuperar datos orientados a objetos vía HTTP sobre Internet o sobre una red de área local (LAN). En la Figura 12 se puede observar la interfaz gráfica.

Es una plataforma basada en Ruby on Rails 3.0 (RoR), es un marco de aplicación web basado en Ruby de código abierto con arquitectura basada en el controlador Modelo Vista Controlador (MVC). Se caracteriza por una fácil programación de aplicaciones reales, escribir menos código y una configuración mucho más sencilla que otros frameworks. Otra característica que hace de RoR un framework ideal para el desarrollo de aplicaciones es que permite el uso de la programación de super aplicaciones, haciendo que su sintaxis sea más legible y accesible para un gran número de personas [15].



**Figura 12** Plataforma ThingSpeak [15]

### **3. Marco metodológico**

#### **3.1. Tipo de investigación**

Este proyecto de titulación posee como base en una investigación de tipo exploratoria, que busca tener una visión general y amplia de la realidad, que se caracteriza por su carácter plenamente preparatorio.

Es una investigación exploratoria debido a que sus variables estudiadas no se ejecutan de forma práctica en la Universidad Politécnica Salesiana de la ciudad de Guayaquil; esta es la razón por la cual, se va a desarrollar en una maqueta que va a simular un cultivo de arroz.

Las variables son las mediciones tomadas por el sistema de sensores de la maqueta, el cual indicarán los resultados para la toma de decisiones de los actuadores.

#### **3.2. Diseño de investigación**

Este proyecto se basa en información de varios trabajos de investigación y artículos científicos sobre el mismo tema o similares; además, el estudio también se basa en el análisis de diferentes prototipos para monitorear los parámetros ambientales en cultivos de gramíneas.

#### **3.3. Enfoque de la investigación**

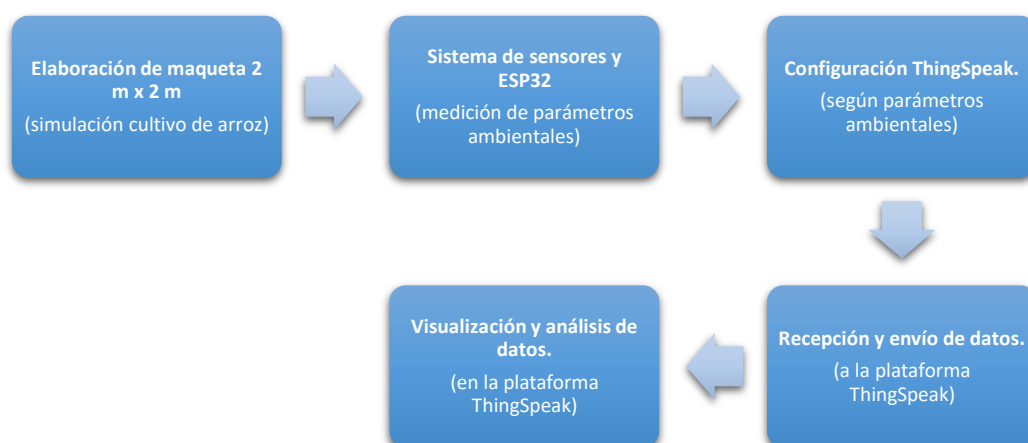
En la investigación en esta área, en vista de que las variables son consideradas desde dos partes principales, se aplica un enfoque mixto, uno cualitativo y el otro cuantitativo. Por citar un ejemplo, se realiza una evaluación teórica a partir de algunas fuentes a través de investigaciones bibliográficas como libros, artículos científicos, documentos relacionados y ciertas publicaciones en las que se estudia cualitativamente la información obtenida.

### 3.4. Metodología de investigación

Se utiliza una metodología de investigación exploratoria y experimental, ya que se realizará la construcción de una maqueta a escala de 2 metros por 2 metros para cultivar semillas de arroz y evaluar su desarrollo, durante el sembrado y su crecimiento. En esta etapa de desarrollo de la planta de arroz, se llenará de agua la maqueta de acuerdo con los niveles de indicados en los manuales técnicos del cultivo del arroz, esta función la cumple el sensor de nivel de agua, ya que enviará los datos de un módulo ESP32 de la placa principal al otro módulo ESP32 de la placa secundaria para que encienda la bomba, mediante la aplicación ThingHTTP de la plataforma ThingSpeak. Las pruebas se las realizará por 3 meses aproximadamente, durante este tiempo se va a realizar la toma de datos obtenidos por el sistema de sensores y se evaluará el desempeño del cultivo en todas sus etapas.

Se analizan los datos obtenidos por los sensores de humedad relativa, temperatura ambiente, temperatura y humedad de suelo, nivel de agua y radiación UV, mismos que a través de un dispositivo como el ESP32 y WIFI serán subidos a la plataforma de ThingSpeak, aquí, la información será procesada de tal forma que las alertas serán enviadas vía mails y vía tweets, estas alertas podrán ser utilizadas para tomar decisiones sobre la siembra de arroz, los datos recibidos serán variables de investigación que le permitirán tomar decisiones informadas y determinar con precisión cuándo sembrar arroz y sus etapas.

En la Figura 13 se muestra un esquema ordenado del proceso realizado para la elaboración del prototipo IoT.



**Figura 13** Esquema del proceso.



### **3.5. Proyectos de investigación vinculados**

Sistema de Monitoreo de Parámetros Agrícolas para la Supervisión de Cultivos de Manera Remota basado en Redes de Sensores Inalámbricos, trata de la elaboración de un sistema de sensores para llevar la medición de parámetros ambientales en el área de la agricultura, los cuales pueden ser analizados en la web [16].

Implementación de un sistema de almacenamiento de la información, monitoreo y control aplicando el internet de las cosas, para la automatización de un invernadero, se refiere al diseño y estructuración de un sistema de invernadero que puede servir como herramienta para controlar plantaciones aplicando el internet de las cosas [17].

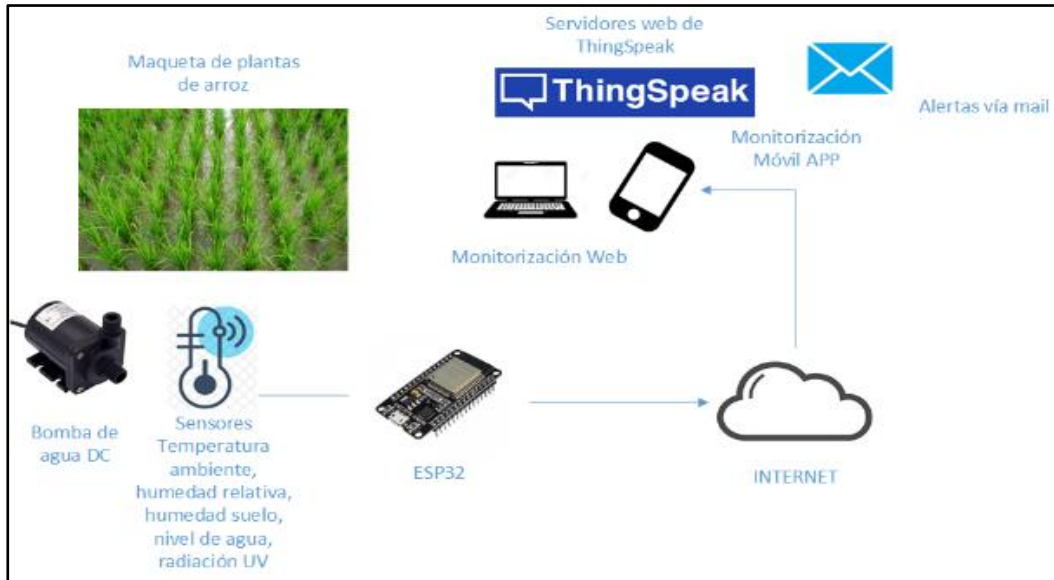
Desarrollo e implementación de un módulo de agricultura de precisión para monitoreo y control de riego, integrado al sistema IOTMACH, está encaminado a llevar un seguimiento mediante sistemas tecnológicos y también de IoT (Internet of Things), de un sistema de riego para contribuir a la agricultura y el buen desarrollo de los cultivos [18].

### **3.5. Título de la propuesta**

Diseño e implementación de un prototipo IoT para el monitoreo de parámetros ambientales aplicados al cultivo de arroz utilizando ESP32 y ThingSpeak.

### **3.6. Descripción de la propuesta**

En la Figura 14, a continuación, se puede observar la descripción de la propuesta en este trabajo de investigación.



**Figura 14** Prototipo IoT para cultivo de arroz

Se observa el esquemático del prototipo IoT para el cultivo de arroz el cual está conformado de elementos de bajo presupuesto montados sobre una maqueta acrílica donde se realizará el cultivo de plantas de arroz.

## 4. Resultados

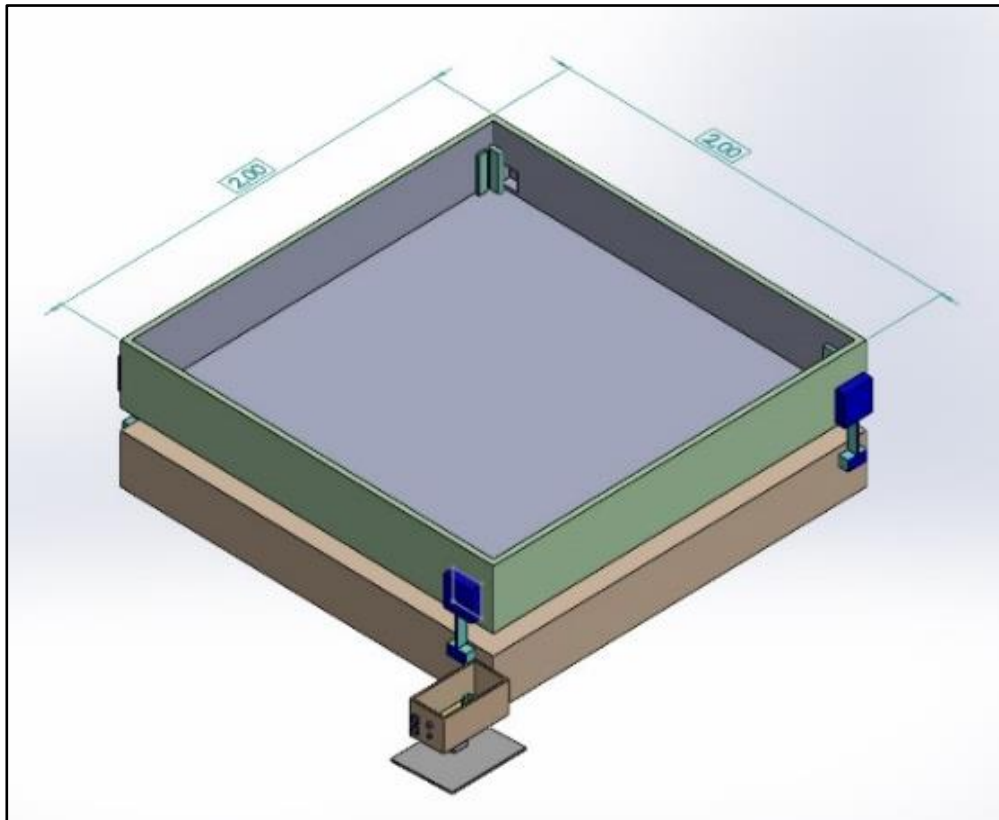
En este apartado se realiza un detalle de los resultados obtenidos en el diseño y aplicación de la maqueta del prototipo IoT para el cultivo de arroz.

### 4.1. Maqueta

La implementación a escala del cultivo de arroz se llevará a cabo en una maqueta acrílica, donde se encontrarán los sensores de humedad y temperatura del suelo, temperatura ambiente, humedad relativa, nivel de agua y radiación ultravioleta, como se puede apreciar en la Figura 15, además contará con un panel de control para verificar cada uno de los componentes mencionados para el cultivo de arroz.

La maqueta consta de una superficie estructural acrílica de 2 metros por 2 metros de área y 0,5 metros de altura, además el sistema está conformado por dos módulos ESP32, el primer módulo que se denomina placa principal, donde se encuentran instalados los sensores de humedad y temperatura del suelo, temperatura ambiente, humedad relativa, nivel de agua y radiación UV, que realiza el monitoreo del cultivo de arroz, como se puede observar en la Figura 16, y un segundo módulo que tiene por nombre placa secundaria, dicha placa está conectada con la bomba de riego para el cultivo del arroz, cabe indicar que en la estructura acrílica se coloca la tierra de sembrío y las plantas de arroz para la monitorización de su crecimiento durante las etapas de pruebas y recolección de datos. La maqueta con las dimensiones de 2 metros por 2 metros fue para las pruebas en sitio, sin embargo, para entrega al laboratorio de la Universidad Politécnica Salesiana con sede en la ciudad de Guayaquil, se realiza otra maqueta con las siguientes medidas: 0,4 metros x 0,4 metros x 0,15 metros.

Además, cuenta con una base de PVC para preservar la humedad del suelo, cerca de la maqueta existe una conexión a internet para cargar la información a la plataforma de ThingSpeak.



**Figura 15** Maqueta

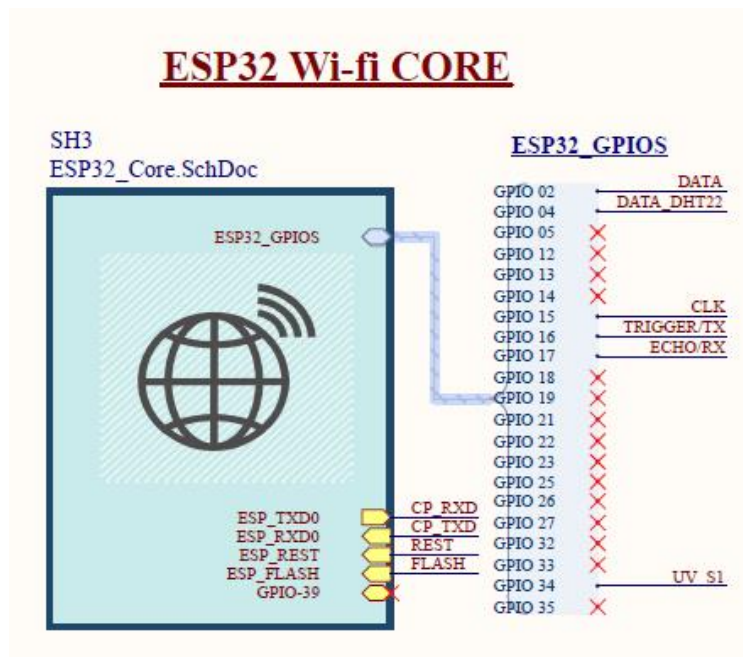


**Figura 16** Sistema de sensores

## 4.2. Diseño electrónico del prototipo

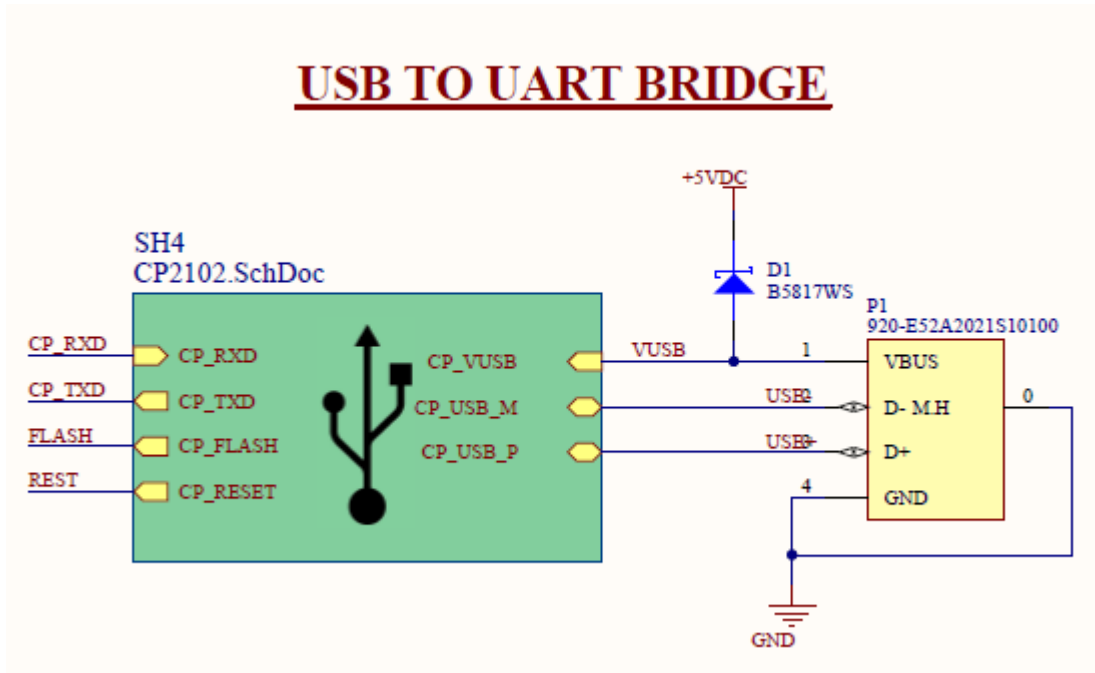
A continuación, se muestran los diseños electrónicos que conforman prototipo IoT:

En la Figura 17 se observa el esquema de conexiones del ESP32 en la placa principal utilizado en el proyecto.



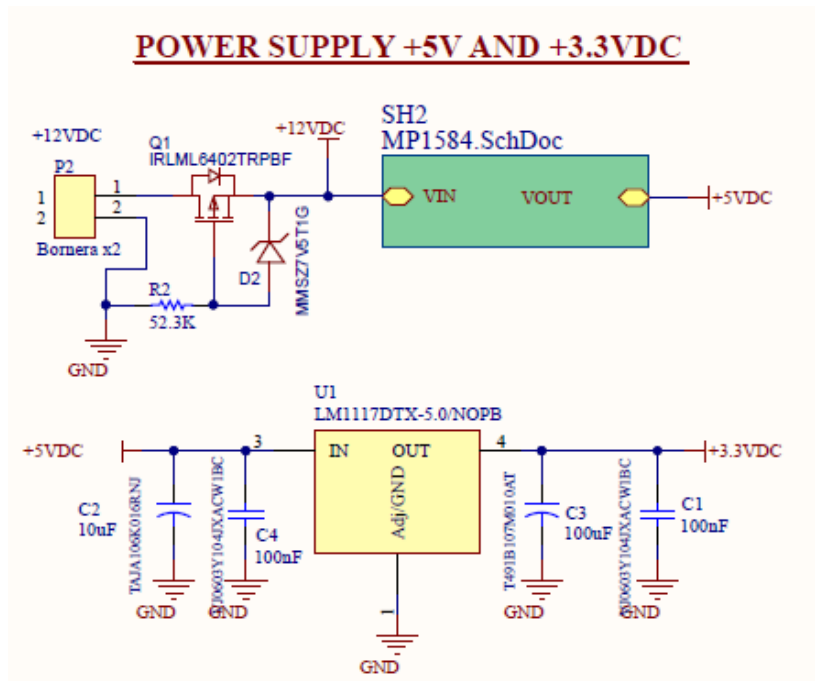
**Figura 17** Conexiones del ESP32 placa principal

En la Figura 18 se observa la conectividad USB a UART Bridge.



**Figura 18** Conexión USB UART Bridge

En la Figura 19 se observa el esquemático de conexiones del sistema de energía de la tarjeta electrónica del prototipo IoT.



**Figura 19** Diseño del power supply

En la Figura 20 se observa las conexiones de los sensores de UV, humedad y temperatura y el sensor de nivel de agua.

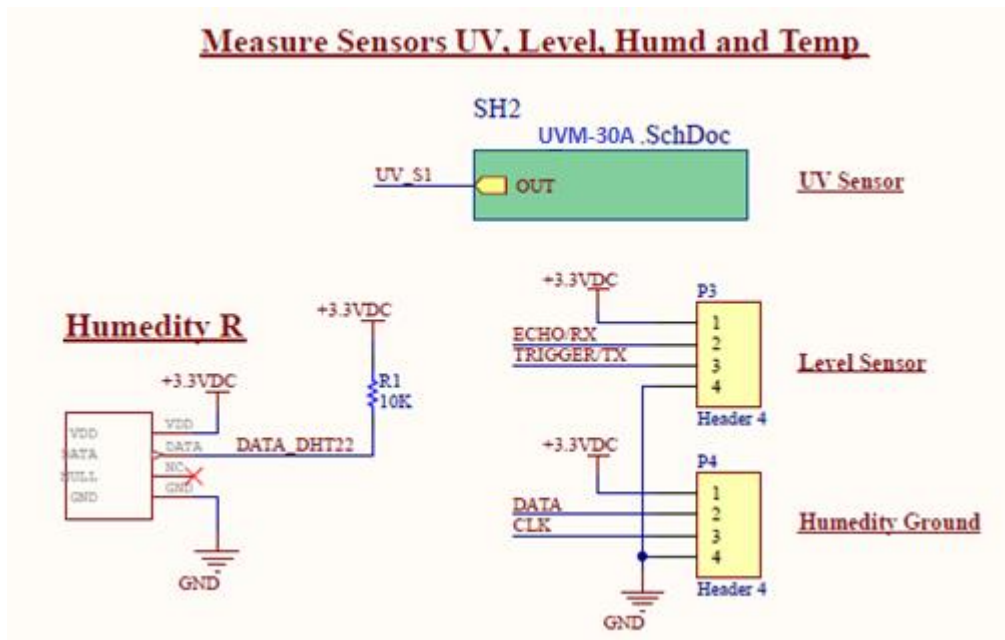


Figura 20 Conexiones de los sensores

En la Figura 21 se visualizan las conexiones GPIO del ESP32.

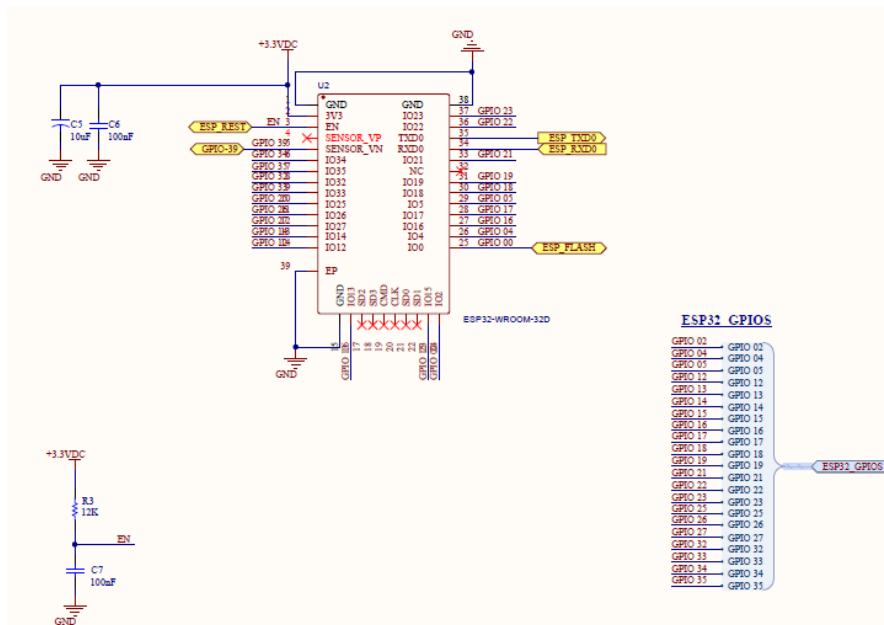


Figura 21 Conexiones GPIO del ESP32





En la Figura 24 se observa las conexiones del UVM-30A.

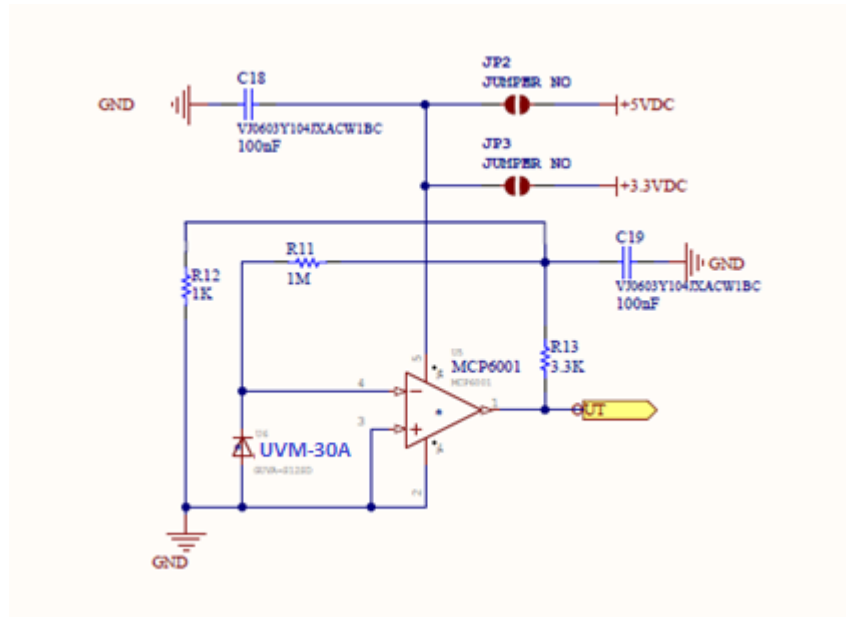


Figura 24 UVM-30A

En la Figura 25 se observa el esquemático de conexiones del ESP32 en la placa secundaria.

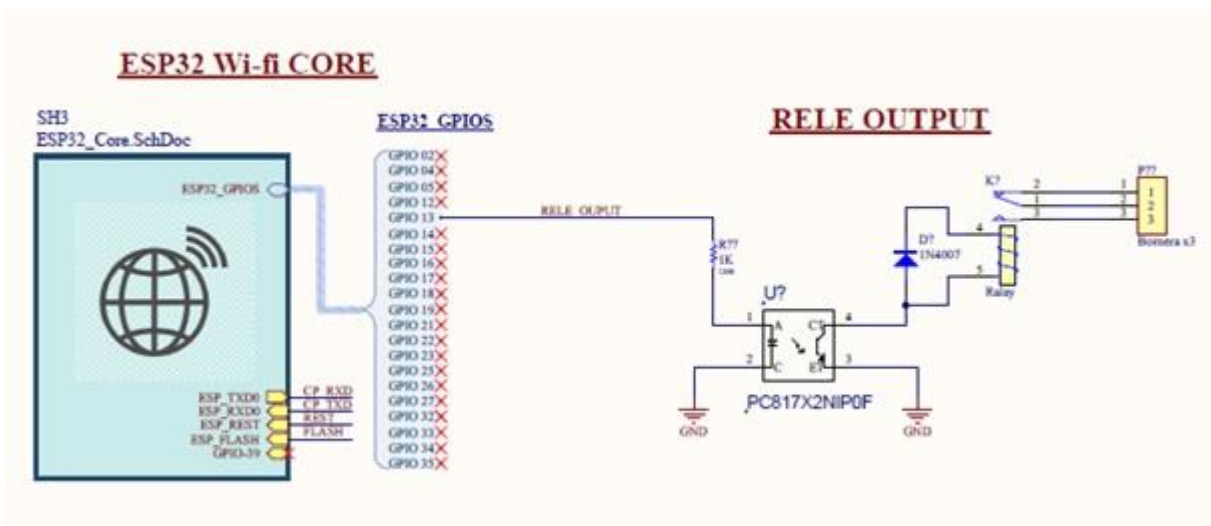
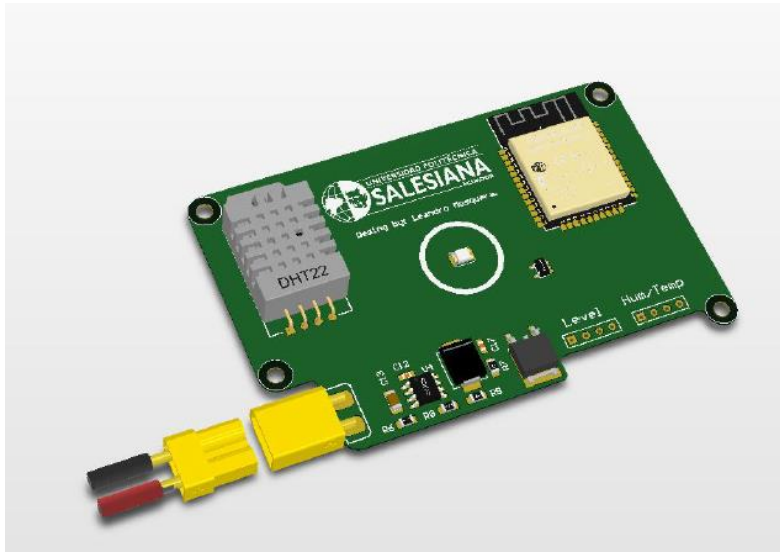


Figura 25 Conexiones del ESP32 placa secundaria

El diseño de la placa del prototipo IoT se observa en la Figura 26.



**Figura 26** Placa de prototipo en 3D

### 4.3. Programación del prototipo

A continuación, se muestran los bloques de códigos con que se configuraron los módulos ESP32 en la placa primaria y en la placa secundaria.

En la Figura 27 se observa el bloque de código que permite al módulo ESP32 conectarse a la red WiFi.

```

// CONECTANDO A RED WIFI

void setup_wifi() {
  delay(10);

  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  ThingSpeak.begin(espClient); // Inicia WIFI

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

  randomSeed(micros());

  Serial.println("");
  Serial.println("WiFi connectado");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

```

**Figura 27** Código para conectarse a la red WiFi.

El siguiente código de configuración que se muestra en la Figura 28 establece la comunicación entre el módulo ESP32 de la placa principal y la plataforma ThingSpeak para enviar los datos al ESP32 de la placa secundaria.

```

void SendGETRequest(int data)
{
  url = "http://api.thingspeak.com/update?api_key=";
  url = url + API;
  url = url + "&field";
  url = url + FieldNo;
  url = url + "=";
  url = url + data;
  http.begin(espClient,url);
  Serial.println("Esperando respuesta");
  Serial.println(" ");
  httpcode = http.GET();
  if (httpcode > 0)
  {
    Serial.println ("Datos enviados satisfactoriamente");
    Serial.println(" ");
  }
  else
  {
    Serial.println ("Error en el envio");
    Serial.println(" ");
  }
  http.end ();
}

```

**Figura 28** Código que comunica los módulos ESP32 con ThingSpeak.

La Figura 29 muestra el código que utiliza el sensor de temperatura ambiente y humedad relativa.

```
//TEMPERATURA AMBIENTE Y HUMEDAD RELATIVA // DHT22*****

float t = 0;
float h = 0;
if (dht22.read2(pinDHT22, &t, &h, NULL)){
  Serial.println("Error en la lectura.");
  return;
}
//Muestra por el puerto serie las medidas obtenidas
Serial.println("*****");
Serial.println("*TEMPERATURA AMBIENTE Y HUMEDAD RELATIVA*");
Serial.println("*****");
Serial.print("Humedad relativa: ");
Serial.print((float)h);
Serial.println(" %");
Serial.print("Temperatura Ambiente: ");
Serial.print((float)t);
Serial.println(" *C");
Serial.println(" ");
```

**Figura 29** Código para sensor de temperatura ambiente y humedad relativa.

En la siguiente Figura 30 se muestra el bloque de código empleado para el sensor de temperatura y humedad del suelo.

```
//SENSOR TEMPERATURA Y HUMEDAD DEL SUELO //SHT10*****

float temp_c;
float temp_f;
float humidity;

// Read values from the sensor
temp_c = sht1x.readTemperatureC();
temp_f = sht1x.readTemperatureF();
humidity = sht1x.readHumidity();

// Print the values to the serial port
Serial.println("*****");
Serial.println("*TEMPERATURA Y HUMEDAD DE SUELO*");
Serial.println("*****");
Serial.print("Temperatura del suelo: ");
Serial.print(temp_c, DEC);
Serial.print(" *C /");
Serial.print(temp_f, DEC);
Serial.println(" *F");
Serial.print("Humedad del suelo: ");
Serial.print(humidity);
Serial.println(" %");
Serial.println(" ");
delay(1000);
```

**Figura 30** Código para sensor de temperatura y humedad del suelo.

El sensor ultrasónico US100 se encuentra configurado por el siguiente bloque de código que se muestra en la Figura 31.

```
//SENSOR ULTRASÓNICO //US100*****
US100Serial.flush();
US100Serial.write(0x55);
delay(500);

if(US100Serial.available() >= 2){
  MSByteDist = US100Serial.read();
  LSByteDist = US100Serial.read();
  mmDist = MSByteDist * 256 + LSByteDist;
  cmDist = (mmDist)*(0.1);
  if((mmDist > 1) && (mmDist < 10000)){
    if(cmDist <= 50){
      NewcmDist = TOPE - (cmDist);
      Serial.println("*****");
      Serial.println("*NIVEL DEL AGUA*");
      Serial.println("*****");
      Serial.print("Distancia: ");
      Serial.print(NewcmDist);
      Serial.println(" cm");
      Serial.println(" ");
    } else {
      Serial.println("*****");
      Serial.println("*NIVEL DEL AGUA*");
      Serial.println("*****");
      Serial.print("Distancia: ");
      Serial.print(" 0 ");
      Serial.println(" cm");
      Serial.println(" ");
    }
  }
}
```

**Figura 31** Código para sensor ultrasónico US100.

El sensor ultravioleta se encuentra configurado por el siguiente bloque de código que muestra la Figura 32.

```
//SENSOR ULTRAVIOLETA // UVM30A*****
float sensorVoltage;
float sensorValue;
unsigned int UVIndex = 0;

sensorValue = analogRead(34);
sensorVoltage = sensorValue/4095*3.3;
UVIndex = sensorVoltage*10;

Serial.println("*****");
Serial.println("*RAYOS ULTRAVIOLETA*");
Serial.println("*****");
Serial.print("sensor reading = ");
Serial.println(sensorValue);
Serial.print("sensor voltage = ");
Serial.print(sensorVoltage); // tiene que hacer un map de los valores leídos OUT: 0V to 1V
Serial.println(" V");
Serial.print("UVIndex = ");
Serial.println(UVIndex);
Serial.println(" ");
delay(1000);
```

**Figura 32** Código para sensor ultravioleta.

Una vez que los sensores realizan las mediciones, los datos son enviados por el módulo ESP32 de la placa principal a la plataforma ThingSpeak, el código de configuración lo muestra la Figura 33.

```
//ThingSpeak
//Carga los valores a enviar
ThingSpeak.setField(1, (float)t); // TEMPERATURA AMBIENTE
ThingSpeak.setField(2, (float)h); // HUMEDAD RELATIVA
ThingSpeak.setField(3, (float)humidity); //HUMEDAD DEL SUELO
ThingSpeak.setField(4, (float)temp_c); // TEMPERATURA DEL SUELO
ThingSpeak.setField(5, (int)NewcmDist); // NIVEL DEL AGUA
ThingSpeak.setField(6, (int)UVIndex); // RAYOS ULTRAVIOLETA

if (NewcmDist <= NIVEL_AGUA_1) {
Serial.println("SE ENCIENDE BOMBA");
Serial.println(" ");
ThingSpeak.setField(7, (float)HIGH); // ESTADO DE LA BOMBA DE AGUA ENCENDIDA

} else if (NewcmDist > NIVEL_AGUA_1) {
Serial.println("SE APAGA LA BOMBA");
Serial.println(" ");
ThingSpeak.setField(7, (float)LOW); // ESTADO DE LA BOMBA DE AGUA APAGADA

}

//Escribe todos los campos a la vez.
ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
Serial.println("***Datos enviados a ThingSpeak***");
Serial.println("                ");

delay(10000);

stepvalue = NewcmDist;
SendGETRequest (stepvalue);
```

**Figura 33** Código para subir los datos a la plataforma ThingSpeak.

En la Figura 34 se observa la codificación del módulo ESP32 de la placa secundaria en la cual se encuentra la bomba de agua que se prende y se apaga según los datos receptados desde la plataforma ThingSpeak.

```

void loop()
{
  Command_string = GETCommand ();
  if (Command_string == "L1">// Lectura del nivel inferior, prende bomba
  {
    digitalWrite(PUMP, LOW);
  }
  if (Command_string == "L2">// Lectura del nivel superior, apaga la bomba
  {
    digitalWrite(PUMP, HIGH);
  }
  Serial.println(Command_string);
  delay (1000);
}
//Código que recibe la orden de la placa ESP32 principal.
String GETCommand ()
{
  http.begin(espClient,"http://api.thingspeak.com/talkbacks/44794/commands/26722102.json?api_key=APSQIAV8D16NAADS");
  httpCode = http.GET();
  if (httpCode > 0)
  {
    response = http.getString ();
    int index_start = response.indexOf("command_string:\n") + 17;
    result = response.substring (index_start, index_start+2);
  }
  else { Serial.println ("Falla de conexion con el servidor"); }
  http.end ();
  return result;
}

```

**Figura 34** Código placa secundaria.

#### 4.4. Montaje y pruebas de prototipo

El prototipo IoT es montado sobre una estructura de aluminio el cual va enterrado en la maqueta del cultivo de arroz. Tal como se muestran en la Figura 35 y Figura 36.



*Figura 35 Montaje de estructura de aluminio.*



*Figura 36 Montaje de sensores.*



En la Figura 37 y Figura 38 se observa el montaje del prototipo en la maqueta de cultivo de arroz, el prototipo va enterrado dentro de la maqueta.



**Figura 37** Montaje del prototipo en maqueta de cultivo.



**Figura 38** Alineación de prototipo IoT.

En la Figura 39 se observa el sensor UV que debe estar al aire libre para la obtención de la intensidad de los rayos solares y detectar el nivel del índice UV.



**Figura 39** Prototipo IoT, sensor UV.

En la Figura 40 se observa el módulo de la bomba de agua con su cargador de energía.



**Figura 40** Bomba de agua.

En la Figura 41 se observa el sensor de proximidad que se utilizará para calcular el nivel de agua sobre el terreno del cultivo de arroz.



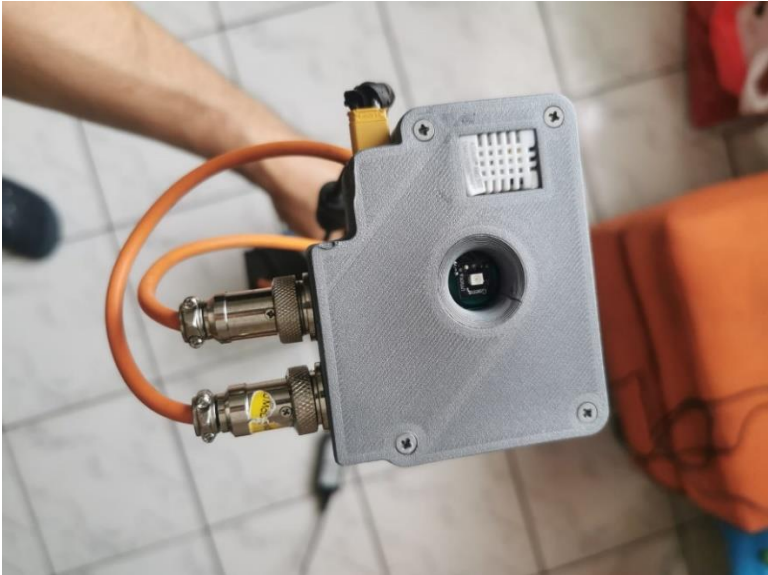
**Figura 41** Sensor de proximidad.

En la Figura 42 se observa el sensor de humedad y temperatura de suelo empotrado al soporte del prototipo IoT.



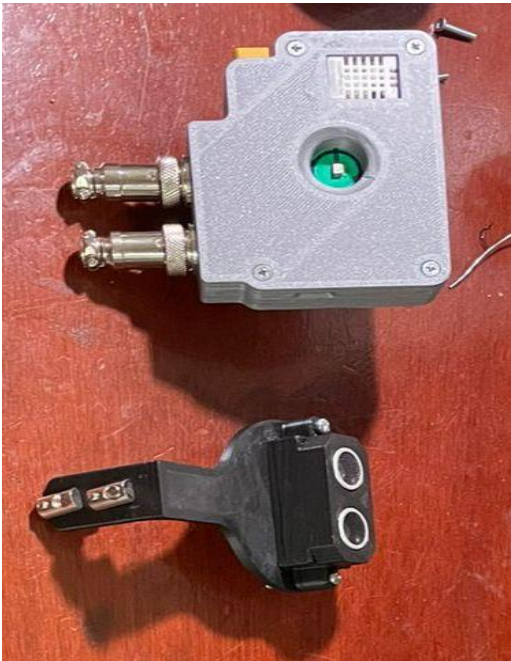
**Figura 42** Sensor de humedad y temperatura de suelo.

En la Figura 43 se observa el sensor de temperatura y humedad de ambiente junto con el sensor UV.



**Figura 43** Sensor de temperatura y ambiente.

En la Figura 44 se observan los sensores desmontados de su base.



**Figura 44** Sensores desmontados.

En la Figura 45 y Figura 46 se observan el preparado de la tierra de cultivo en la maqueta.



**Figura 45** Preparado de tierra de cultivo.



**Figura 46** Colocación de tierra en maqueta de cultivo.

En la Figura 47 y Figura 48 se observan las plantas de arroz colocadas en la maqueta para su cultivo.



**Figura 47** Maqueta con cultivo de arroz.



**Figura 48** Prototipo en cultivo de arroz.

En la Figura 49 y Figura 50 se observa el prototipo IoT listo para su funcionamiento.



**Figura 49** Testeo de prototipo en maqueta.

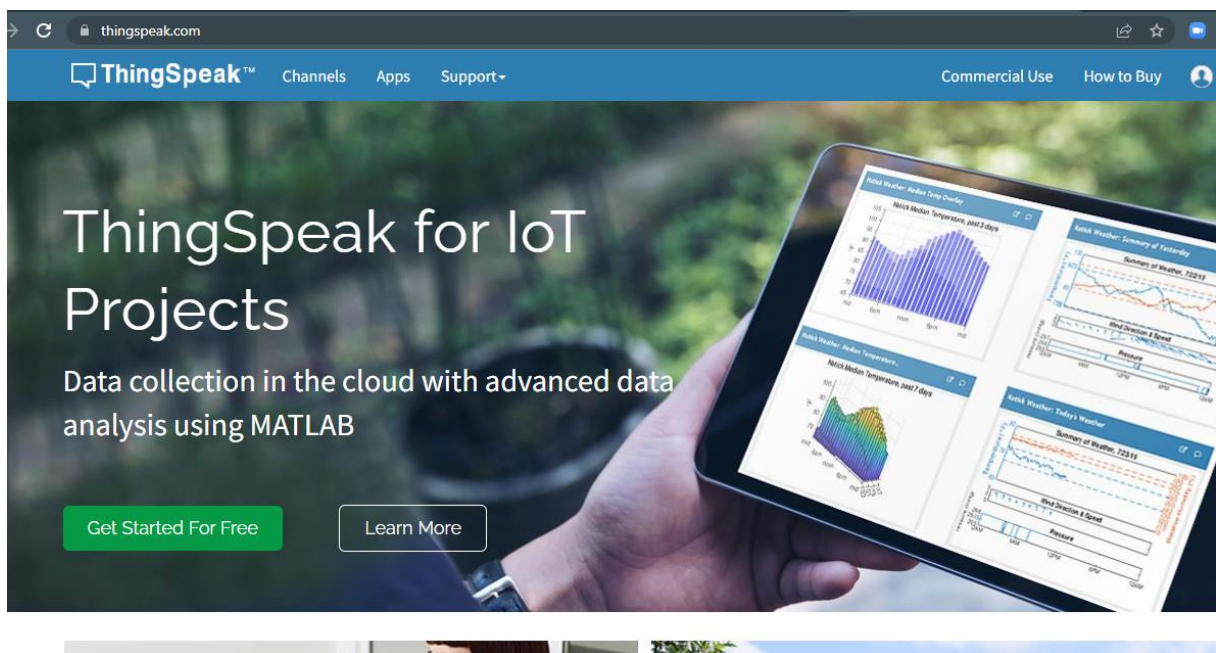


**Figura 50** Puesta en marcha de prototipo en maqueta.

## 4.5. Configuraciones en ThingSpeak

Esta sección detalla las configuraciones realizadas a nivel de ThingSpeak para el monitoreo remoto de los parámetros obtenidos por los sensores del prototipo IoT.

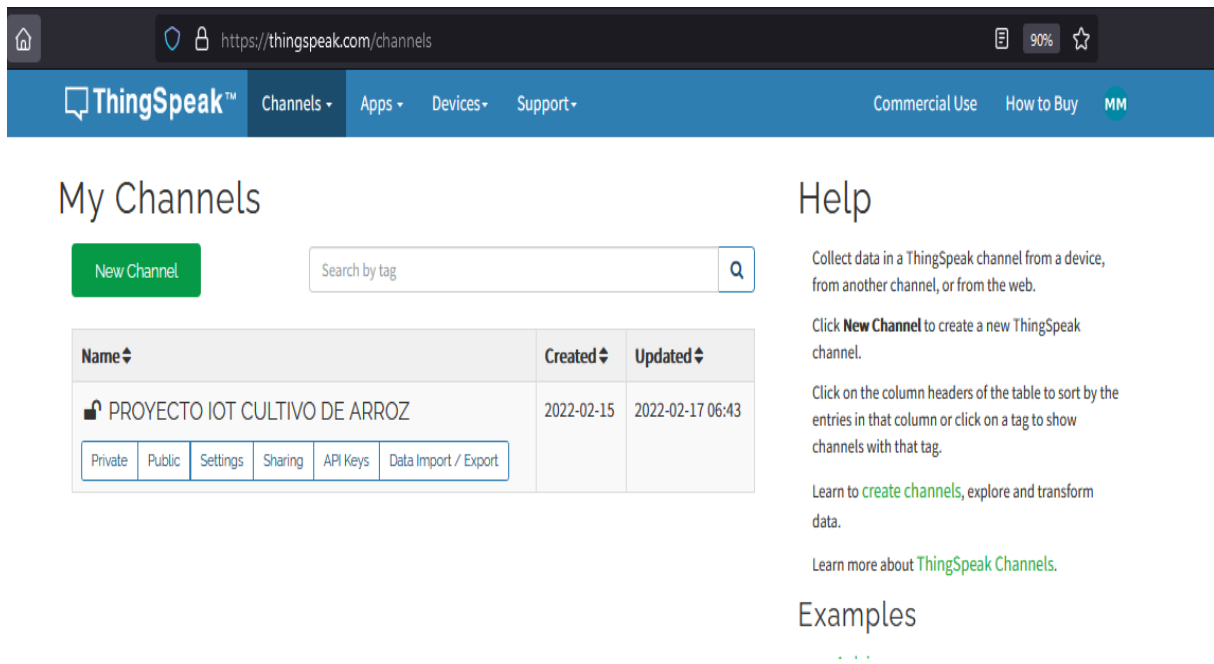
En la Figura 51 se observa la pantalla de inicio de ThingSpeak, la cual se accede con el enlace <https://thingspeak.com/>, inicialmente se debe crear una cuenta con usuario de MathWorks, se debe disponer de un usuario de MatLab para poder acceder a la herramienta de ThingSpeak.



**Figura 51** ThingSpeak.

Los estudiantes de UPS disponen de licencia de campus MathWorks con su correo institucional, para este trabajo se utiliza un correo hecho exclusivamente para el canal. Se accede con usuario y contraseña tal como se observa en la Figura 52.

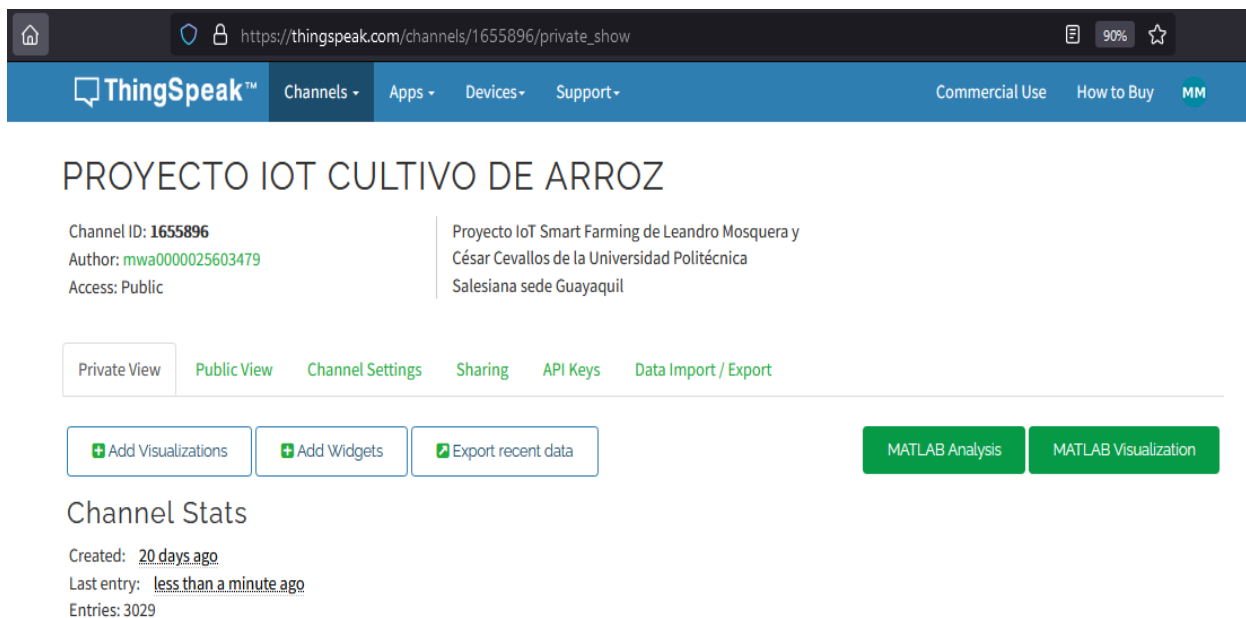




**Figura 52** Cuenta de ThingSpeak.

Se crea un proyecto denominado “Proyecto IoT cultivo de arroz”, dentro de este proyecto se procede a realizar las configuraciones necesarias para esta investigación.

En la Figura 53 se observa el canal configurado para el proyecto.



**Figura 53** Canal de ThingSpeak.

En la Figura 54 se visualizan las configuraciones básicas del canal, como el channel ID y la descripción.

The screenshot shows the ThingSpeak interface for channel ID 1655896. The page title is 'PROYECTO IOT CULTIVO DE ARROZ'. The channel ID is 1655896, the author is mwa000025603479, and the access is public. The channel description is 'Proyecto IoT Smart Farming de Leandro Mosquera y César Cevallos de la Universidad Politécnica Salesiana sede Guayaquil'. The channel settings are configured with four fields: 'Temperatura Ambient', 'Humedad Relativa', 'Humedad de Suelo', and 'Temperatura de Suelo', all of which are enabled. The 'Percentage complete' is 50%. The page also includes a 'Help' section and a 'Channel Settings' section with instructions for various fields.

Channel ID: 1655896  
Author: mwa000025603479  
Access: Public

Proyecto IoT Smart Farming de Leandro Mosquera y César Cevallos de la Universidad Politécnica Salesiana sede Guayaquil

Private View Public View Channel Settings Sharing API Keys Data Import / Export

### Channel Settings

Percentage complete 50%

Channel ID 1655896

Name PROYECTO IOT CULTIVO DE ARROZ

Description Proyecto IoT Smart Farming de Leandro Mosquera y César Cevallos de la Universidad Politécnica Salesiana sede Guayaquil

Field 1 Temperatura Ambient

Field 2 Humedad Relativa

Field 3 Humedad de Suelo

Field 4 Temperatura de Suelo

### Help

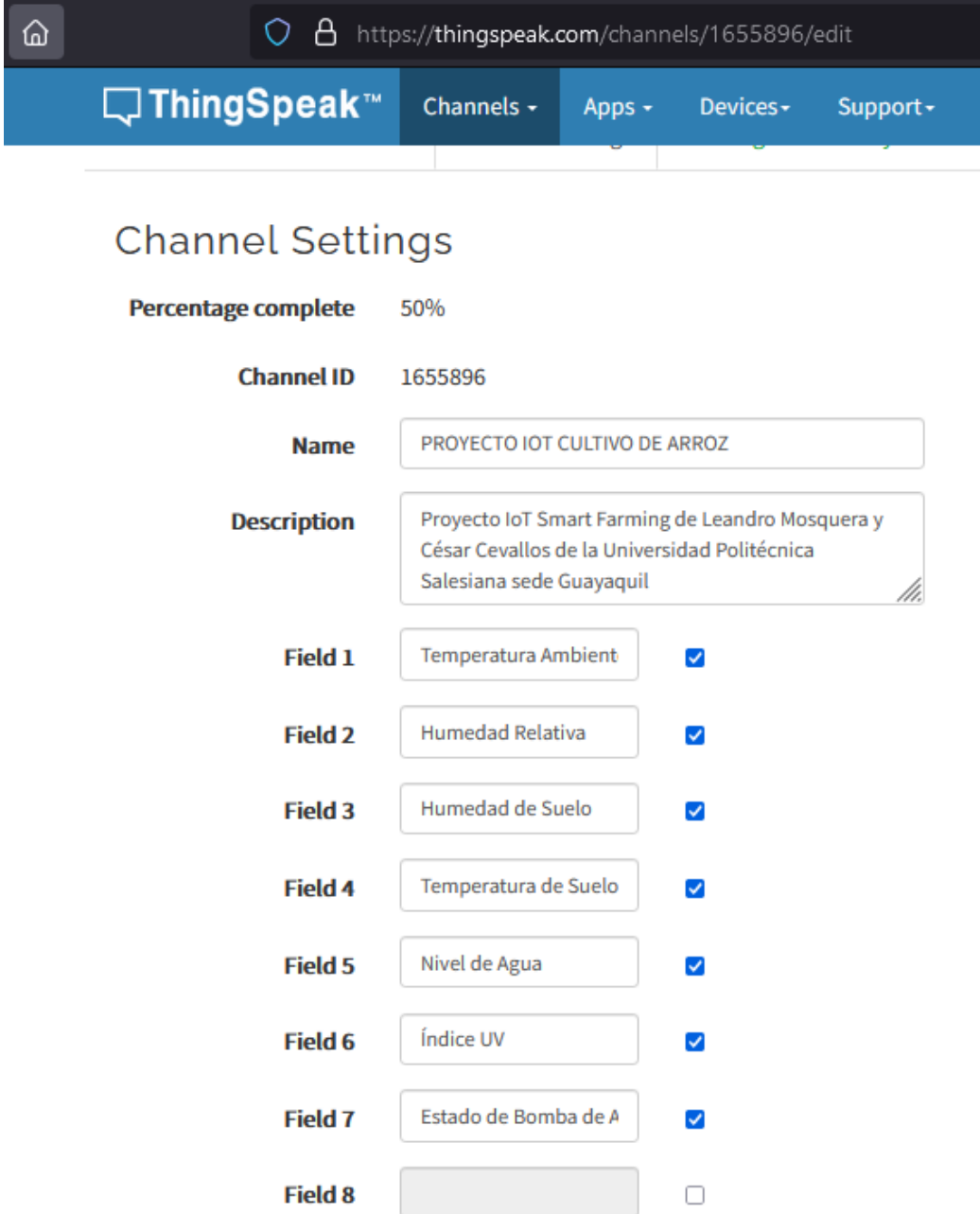
Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

### Channel Settings

- **Percentage complete:** Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- **Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- **Tags:** Enter keywords that identify the channel. Separate tags with commas.
- **Link to External Site:** If you have a website that contains information about your

**Figura 54** Configuraciones de canal.

En la Figura 55 se observan los Fields con los nombres de los sensores del prototipo.



The screenshot shows the 'Channel Settings' page for a ThingSpeak channel. The page is titled 'Channel Settings' and shows a progress indicator for 'Percentage complete' at 50%. The channel ID is 1655896. The channel name is 'PROYECTO IOT CULTIVO DE ARROZ' and the description is 'Proyecto IoT Smart Farming de Leandro Mosquera y César Cevallos de la Universidad Politécnica Salesiana sede Guayaquil'. There are eight fields listed, each with a name and a checkbox indicating if it is active. Fields 1 through 7 are active, while Field 8 is inactive.

Field	Name	Active
Field 1	Temperatura Ambient	<input checked="" type="checkbox"/>
Field 2	Humedad Relativa	<input checked="" type="checkbox"/>
Field 3	Humedad de Suelo	<input checked="" type="checkbox"/>
Field 4	Temperatura de Suelo	<input checked="" type="checkbox"/>
Field 5	Nivel de Agua	<input checked="" type="checkbox"/>
Field 6	Índice UV	<input checked="" type="checkbox"/>
Field 7	Estado de Bomba de A	<input checked="" type="checkbox"/>
Field 8		<input type="checkbox"/>

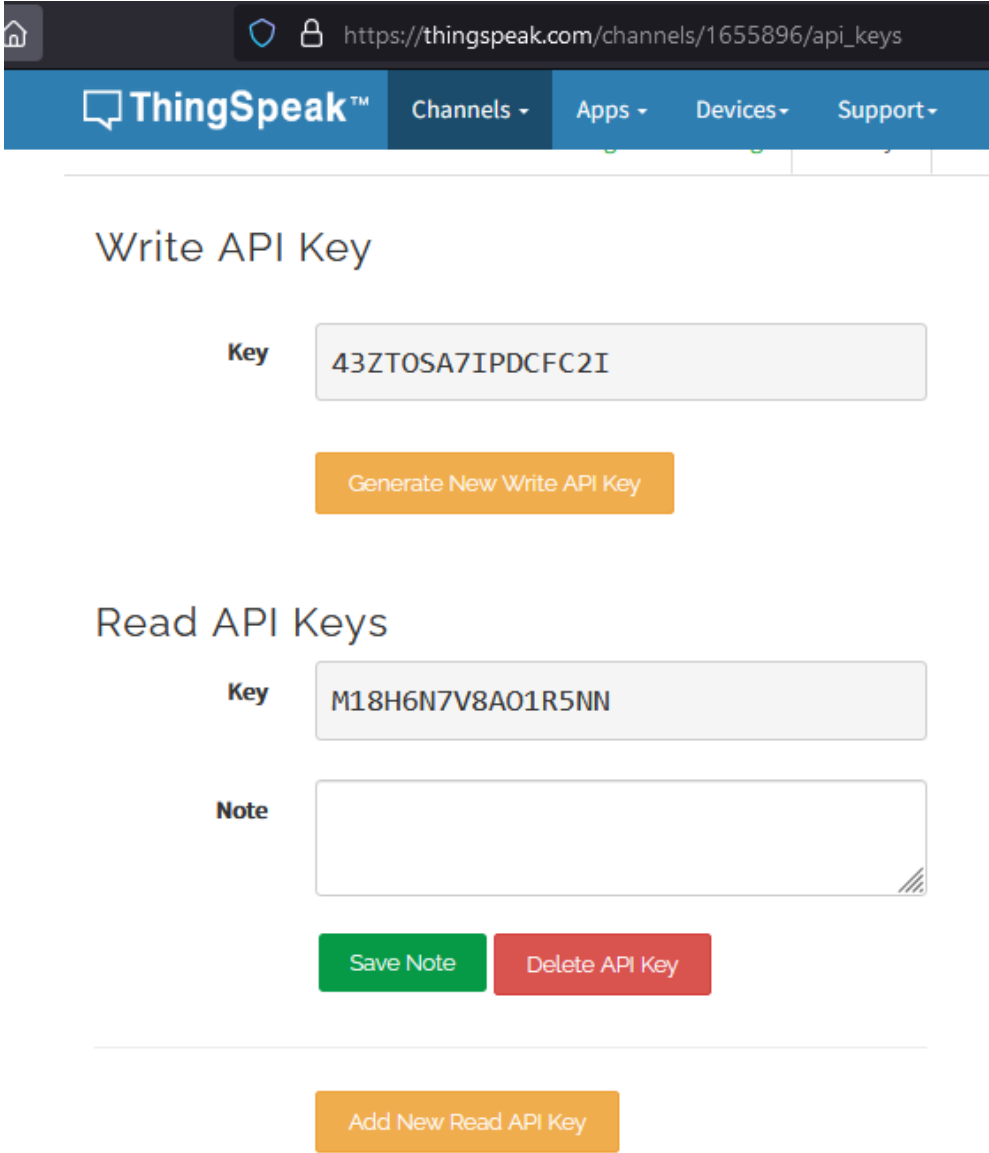
**Figura 55** Configuraciones de Fields.

En la Figura 56 se observan las configuraciones de latitud y longitud con la ubicación del prototipo.

Link to External Site	<input type="text" value="http://"/>
Link to GitHub	<input type="text" value="https://github.com/"/>
Elevation	<input type="text"/>
Show Channel Location	<input checked="" type="checkbox"/>
Latitude	<input type="text" value="-2.132149"/>
Longitude	<input type="text" value="-79.885437"/>
Show Video	<input type="checkbox"/>
	<input checked="" type="radio"/> YouTube
	<input type="radio"/> Vimeo
Video URL	<input type="text" value="http://"/>
Show Status	<input type="checkbox"/>
<input type="button" value="Save Channel"/>	

**Figura 56** Configuraciones de localidad.

En la Figura 57 se observa los API Keys de lectura y escritura.



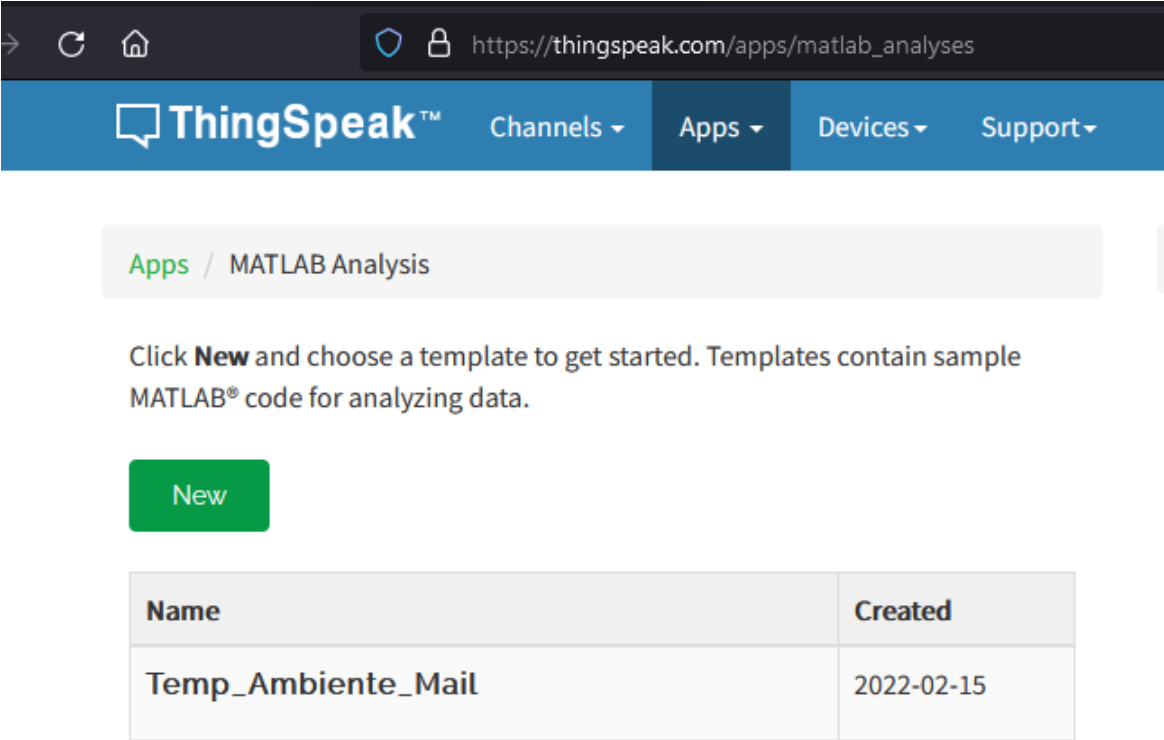
**Figura 57** API Keys.

En la Figura 58 se observa la sección de import y export para realizar respaldos de los datos obtenidos por el sistema de sensores.

The screenshot displays a web interface for data management. At the top, there is a navigation bar with tabs: Private View, Public View, Channel Settings, Sharing, API Keys, and Data Import / Export (which is active). Below the navigation bar, the main content is divided into two sections: Import and Export. The Import section is titled 'Import' and includes the instruction 'Upload a CSV file to import data into this channel.' It features a 'File' field with a 'Seleccionar archivo' button and the text 'Ningún archivo seleccionado', and a 'Time Zone' dropdown menu set to '(GMT+00:00) UTC'. A green 'Upload' button is positioned below these fields. The Export section is titled 'Export' and includes the instruction 'Download all of this Channel's feeds in CSV format.' It features a 'Time Zone' dropdown menu set to '(GMT+00:00) UTC' and a green 'Download' button below it. On the right side of the interface, there is a 'Help' section titled 'Help' with a sub-section 'Import' containing the text 'The correct format for data import is CSV with field names field1, field2, and so on.' Below this, there is a 'CSV Import Format' section with a code block showing a sample CSV row: 'created\_at,field1,field3,field4,field5\n2019-01-01T10:11:12-05:00,1,2,3,4,5'. At the bottom of the help section, there are links for 'Read Data' and 'Write Data'. The overall layout is clean and modern, with a white background and green accents for buttons and links.

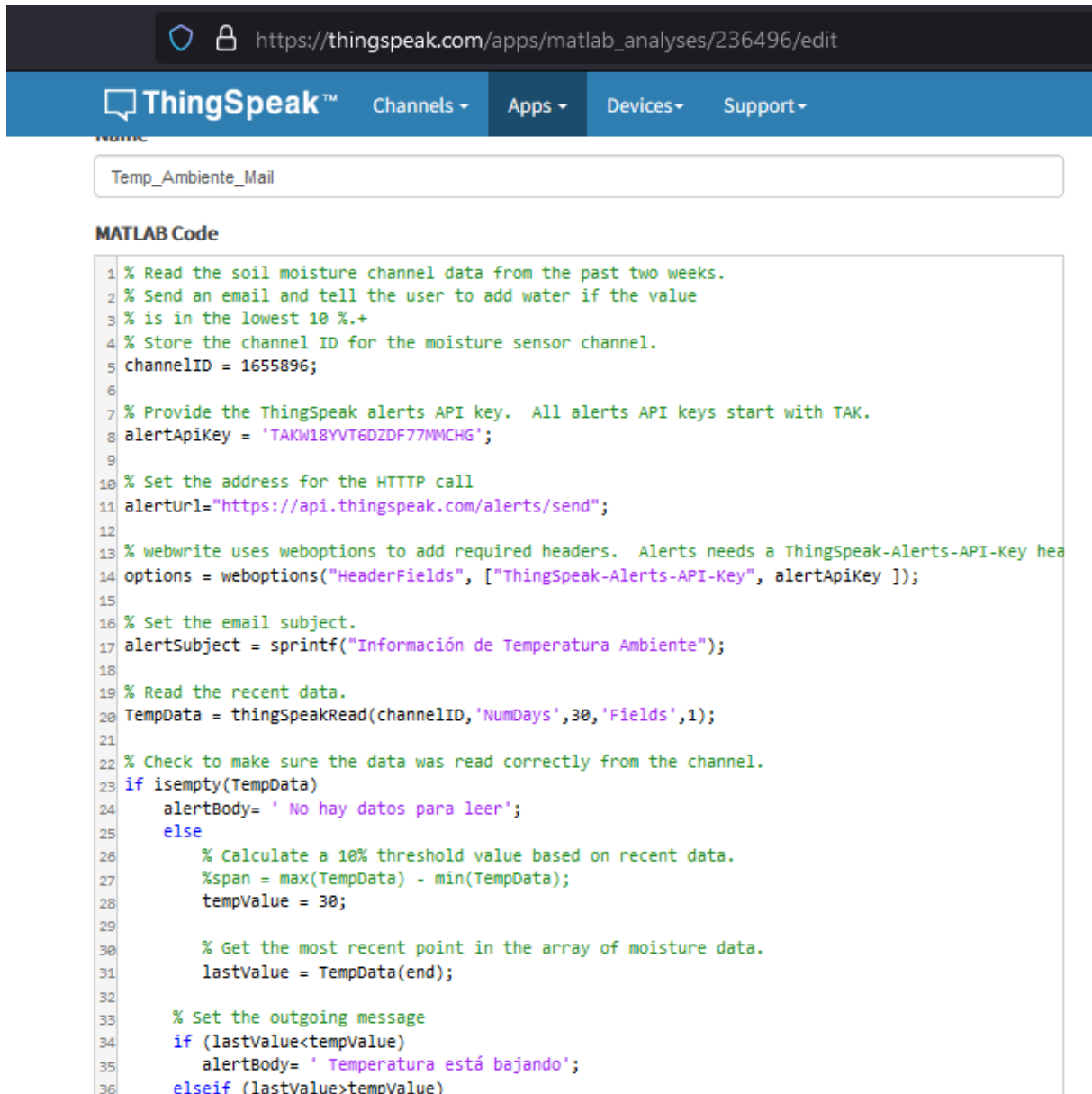
**Figura 58** Sección de importar y exportar.

En la Figura 59 se observan las configuraciones de la APP para el envío de mails de alertas, en caso de que la temperatura sobrepase el umbral configurado en el prototipo.



**Figura 59** Configuraciones de APP.

En la Figura 60 se aprecia la programación del envío de mails, el cual consiste en un código de programación de Matlab que se ejecuta cada cierto tiempo según lo configurado, de esta manera si se cumple la condición, la regla se activa y se envía el mail al configurado inicialmente en la cuenta de ThingSpeak.



The screenshot shows the ThingSpeak web interface. At the top, there is a navigation bar with the ThingSpeak logo and menu items: Channels, Apps, Devices, and Support. Below the navigation bar, the name of the app is displayed as 'Temp\_Ambiente\_Mail'. The main content area is titled 'MATLAB Code' and contains a code editor with the following MATLAB code:

```
1 % Read the soil moisture channel data from the past two weeks.
2 % Send an email and tell the user to add water if the value
3 % is in the lowest 10 %.+
4 % Store the channel ID for the moisture sensor channel.
5 channelID = 1655896;
6
7 % Provide the ThingSpeak alerts API key. All alerts API keys start with TAK.
8 alertApiKey = 'TAKW18YVT6DZDF77MMCHG';
9
10 % Set the address for the HTTP call
11 alertUrl="https://api.thingspeak.com/alerts/send";
12
13 % webwrite uses weboptions to add required headers. Alerts needs a ThingSpeak-Alerts-API-Key header
14 options = weboptions("HeaderFields", ["ThingSpeak-Alerts-API-Key", alertApiKey]);
15
16 % Set the email subject.
17 alertSubject = sprintf("Información de Temperatura Ambiente");
18
19 % Read the recent data.
20 TempData = thingspeakRead(channelID,'NumDays',30,'Fields',1);
21
22 % Check to make sure the data was read correctly from the channel.
23 if isempty(TempData)
24     alertBody= ' No hay datos para leer';
25 else
26     % Calculate a 10% threshold value based on recent data.
27     %span = max(TempData) - min(TempData);
28     tempValue = 30;
29
30     % Get the most recent point in the array of moisture data.
31     lastValue = TempData(end);
32
33     % Set the outgoing message
34     if (lastValue<tempValue)
35         alertBody= ' Temperatura está bajando';
36     elseif (lastValue>tempValue)
```


**Figura 60** APP para envío de mails.




En la Figura 61, Figura 62 y Figura 63 se observan las configuraciones de React el cual determina la acción cuando la temperatura sube o baja.

Schedule Actions

Notify me via email if this MATLAB Analysis fails when triggered by TimeControl or React.

 TimeControl

Name	Recurrence	Last Ran	Run At
<input checked="" type="checkbox"/> Temperatura Ambiente_Mail_Alert	Every 5 minutes	2022-02-23 9:55 pm	2022-02-23 10:00 pm

 React

Name	Test Frequency
<input checked="" type="checkbox"/> Temperatura Ambiente_Mail_Up	On Data Insertion
<input checked="" type="checkbox"/> Temperatura Ambiente_Mail_Down	On Data Insertion

**Figura 61** Configuración de React.

**React Name**

**Condition Type**

**Test Frequency**

**Condition** If channel

field

**Action**

**Figura 62** Configuración de acción.

**Action**

Code to execute

**Options**  Run action only the first time the condition is met  
 Run action each time condition is met

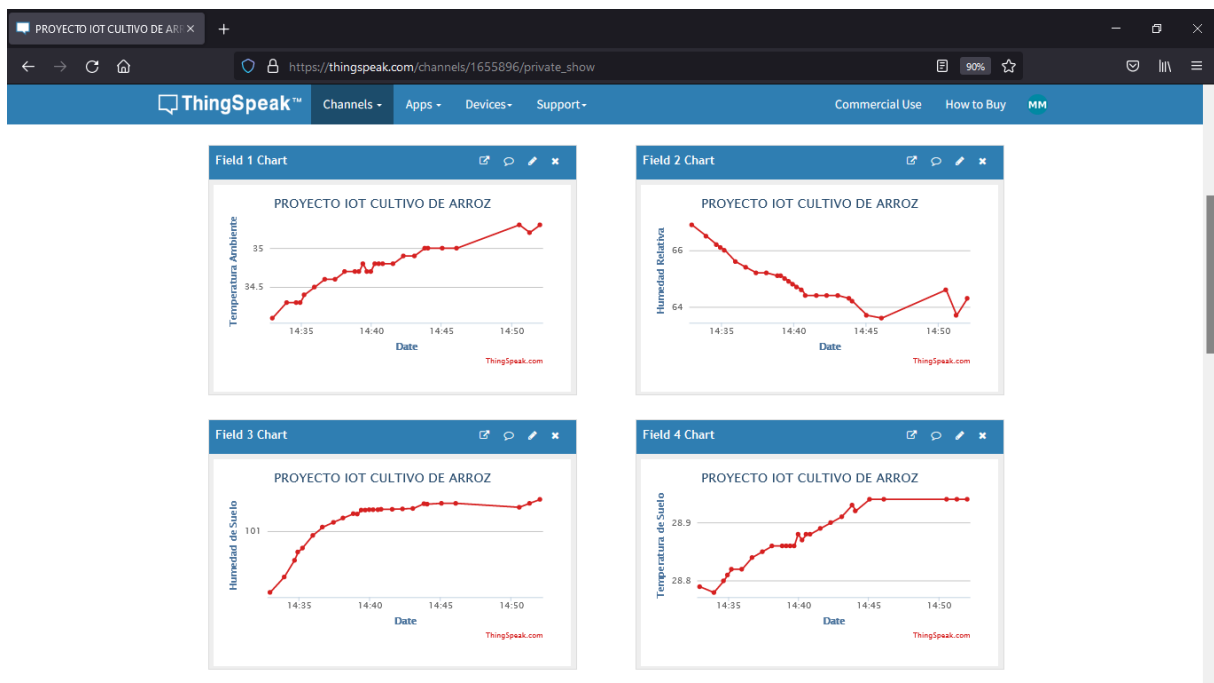
Want to delete this React?

**Figura 63** Configuración de acción y react.

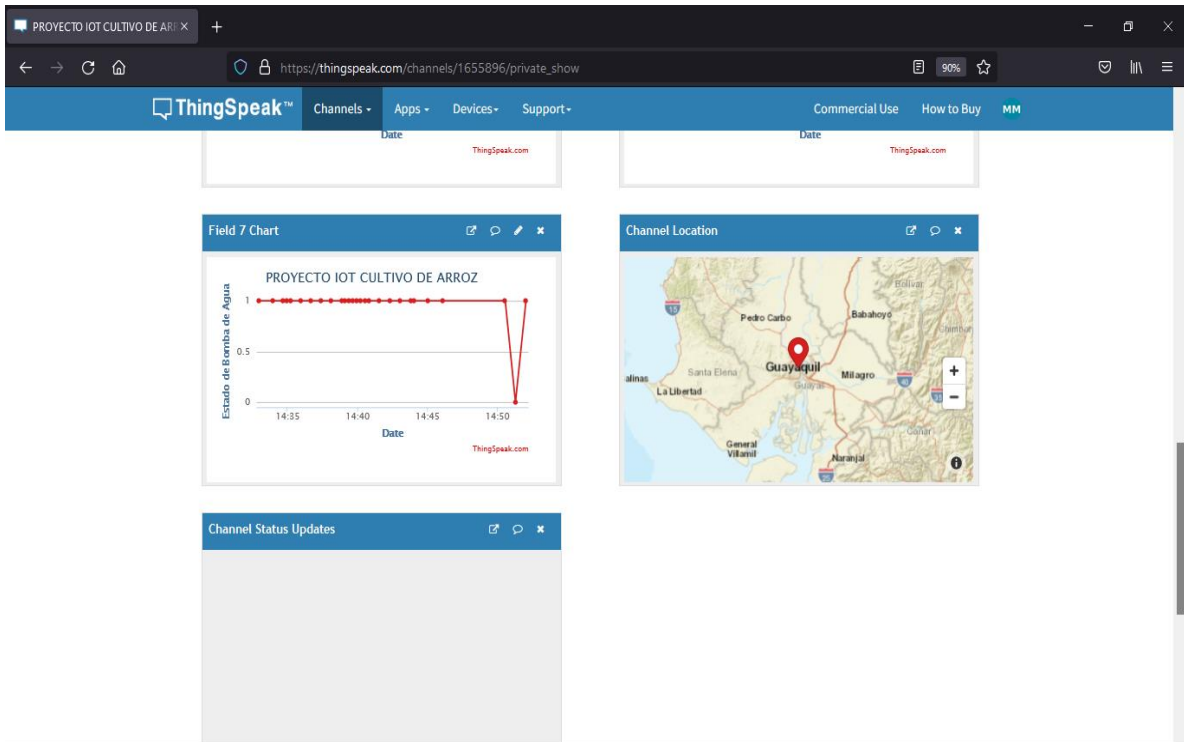
## 5. Análisis de resultados

El prototipo IoT para el monitoreo de parámetros ambientales para cultivo de arroz se implementó en el domicilio del autor de este proyecto de investigación, desde esa ubicación se tomaron los datos que a continuación se muestran en esta sección.

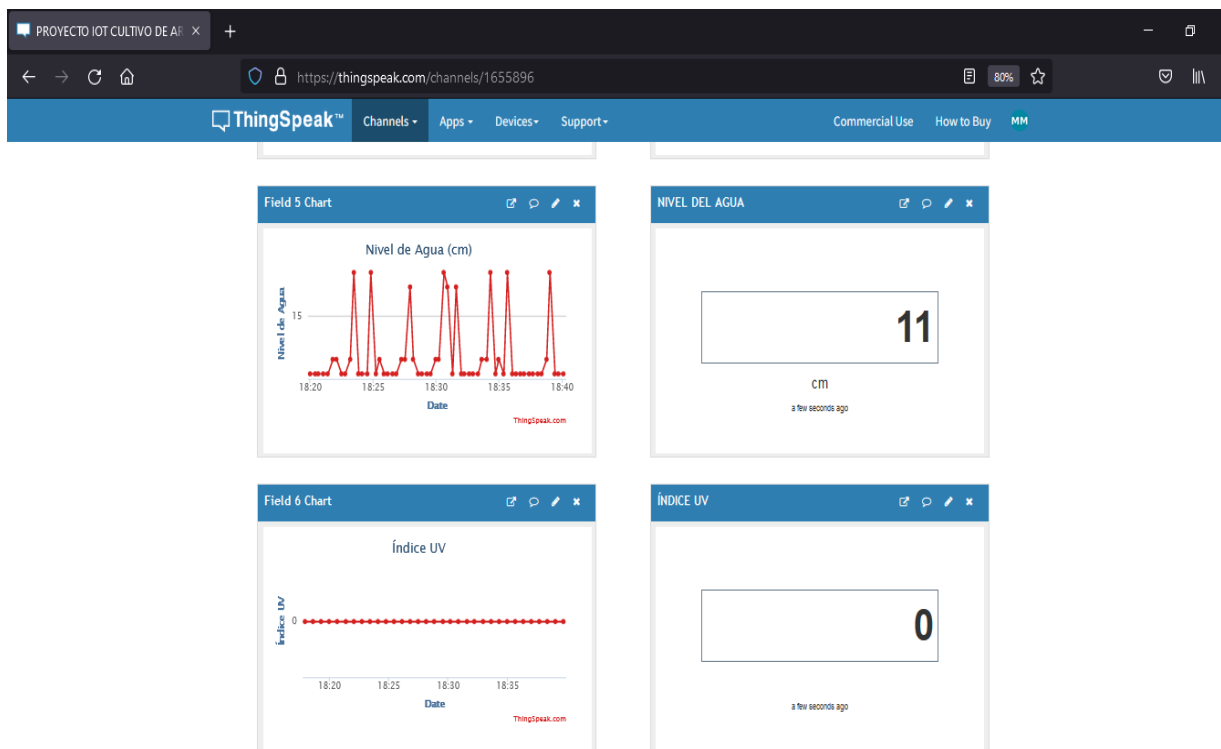
En la Figura 64, Figura 65, Figura 66, Figura 67 y Figura 68 se observan los resultados en los dashboards de los sensores, los datos visualizados son en tiempo real tomados cada 5 minutos.



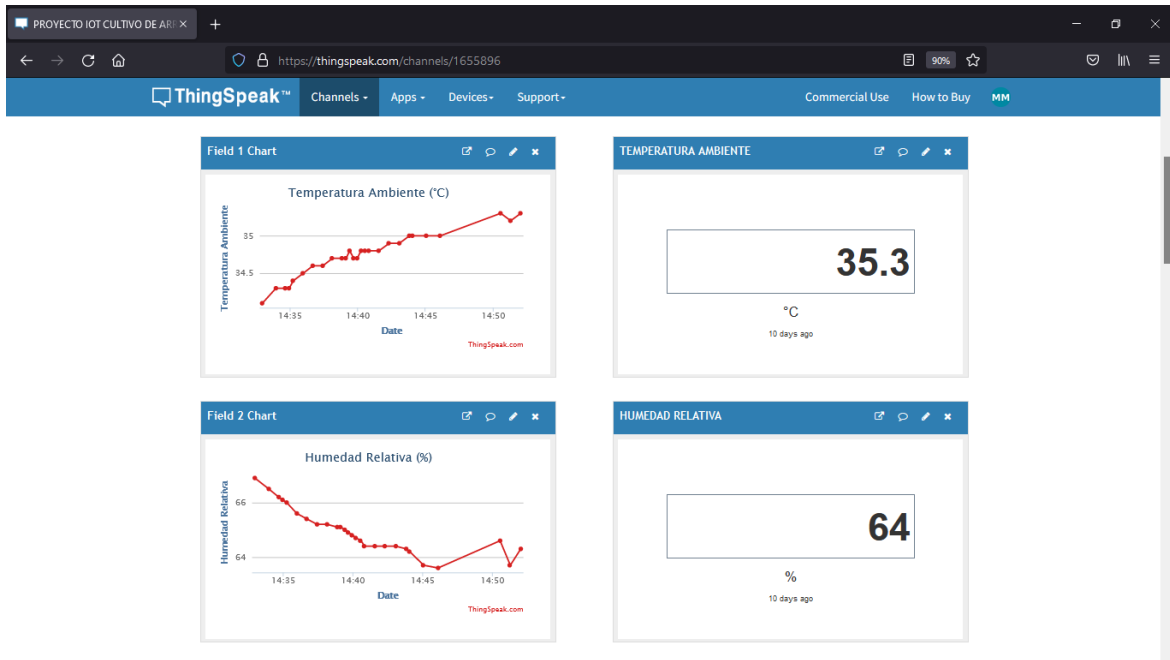
**Figura 64** Dashboards de sensores.



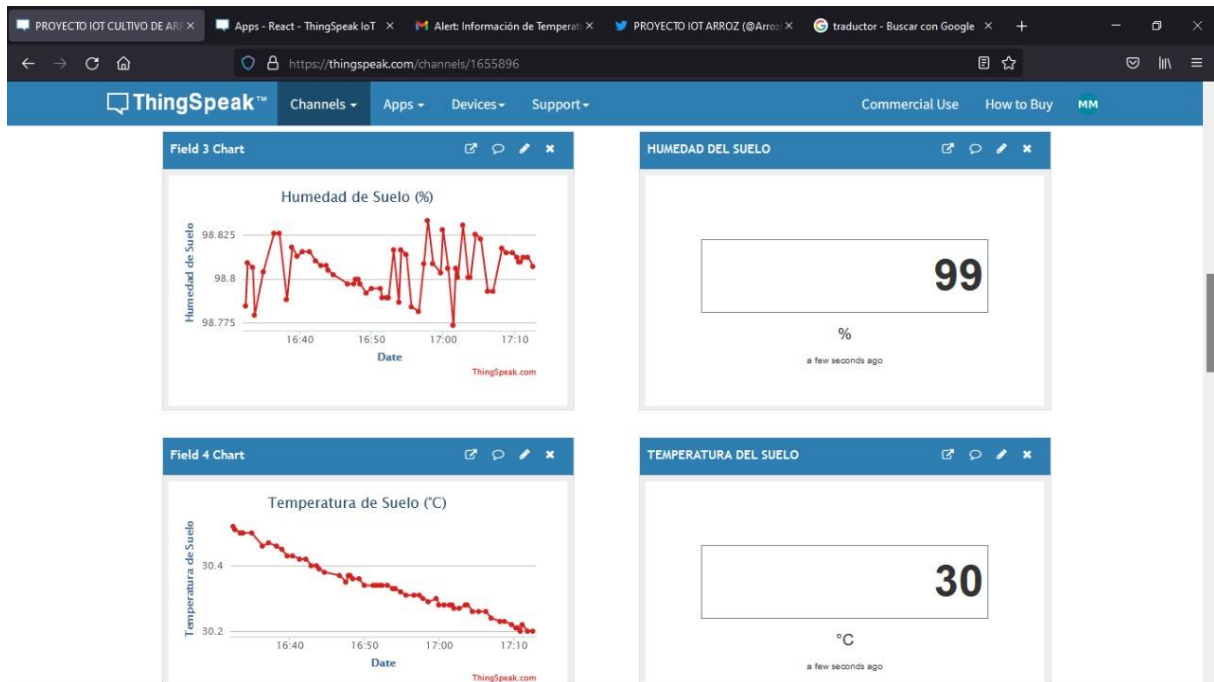
**Figura 65** Dashboards de ubicación.



**Figura 66** Datos de nivel de agua y UV.



**Figura 67** Datos de temperatura ambiente y humedad relativa.

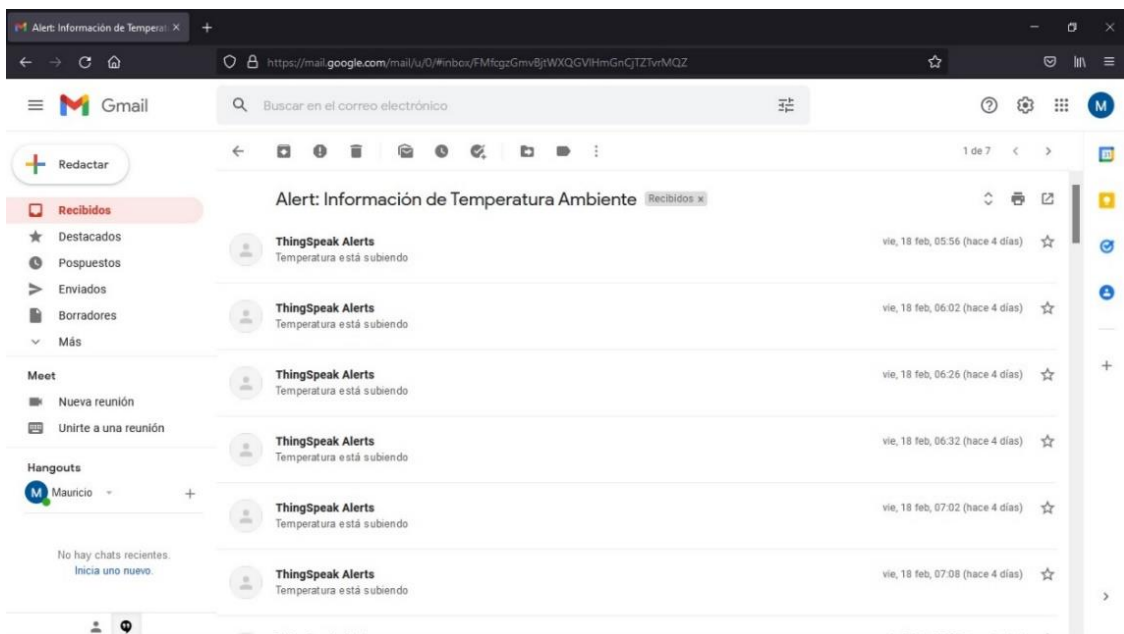


**Figura 68** Datos de Humedad y Temperatura del suelo.

Como se puede observar en esta sección los sensores toman la información y el prototipo envía los datos a la nube, específicamente a la plataforma de ThingSpeak para visualizarlos de manera gráfica y amigable. Adicionalmente se configura las reacciones para alertas por mail. El encendido de la bomba se realiza de manera automática según el umbral configurado previamente.

## 5.1. Mails de alertas recibidos

En esta sección en la Figura 69 y Figura 70, se muestran los correos recibidos como alertas de acuerdo con la configuración en ThingSpeak.



**Figura 69** Mails recibidos de ThingSpeak.

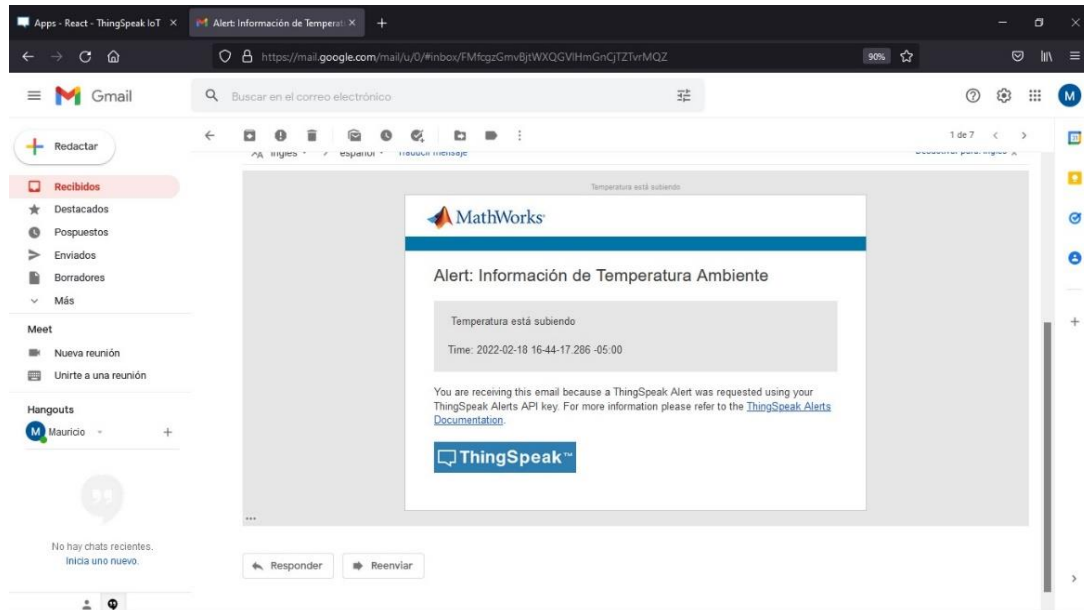


Figura 70 Alerta recibida por mail.

## 5.2. Tweets de alertas recibidos

En esta sección se muestran los tweets recibidos en la cuenta creada para este fin, la configuración en el canal de ThingSpeak de los tweets, y la configuración de los tweets de alerta, la cuenta de Twitter del proyecto es @ArrozlotTesis. Observar Figura 71, Figura 72 y Figura 73.

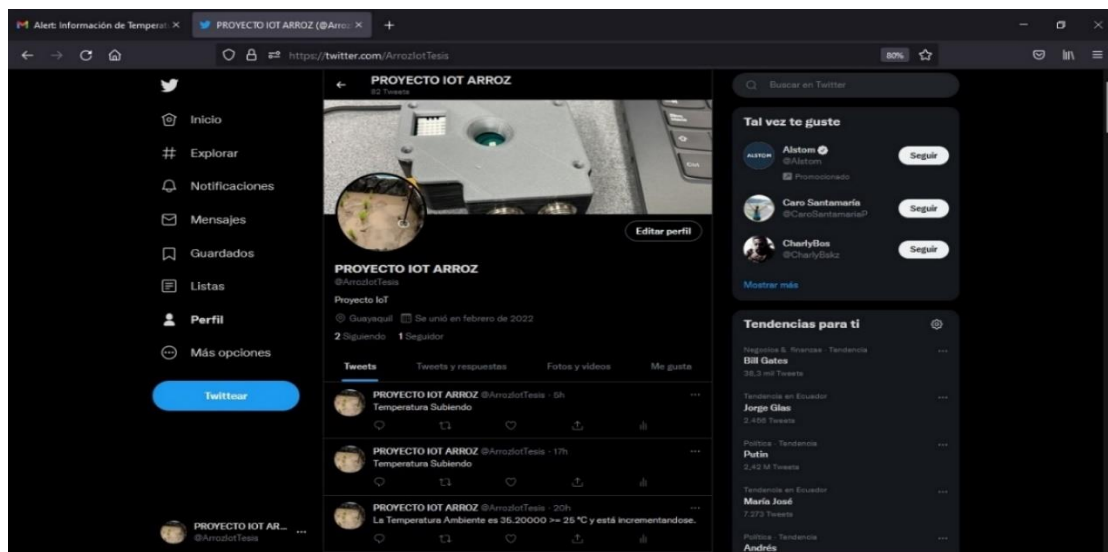


Figura 71 Cuenta de Twitter del proyecto.

Apps / ThingTweet

[Link Twitter Account](#)

Twitter Account	API Key	Action
ArrozlotTesis	388547908525UJ3T	<a href="#">Regenerate API Key</a> <a href="#">Unlink Account</a>

Reacts using ThingTweet

Name	Message	Last Sent	Twitter Account
Temp_Ambiente_Up_Tweet	La Temperatura Ambiente es %%channel_1655896_field_1%% >= 25 °C y está incrementándose.	2022-02-21 14:52	ArrozlotTesis
Nivel_Agua_Baja_Tweet	Nivel de agua es %%channel_1655896_field_5%% está bajo.	2022-02-17 16:37	ArrozlotTesis
Nivel_Agua_Alta_Tweet	Nivel de agua es %%channel_1655896_field_5%% está alto.	2022-02-21 14:52	ArrozlotTesis
Bomba_ON_Tweet	El estado de la Bomba es %%channel_1655896_field_7%%	2022-02-21 14:52	ArrozlotTesis

Help

Use the ThingTweet app to link a Twitter® account to your ThingSpeak account. Your devices can then send alerts via Twitter using ThingTweet. For example, you can make a device tweet you when the temperature in your greenhouse drops or when the batteries in a device are running low.

Click [Link Twitter Account](#) to link a Twitter account to your ThingSpeak account. When asked to authorize ThingTweet to use your account:

- Enter your Twitter username and password.
- Check the **Remember me** box to cache your Twitter credentials on your computer.
- Click **Authorize app**.

Example API Endpoint

Update Twitter Status

```
POST https://api.thingspeak.com/apps/thingtweet/1/status?api_key=388547908525UJ3T&status=I just posted this from my t
```

[Learn More](#)

Examples

[Update Twitter with ThingTweet and Arduino Ethernet Shield](#)

Figura 72 Configuraciones de Twitter en ThingSpeak.

Apps / React / Temp\_Ambiente\_Up\_Tweet / Edit

React Name: Temp\_Ambiente\_Up\_Tweet

Condition Type: Numeric

Test Frequency: On Data Insertion

Condition: If channel

PROYECTO IOT CULTIVO DE ARROZ (1655896)

field: 1 (Temperatura Ambiente)

is greater than or equal to

25

Action: ThingTweet

then tweet

La Temperatura Ambiente es %%channel\_1655896\_field\_1%% >= 25 °C y está incrementándose.

Help

React Settings

- **React Name:** Enter a unique name for your React.
- **Condition Type:** Select a condition type corresponding with your data. A channel can hold numeric sensor data, text, strings, status updates, or geographic location information.
- **Test Frequency:** Choose whether to test your condition every time data enters the channel or on a periodic basis.
- **Condition:** Select a channel, a field and the condition for your React.
- **Action:** Select ThingTweet, ThingHTTP, or MATLAB Analysis to run when the condition is met.
- **Options:** Select when the React runs.

[Learn More](#)

Figura 73 Configuraciones de Tweet alert React.



### 5.3. Resultados de encendido de la bomba

En esta sección en la Figura 74, Figura 75, Figura 76 y Figura 77, se visualizan los prints de la web q controla el encendido de la bomba y las configuraciones realizadas en ThingSpeak para el control de la bomba mediante el protocolo HTTP.

The screenshot shows the configuration page for a React named 'Bomba\_OFF React'. The 'Condition Type' is set to 'Numeric'. The 'Test Frequency' is 'On Data Insertion'. The 'Condition' is configured as 'If channel' 'PROYECTO IOT CULTIVO DE ARROZ (1655896)' with the 'field' 'S (Nivel de Agua)' and the comparison 'is less than or equal to' the value '47'. The 'Action' is 'ThingHTTP' which will 'then perform ThingHTTP' 'Bomba\_OFF'. The 'Options' are set to 'Run action each time condition is met'.

**React Name:** Bomba\_OFF React

**Condition Type:** Numeric

**Test Frequency:** On Data Insertion

**Condition:** If channel: PROYECTO IOT CULTIVO DE ARROZ (1655896)  
field: S (Nivel de Agua)  
is less than or equal to  
47

**Action:** ThingHTTP  
then perform ThingHTTP: Bomba\_OFF

**Options:**  Run action only the first time the condition is met  
 Run action each time condition is met

**React Settings Help:**

- **React Name:** Enter a unique name for your React.
- **Condition Type:** Select a condition type corresponding with your data. A channel can hold numeric sensor data, text, strings, status updates, or geographic location information.
- **Test Frequency:** Choose whether to test your condition every time data enters the channel or on a periodic basis.
- **Condition:** Select a channel, a field and the condition for your React.
- **Action:** Select ThingTweet, ThingHTTP, or MATLAB Analysis to run when the condition is met.
- **Options:** Select when the React runs.

Figura 74 Configuraciones de React para apagado de bomba.

The screenshot shows the configuration page for a React named 'Bomba\_ON React'. The 'Condition Type' is set to 'Numeric'. The 'Test Frequency' is 'On Data Insertion'. The 'Condition' is configured as 'If channel' 'PROYECTO IOT CULTIVO DE ARROZ (1655896)' with the 'field' 'S (Nivel de Agua)' and the comparison 'is greater than' the value '47'. The 'Action' is 'ThingHTTP' which will 'then perform ThingHTTP' 'Bomba\_ON'. The 'Options' are set to 'Run action each time condition is met'.

**React Name:** Bomba\_ON React

**Condition Type:** Numeric

**Test Frequency:** On Data Insertion

**Condition:** If channel: PROYECTO IOT CULTIVO DE ARROZ (1655896)  
field: S (Nivel de Agua)  
is greater than  
47

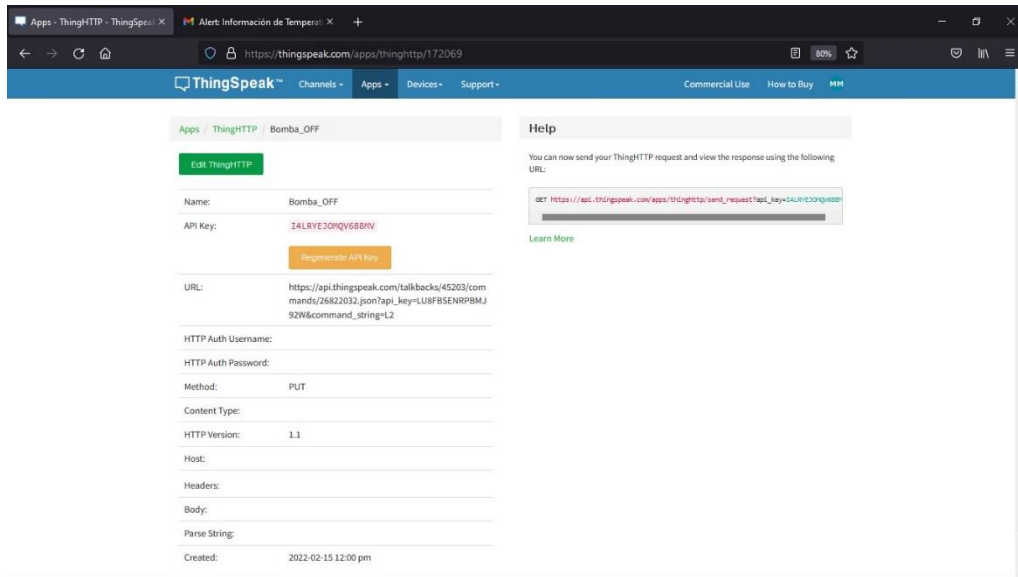
**Action:** ThingHTTP  
then perform ThingHTTP: Bomba\_ON

**Options:**  Run action only the first time the condition is met  
 Run action each time condition is met

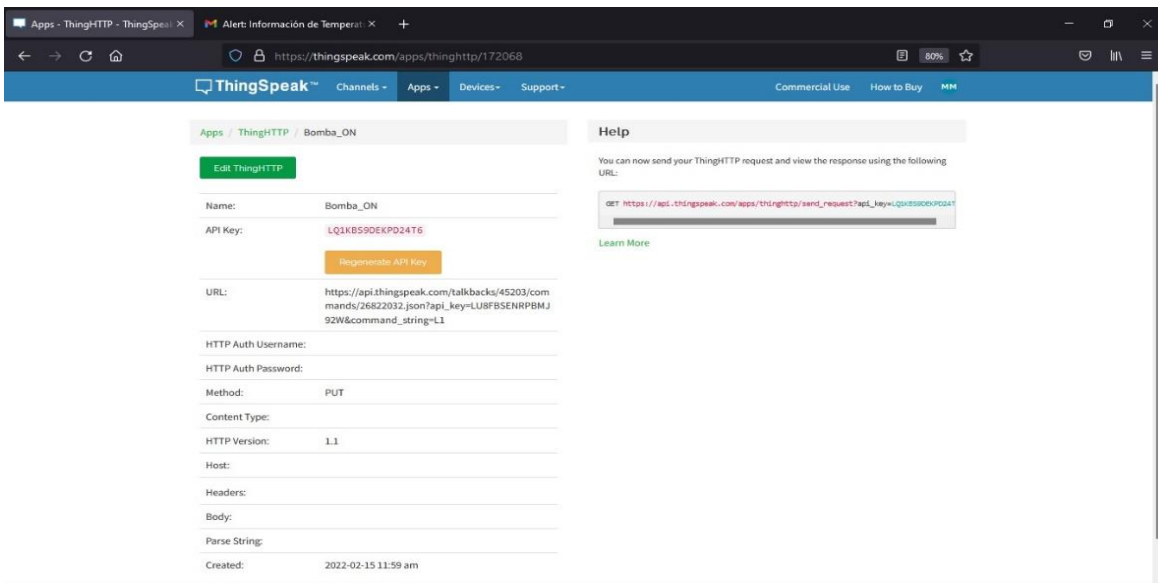
**React Settings Help:**

- **React Name:** Enter a unique name for your React.
- **Condition Type:** Select a condition type corresponding with your data. A channel can hold numeric sensor data, text, strings, status updates, or geographic location information.
- **Test Frequency:** Choose whether to test your condition every time data enters the channel or on a periodic basis.
- **Condition:** Select a channel, a field and the condition for your React.
- **Action:** Select ThingTweet, ThingHTTP, or MATLAB Analysis to run when the condition is met.
- **Options:** Select when the React runs.

Figura 75 Configuraciones de React para encendido de bomba.



**Figura 76** Configuraciones de ThingHTTP para apagado de bomba.



**Figura 77** Configuraciones de ThingHTTP para encendido de bomba.

## 5.4. Resultados de los datos obtenidos por los sensores

En esta sección se realiza el análisis de los datos obtenidos en ThingSpeak, se descarga el archivo Excel de los datos que se ha obtenido hasta el momento y se realizan gráficas para un posterior análisis.

### 5.4.1 Análisis de datos del Sensor Temperatura Ambiente

La temperatura ambiente es un factor de gran importancia para el cultivo del arroz, y en la Figura 78, se puede observar que el valor es de aproximadamente 35 grados Celsius, esto quiere decir que está dentro del rango para un crecimiento óptimo.

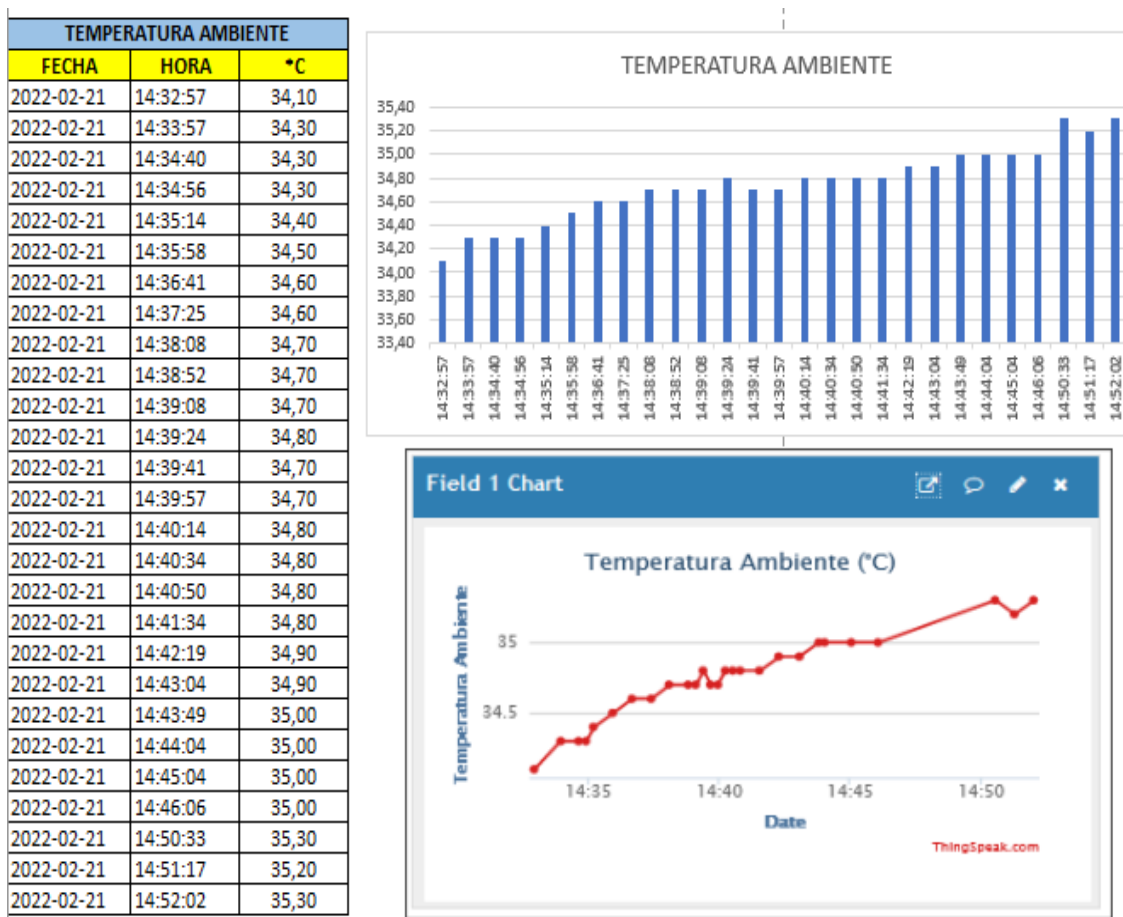


Figura 78 Datos de temperatura ambiente.

### 5.4.2 Análisis de datos del Sensor Humedad Relativa

La humedad relativa en zonas costeras, que es donde generalmente se encuentran los cultivos de arroz, tiende a ser un poco elevada en horas de la tarde, la medición de este parámetro fue de aproximadamente 67% como se observa en la Figura 79.

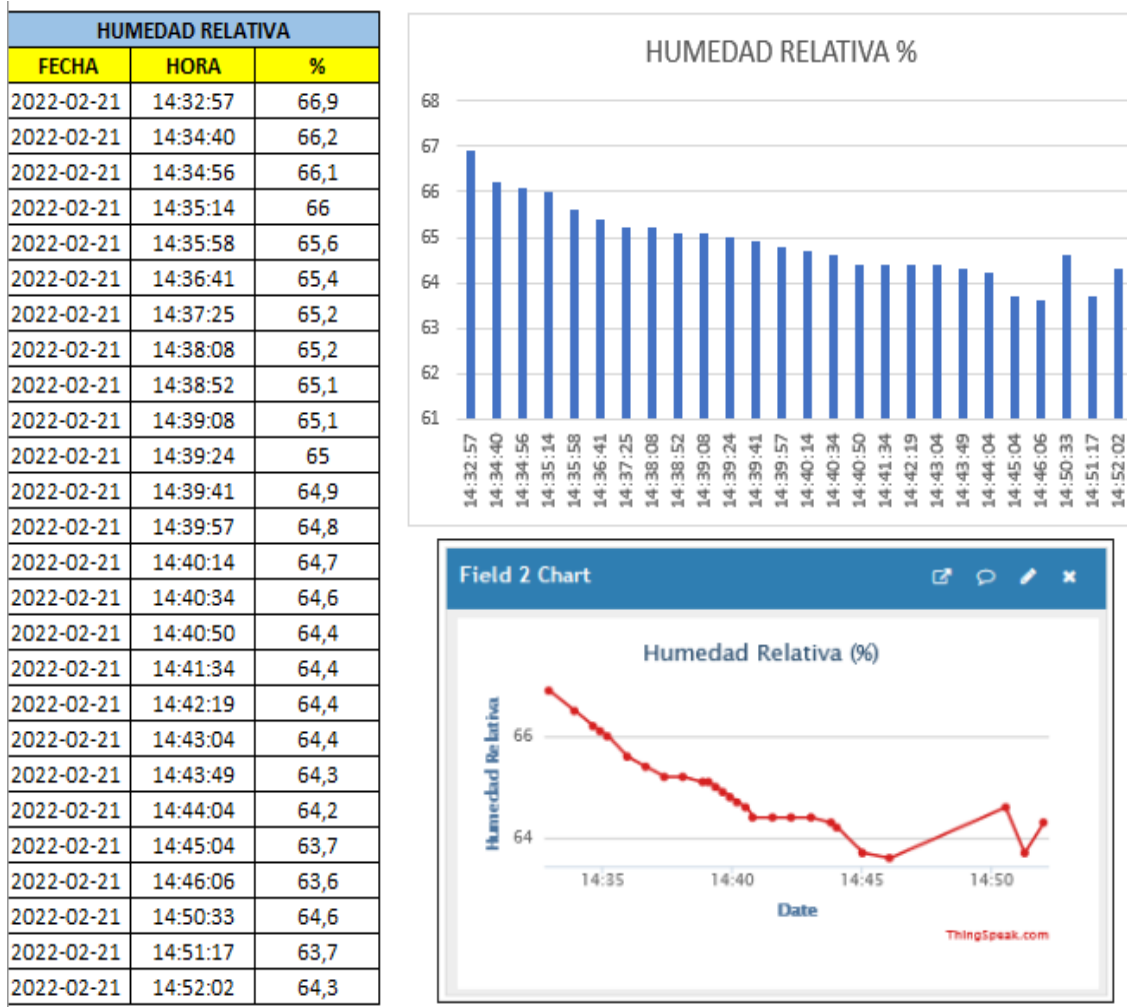


Figura 79 Datos de Humedad relativa.

### 5.4.3 Análisis de datos del Sensor Humedad del Suelo

El sensor que mide este parámetro es uno de los más precisos en su categoría, el cual nos conviene ya que la humedad del suelo juega un papel primordial para el crecimiento de la planta del arroz, debido a que por el suelo se nutre la planta y también se puede combatir enfermedades que pueden afectar al cultivo, el sensor en este caso tomó una lectura de 98.82% como máxima y de 98.78% como mínima en ese lapso, en la Figura 80 se observa el detalle de la medición tomada por el sensor de humedad del suelo.

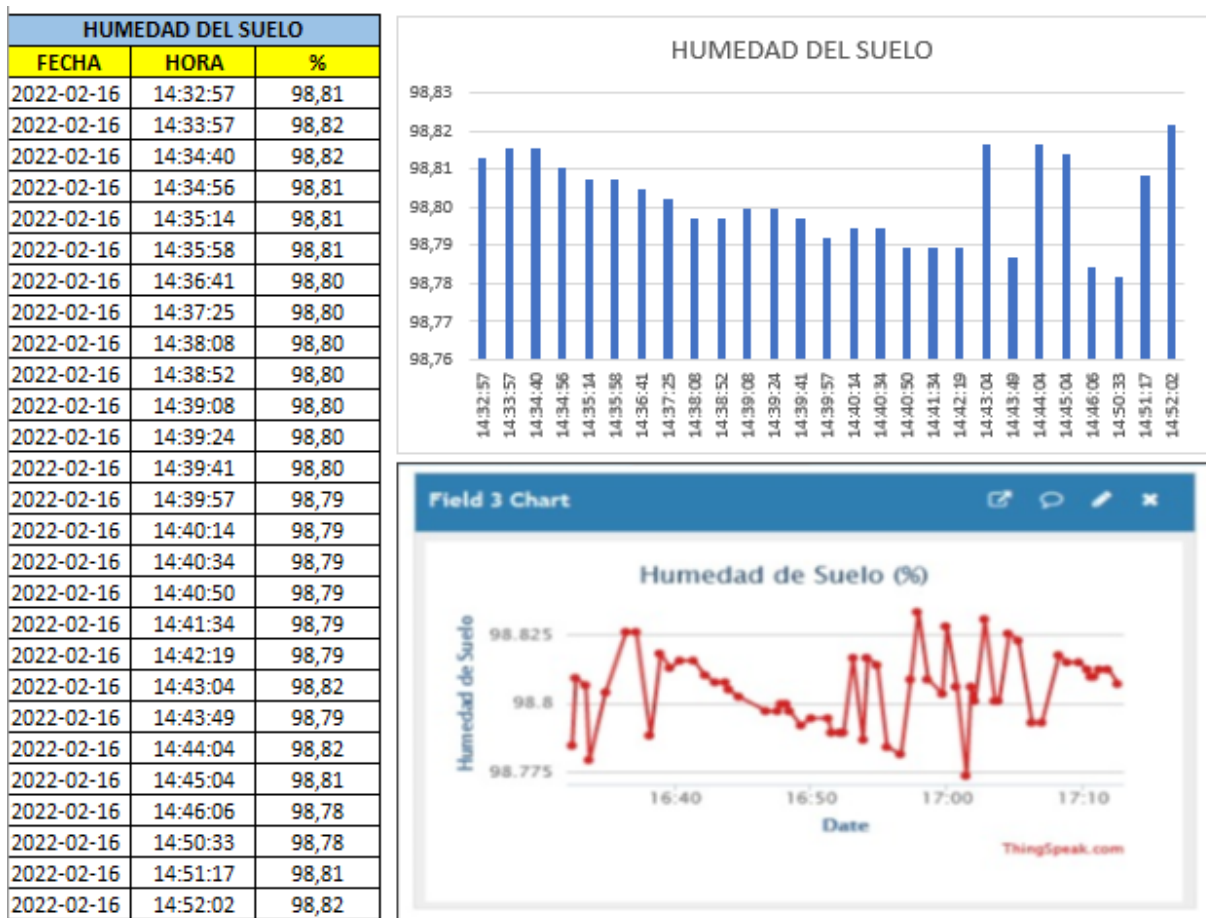


Figura 80 Datos de Humedad de suelo.

#### 5.4.4 Análisis de datos del Sensor Temperatura del Suelo

Los datos fueron tomados por el Sensor SHT-10, como se muestra en la Figura 81, éste mide también la humedad del suelo, como se mencionó anteriormente, es uno de los más precisos, nos proporcionó valores entre 28 grados Celsius y 29 grados Celsius, los cuales nos manifiestan que la temperatura del suelo se encuentra óptimas condiciones para el cultivo.

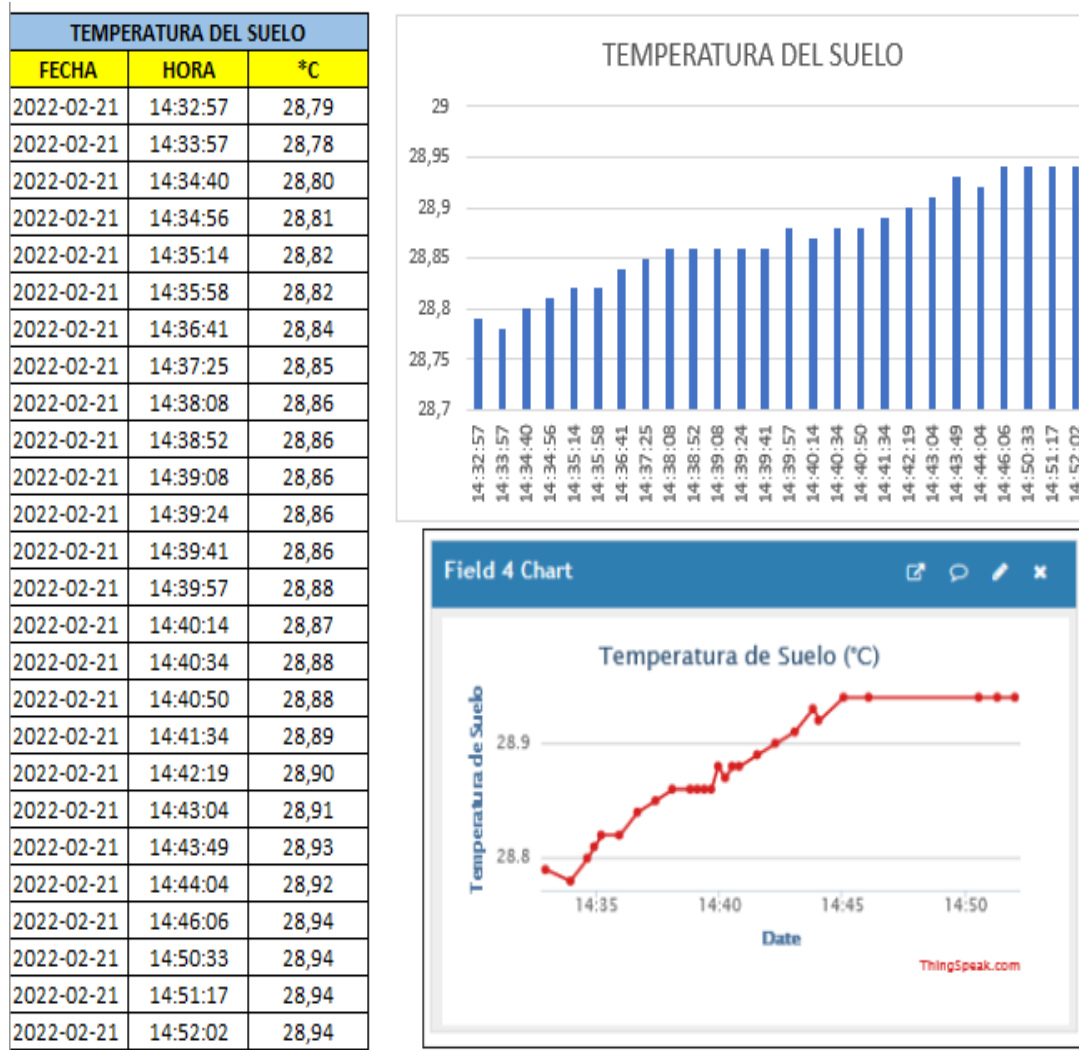


Figura 81 Datos de temperatura de suelo.

### 5.4.5 Análisis de datos del Sensor del Nivel del Agua

Este sensor es el encargado de medir la altura del agua, es el que envía la orden de que se encienda y se apague la bomba de riego, en la Figura 82 se detalla que su lectura es de 12 cm, lo que nos manifiesta es que la bomba se encuentra apagada, ya que, si fuera menor a 10 cm, la bomba automáticamente se enciende.

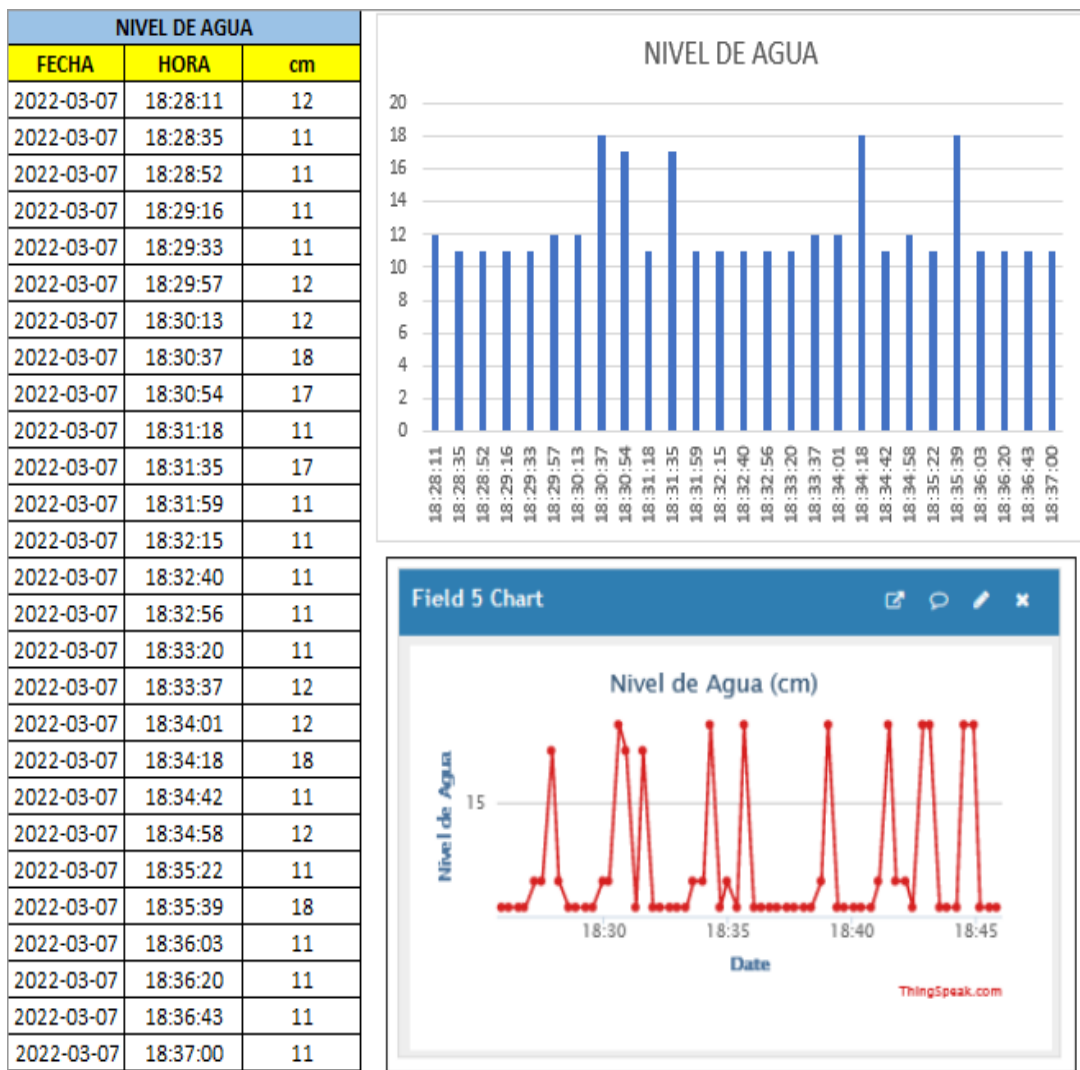


Figura 82 Datos de nivel de agua.

### 5.4.6 Análisis de datos del Sensor del Índice UV

Los rayos UV también tienen un papel primordial en el cultivo del arroz, porque el aumento de los rayos podría afectar a la planta y a su vez al agricultor, pudiendo ocasionar daños en la piel, el sensor nos indica una lectura de 0 en el índice UV, debido a que se realizó la práctica en un día nublado, pero, se observa un valor de 6, esto fue porque se hizo la prueba del sensor con una lampara de rayos UV, para demostrar que el sensor se encontraba en perfecto funcionamiento, como se puede observar en la Figura 83.

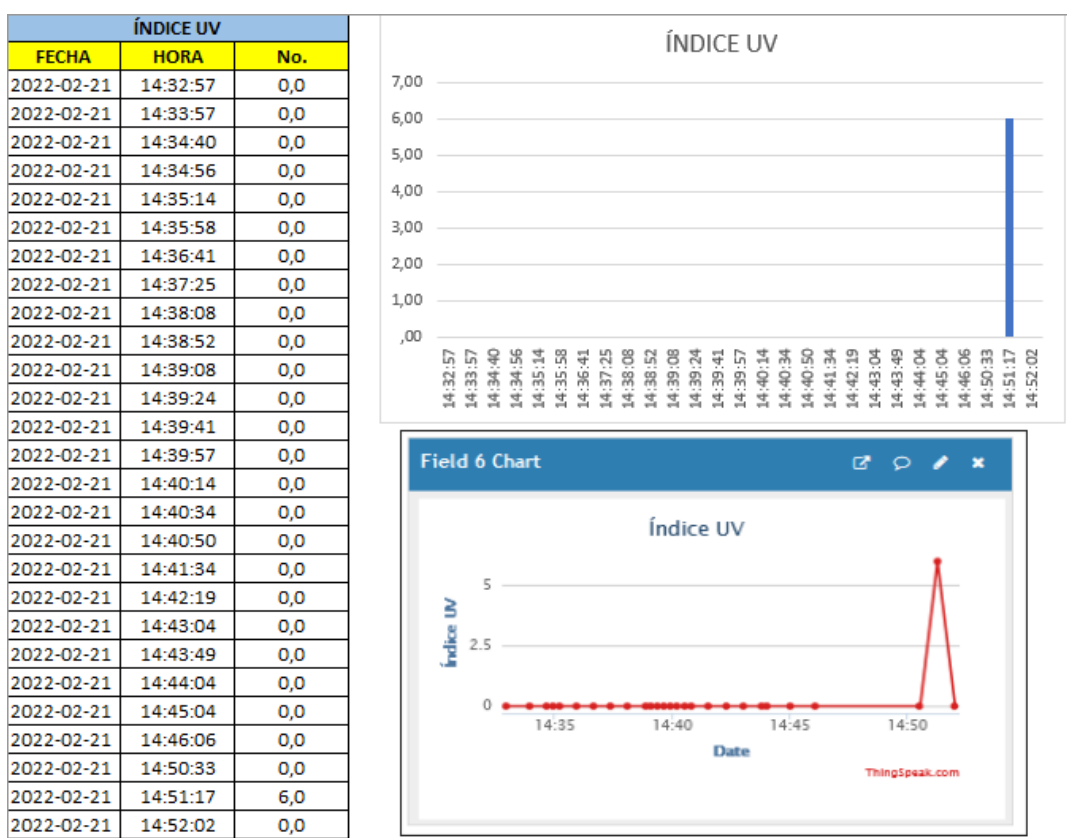


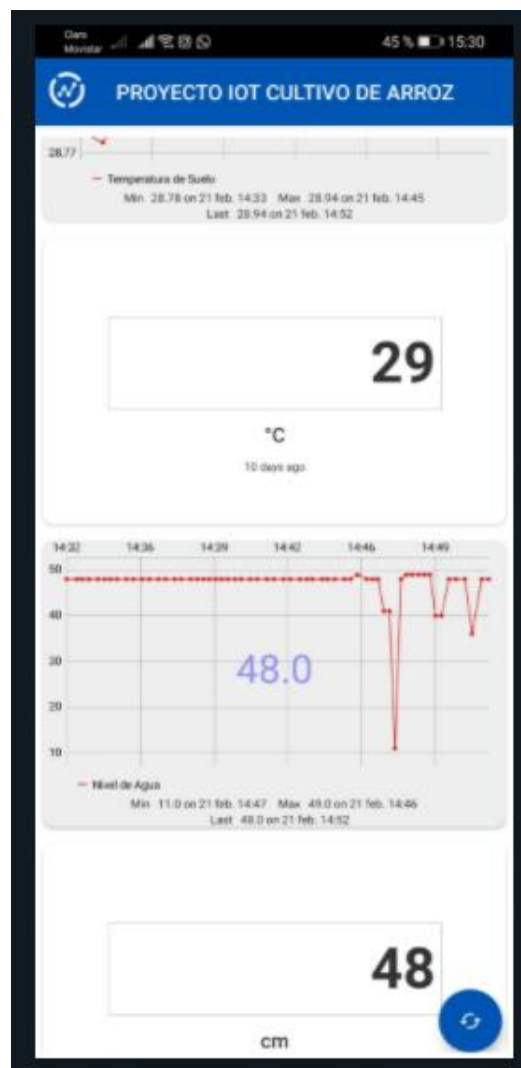
Figura 83 Datos de índice UV.





## 5.5. Visualización de resultados por medio de aplicación móvil ThingView

Por medio de la aplicación para teléfonos móviles llamada ThingView como se muestra en la Figura 85, también se puede visualizar los valores tomados por los sensores. El procedimiento es sencillo, se descarga la aplicación al dispositivo, luego se introduce el número del canal y automáticamente se visualizan las gráficas con el comportamiento de los parámetros ambientales que se están monitoreando.



**Figura 85** Aplicación móvil ThingView.

## 6. Conclusiones

- Se cumple con el diseño de un sistema de red de sensores de temperatura ambiente, humedad relativa, humedad y temperatura del suelo, nivel de agua, y radiación UV para el dispositivo IoT de monitoreo de parámetros ambientales para el cultivo de arroz.
- Se realiza la implementación de servicios en la nube para el almacenamiento y monitoreo web de la información obtenida por los sensores mediante la plataforma de ThingSpeak, además se lleva a cabo el control mediante una aplicación para dispositivos móviles de todos los datos obtenidos por los sensores instalados en los cultivos de arroz.
- Se cumple con la evaluación y análisis de los diferentes datos obtenidos de la red de sensores mediante un sistema vía mails de ThingSpeak y tweets de Twitter para alertar de acuerdo con los umbrales de cada sensor. Se realiza las configuraciones en las APP de ThingSpeak para el uso de estas funciones especiales.
- Se realiza el control vía web mediante la aplicación ThingHTTP de ThingSpeak el encendido y apagado automático del motor DC para el riego de agua del cultivo de arroz en maqueta de acuerdo con los rangos establecidos del nivel de agua, estos valores de rangos e intervalos serán definidos por expertos en el área del cultivo de arroz. Se utiliza datos característicos de cultivos para la configuración de los umbrales.

## 7. Recomendaciones

- Se recomienda realizar pruebas en sitio de cultivo de arroz por lo menos durante 6 meses para evaluar el completo funcionamiento del prototipo IoT.
- Se debe realizar la instalación del prototipo en zonas del cultivo donde la tierra esté lo más nivelada posible, para que la toma de medidas del sensor del nivel de agua sean lo más precisas posibles, además de cubrir la placa principal con un protector transparente para evitar que haya daños en los sensores de temperatura y humedad ambiental como del sensor de rayos UV.
- Se recomienda adquirir y suscribirse a una cuenta de pago de la plataforma de ThingSpeak para obtener la lectura de los datos de forma rápida y eficiente, por un año calendario, ya que las cuentas gratuitas solo permiten la lectura de datos de manera limitada debido a que estos datos se actualizan en aproximadamente 40 segundos.
- Se debe instalar una bomba de agua de tipo industrial para el riego en un sembrío de cultivo real, para el caso del prototipo IoT se utilizó una bomba DC adecuada para la maqueta.

## Bibliografía

- [1] Expreso.ec, “El arroz levanta un récord de consumo,” 2021. [Online]. Available: <https://www.expreso.ec/actualidad/economia/coronavirus-ecuador-arroz-levanta-record-consumo-12079.html>. [Accessed: 04-May-2021].
- [2] El Telégrafo, “El Telégrafo - El arroz mejora con ayuda de la tecnificación,” 2021. [Online]. Available: <https://www.eltelegrafo.com.ec/noticias/economia/4/arroz-tecnificacion-ecuador>. [Accessed: 04-May-2021].
- [3] INiap, “Arroz,” 2021. .
- [4] Jose Manuel Fernández, “Agricultura inteligente (Parte 3): IoT aplicado a la agricultura, el poder de la comunicación,” 2019. .
- [5] El Productor, “Sector arrocero de Santa Lucia indica que cosecha fuerte de arroz se verá en mayo | Noticias Agropecuarias,” 2021. .
- [6] INiap, “Cosecha y Poscosecha,” 2021. .
- [7] M. R. Pérez, ; Mendoza, Miguel A, & Suarez, and Marco J, “Paradigma IoT: desde su conceptualización hacia su aplicación en la agricultura IoT paradigm: from its concept towards its application in agriculture Contenido.”
- [8] Interxion, “IoT para la industria: las fábricas aceleran su transformación digital | Interxion,” 2020. [Online]. Available: <https://www.interxion.com/es/blogs/2018/08/iot-para-la-industria--las-fabricas-aceleran-su-transformacion-digital>. [Accessed: 13-Apr-2020].
- [9] V. Puranik, Sharmila, A. Ranjan, and A. Kumari, “Automation in Agriculture and IoT,” Proc. - 2019 4th Int. Conf. Internet Things Smart Innov. Usages, IoT-SIU 2019, Apr. 2019.
- [10] I. Mohanraj, K. Ashokumar, and J. Naren, “Field Monitoring and Automation Using IOT in Agriculture Domain,” Procedia Comput. Sci., vol. 93, pp. 931–939, 2016.
- [11] A. Kassem, M. Tamazin, and M. H. Aly, “A Context-Aware IoT-Based Smart Wearable Health Monitoring System,” 2021, pp. 1–6.
- [12] mouser, “ESP32-WROOM-32D(M113DH3200PH3Q0) Espressif Systems | Mouser Ecuador,” 2021. .
- [13] mouser, “ESP32-WROOM-32U(M113DH6400UH3Q0) Espressif Systems | Mouser Ecuador,” 2021. .
- [14] naylampmechatronics, “Sensor de temperatura y humedad relativa DHT22 (AM2302),” 2021. .

- [15] ThinkSpeak, "ThingSpeak para la agricultura inteligente - ThingSpeak IoT," 2021. [Online]. Available: [https://thingspeak.com/pages/smart\\_farming](https://thingspeak.com/pages/smart_farming). [Accessed: 04-May-2021].
- [16] H. M. Torres; J. F. Guillén; J. D. Mogrovejo, *Sistema de Monitoreo de Parámetros Agrícolas para la Supervisión de Cultivos de Manera Remota basado en Redes de Sensores Inalámbricos*. Azuay: Universidad del Azuay, 2019.
- [17] N. J. Melo; W. A. Valverde, *Implementación de un sistema de almacenamiento de la información, monitoreo y control aplicando el internet de las cosas, para la automatización de un invernadero*. Chimborazo: Universidad Nacional de Chimborazo, 2017.
- [18] D. A. Sánchez ; J. L. Honores, *Desarrollo e implementación de un módulo de agricultura de precisión para monitoreo y control de riego, integrado al sistema IOTMACH*. Machala: Universidad Técnica de Machala, 2016.

## Anexos

### Anexo 1: Codificación de los módulos ESP32.

A continuación, se muestra el código grabado en los módulos ESP32 de la placa principal y placa secundaria.

CODIFICACION DE LA PLACA ESP 32 PRINCIPAL CON TODOS LOS SENSORES.

// DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO IOT PARA EL MONITOREO DE PARÁMETROS AMBIENTALES APLICADOS AL CULTIVO DE ARROZ UTILIZANDO ESP32 Y THINGSPEAK.

// AUTORES: LEANDRO MOSQUERA - CESAR CEVALLOS

//LIBRERIAS

#include <WiFi.h>

#include <ThingSpeak.h>

#include <SimpleDHT.h> //DHT22 Sensor Temperatura y Humedad Ambiente.

#include <Arduino.h> // SHT10 Sensor Temperatura y Humedad del Suelo.

#include <SHT1x-ESP.h> // SHT10 Sensor Temperatura y Humedad del Suelo.

#include <SoftwareSerial.h> // US100 Sensor Ultrasónico.

#include <HTTPClient.h>

//CONSTANTES

const char\* ssid = "TP-TESIS"; // SSID WIFI

const char\* password = "tesis123"; // Password WIFI

unsigned long myChannelNumber = 1655896; // Thingspeak número de canal

const char \* myWriteAPIKey = "43ZTOSA7IPDCFC2I"; // ThingSpeak write API Key

WiFiClient espClient;

HTTPClient http;

String url;

String API = "43ZTOSA7IPDCFC2I";

```

String FieldNo = "5";

const int US100_TX = 16;
const int US100_RX = 17;

const int NIVEL_AGUA_1 = 3; // NIVEL DE AGUA máx.
const int TOPE = 50; // NIVEL DE AGUA TOPE

SoftwareSerial US100Serial(US100_RX, US100_TX);

unsigned int MSByteDist = 0;
unsigned int LSByteDist = 0;
unsigned int mmDist = 0;
unsigned int cmDist = 0;
unsigned int NewcmDist = 0;

//VARIABLES DE SENSORES*****

//TEMPERATURA      AMBIENTE      Y      HUMEDAD      RELATIVA      //
DHT22*****

int pinDHT22 = 04; // Puerto D0 entrada de datos
SimpleDHT22 dht22; // Modelo de sensor

//TEMPERATURA Y HUMEDAD DEL SUELO // SHT10 *****

// Specify data and clock connections and instantiate SHT1x object
#define dataPin 02
#define clockPin 15

// default to 5.0v boards, e.g. Arduino UNO
//SHT1x sht1x(dataPin, clockPin);

// if 3.3v board is used

```



```
SHT1x sht1x(dataPin, clockPin, SHT1x::Voltage::DC_3_3v);
```

```
// CONECTANDO A RED WIFI
```

```
void setup_wifi() {
```

```
  delay(10);
```

```
  Serial.println();
```

```
  Serial.print("Connecting to ");
```

```
  Serial.println(ssid);
```

```
  WiFi.mode(WIFI_STA);
```

```
  WiFi.begin(ssid, password);
```

```
  ThingSpeak.begin(espClient); // Inicia WIFI
```

```
  while (WiFi.status() != WL_CONNECTED) {
```

```
    delay(500);
```

```
    Serial.print(".");
```

```
  }
```

```
  randomSeed(micros());
```

```
  Serial.println("");
```

```
  Serial.println("WiFi conectado");
```

```
  Serial.println("IP address: ");
```

```
  Serial.println(WiFi.localIP());
```

```
}
```

```
void setup(){
```

```
  Serial.begin(115200); // Velocidad del puerto serie
```

```
  setup_wifi();
```

```
  pinMode(pinDHT22,INPUT);// DTH_22
```

```

US100Serial.begin(9600); // US100

}

int httpcode;
int stepvalue;

void SendGETRequest(int data)
{

    url = "http://api.thingspeak.com/update?api_key=";
    url = url + API;
    url = url + "&field";
    url = url + FieldNo;
    url = url + "=";
    url = url + data;
    http.begin(espClient,url);
    Serial.println("Esperando respuesta");
    Serial.println(" ");
    httpcode = http.GET();
    if (httpcode > 0)
    {
        Serial.println ("Datos enviados satisfactoriamente");
        Serial.println(" ");
    }
    else
    {
        Serial.println ("Error en el envio");
        Serial.println(" ");
    }
    http.end ();
}

```

```

void loop(){

//TEMPERATURA      AMBIENTE      Y      HUMEDAD      RELATIVA      //
DHT22*****

float t = 0;
float h = 0;
if (dht22.read2(pinDHT22, &t, &h, NULL)){
  Serial.println("Error en la lectura.");
  return;
}
//Muestra por el puerto serie las medidas obtenidas
Serial.println("*****");
Serial.println("TEMPERATURA AMBIENTE Y HUMEDAD RELATIVA");
Serial.println("*****");
Serial.print("Humedad relativa: ");
Serial.print((float)h);
Serial.println(" %");
Serial.print("Temperatura Ambiente: ");
Serial.print((float)t);
Serial.println(" *C");
Serial.println("      ");

//SENSOR ULTRAVIOLETA // UVM30A*****

float sensorVoltage;
float sensorValue;
unsigned int UVIndex = 0;

sensorValue = analogRead(34);
sensorVoltage = sensorValue/4095*3.3;
UVIndex = sensorVoltage*10;

Serial.println("*****");

```

```

Serial.println("RAYOS ULTRAVIOLETA");
Serial.println("*****");
Serial.print("sensor reading = ");
Serial.println(sensorValue);
Serial.print("sensor voltage = ");
Serial.print(sensorVoltage); // tiene que hacer un map de los valores leidos OUT: 0V to 1V (0
to 10 UV Index) para hallar el indice de luz ultravioleta.
Serial.println(" V");
Serial.print("UVIndex = ");
Serial.println(UVIndex);
Serial.println("          ");
delay(1000);

//SENSOR TEMPERATURA Y HUMEDAD DEL SUELO //SHT10*****

float temp_c;
float temp_f;
float humidity;

// Read values from the sensor
temp_c = sht1x.readTemperatureC();
temp_f = sht1x.readTemperatureF();
humidity = sht1x.readHumidity();

// Print the values to the serial port
Serial.println("*****");
Serial.println("TEMPERATURA Y HUMEDAD DE SUELO");
Serial.println("*****");
Serial.print("Temperatura del suelo: ");
Serial.print(temp_c, DEC);
Serial.print(" *C /");
Serial.print(temp_f, DEC);
Serial.println(" *F");
Serial.print("Humedad del suelo: ");

```

```

Serial.print(humidity);
Serial.println(" %");
Serial.println("          ");
delay(1000);

```

```
//SENSOR ULTRASÓNICO //US100*****
```

```

US100Serial.flush();
US100Serial.write(0x55);
delay(500);

```

```

if(US100Serial.available() >= 2){
  MSByteDist = US100Serial.read();
  LSByteDist = US100Serial.read();
  mmDist = MSByteDist * 256 + LSByteDist;
  cmDist = (mmDist)*(0.1);

```

```

if((mmDist > 1) && (mmDist < 10000)){
  if(cmDist <= 50){
    NewcmDist = TOPE - (cmDist);
    Serial.println("*****");
    Serial.println("*NIVEL DEL AGUA*");
    Serial.println("*****");
    Serial.print("Distancia: ");
    Serial.print(NewcmDist);
    Serial.println(" cm");
    Serial.println("          ");
  }

```

```

else {
  Serial.println("*****");
  Serial.println("*NIVEL DEL AGUA*");
  Serial.println("*****");
  Serial.print("Distancia: ");
  Serial.print(" 0 ");

```

```

        Serial.println(" cm");
        Serial.println("          ");
    }
}
}

//ThingSpeak
//Carga los valores a enviar
ThingSpeak.setField(1, (float)t); // TEMPERATURA AMBIENTE
ThingSpeak.setField(2, (float)h); // HUMEDAD RELATIVA
ThingSpeak.setField(3, (float)humidity); //HUMEDAD DEL SUELO
ThingSpeak.setField(4, (float)temp_c); // TEMPERATURA DEL SUELO
ThingSpeak.setField(5, (int)NewcmDist); // NIVEL DEL AGUA
ThingSpeak.setField(6, (int)UVIndex); // RAYOS ULTRAVIOLETA

if (NewcmDist <= NIVEL_AGUA_1) {
Serial.println("SE ENCIENDE BOMBA");
Serial.println(" ");
ThingSpeak.setField(7, (float)HIGH); // ESTADO DE LA BOMBA DE AGUA ENCENDIDA

} else if (NewcmDist > NIVEL_AGUA_1) {
Serial.println("SE APAGA LA BOMBA");
Serial.println(" ");
ThingSpeak.setField(7, (float)LOW); // ESTADO DE LA BOMBA DE AGUA APAGADA

}

//Escribe todos los campos a la vez.
ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
Serial.println("***Datos enviados a ThingSpeak***");
Serial.println("          ");
delay(10000);

stepvalue = NewcmDist;

```

```
    SendGETRequest (stepvalue);  
}
```

CODIFICACION DE LA PLACA ESP 32 SECUNDARIA CONECTADA A LA BOMBA DE AGUA.

// DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO IOT PARA EL MONITOREO DE PARÁMETROS AMBIENTALES APLICADOS AL CULTIVO DE ARROZ UTILIZANDO ESP32 Y THINGSPEAK.

// AUTORES: LEANDRO MOSQUERA - CESAR CEVALLOS

```
#include <WiFi.h>
```

```
#include <HTTPClient.h>
```

```
#include <WiFiClient.h>
```

```
char ssid[] = "TP-TESIS"; // Network SSID.
```

```
char pass[] = "tesis123"; // Network password.
```

```
WiFiClient espClient;
```

```
HTTPClient http;
```

```
int httpCode;
```

```
String response;
```

```
String begin_string;
```

```
String end_string;
```

```
String result;
```

```
String Command_string;
```

```
#define PUMP 13
```

```
// Se define el pin de encendido de la bomba.
```

```
void setup_wifi() {
```

```
    //delay(10);
```

```
    Serial.println();
```

```

Serial.println(ssid);

WiFi.mode(WIFI_STA);
WiFi.begin(ssid, pass);
Serial.print("Connecting to ");
// ThingSpeak.begin(espClient); // Inicia WIFI

while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
}
randomSeed(micros());
Serial.println("");
Serial.println("WiFi conectado");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
Serial.println(WiFi.macAddress());
}

void setup()
{
    Serial.begin(115200); // Inicia la velocidad de la lectura.
    pinMode(PUMP, OUTPUT);

    digitalWrite(PUMP, HIGH); // Inicia apagada la bomba

    setup_wifi();
}

// En el main loop, empieza estableciendo una conexión a la red local WIFI. Crea un POST
message con los parámetros correctos, y luego envía el POST request, verifica el resultado,
y chequea el TalkBack command.

```



```

void loop()
{
  Command_string = GETCommand ();
  if (Command_string == "L1")// Lectura del nivel inferior, prende bomba
  {
    digitalWrite(PUMP, LOW);
  }
  if (Command_string == "L2")// Lectura del nivel superior, apaga la bomba
  {
    digitalWrite(PUMP, HIGH);
  }
  Serial.println(Command_string);
  delay (1000);
}

//Código que recibe la orden de la placa ESP32 principal.
String GETCommand ()
{

  http.begin(espClient,"http://api.thingspeak.com/talkbacks/44794/commands/26216083.json?
  api_key=APSQIAV8D16NAADS");
  httpCode = http.GET();
  if (httpCode > 0)
  {
    response = http.getString ();
    int index_start = response.indexOf("command_string\":\"") + 17;
    result = response.substring (index_start, index_start+2);
  }
  else { Serial.println ("Falla de conexion con el servidor"); }
  http.end ();
  return result;
}

```

## Anexo 2: Codificación en la plataforma ThingSpeak.

A continuación, se observa el código de MatLab donde se debe configurar el channelID, el Alert Api Key, y el umbral de temperatura.

```
% Store the channel ID
channelID = 1655896;

% Provide the ThingSpeak alerts API key. All alerts API keys start with TAK.
alertApiKey = 'TAKW18YVT6DZDF77MMCHG';

% Set the address for the HTTP call
alertUrl="https://api.thingspeak.com/alerts/send";

% webwrite uses weboptions to add required headers. Alerts needs a
ThingSpeak-Alerts-API-Key header.
options = weboptions("HeaderFields", ["ThingSpeak-Alerts-API-Key", alertApiKey
]);

% Set the email subject.
alertSubject = sprintf("Información de Temperatura Ambiente");

% Read the recent data.
TempData = thingSpeakRead(channelID,'NumDays',30,'Fields',1);

% Check to make sure the data was read correctly from the channel.
if isempty(TempData)
    alertBody= ' No hay datos para leer';
else
    % Calculate a 10% threshold value based on recent data.
    %span = max(TempData) - min(TempData);
    tempValue = 30;

    % Get the most recent point in the array of temp data.
    lastValue = TempData(end);

    % Set the outgoing message
    if (lastValue<tempValue)
        alertBody= ' Temperatura está bajando';
    elseif (lastValue>tempValue)
        alertBody= ' Temperatura está subiendo ';
    end
end
```

```
end

% Catch errors so the MATLAB code does not disable a TimeControl if it fails
try
    webwrite(alertUrl , "body", alertBody, "subject", alertSubject, options);
catch someException
    fprintf("Failed to send alert: %s\n", someException.message);
end
```

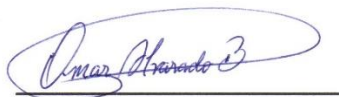
**Anexo 3: Certificado de control por parte del Ingeniero Agrónomo y certificado de registro de título.**

Guayaquil, 03 de marzo del 2022

**CERTIFICADO DE CONTROL**

Yo, **Neptalí Omar Alvarado Barzola** con cédula de identidad No. **0916290059**, Ingeniero Agrónomo de profesión, certifico que he asesorado a los tesisistas Leandro Mauricio Mosquera Meléndrez y César Danilo Cevallos Rojas, durante el desarrollo del proyecto de tesis **DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO IOT PARA EL MONITOREO DE PARÁMETROS AMBIENTALES APLICADOS AL CULTIVO DE ARROZ UTILIZANDO ESP32 Y THINGSPEAK**, proporcionando las respectivas directrices para la correcta ejecución de dicho proyecto, entre las cuales se incluyen los rangos adecuados para la medición de los parámetros ambientales para el cultivo del arroz.

Atentamente.



Neptalí Omar Alvarado Barzola  
C.I.: 0916290059

Quito, 06/05/2022

### CERTIFICADO DE REGISTRO DE TÍTULO

La Secretaría de Educación Superior, Ciencia, Tecnología e Innovación, SENESCYT, certifica que ALVARADO BARZOLA NEPTALI OMAR, con documento de identificación número 0916290059, registra en el Sistema Nacional de Información de la Educación Superior del Ecuador (SNIESE), la siguiente información:

Nombre: ALVARADO BARZOLA NEPTALI OMAR  
Número de documento de identificación: 0916290059  
Nacionalidad: Ecuador  
Género: MASCULINO

#### Título(s) de tercer nivel de grado

Número de registro	1006-2018-1974051
Institución de origen	UNIVERSIDAD DE GUAYAQUIL
Institución que reconoce	
Título	INGENIERO AGRONOMO
Tipo	Nacional
Fecha de registro	2018-06-18
Observaciones	