



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

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE MONITOREO DE
TEMPERATURA, HUMEDAD Y PRESIÓN CON COMUNICACIÓN VÍA RADIO E
INTERNET PARA MEJORAR LA PRODUCCIÓN AGRÍCOLA**

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

AUTORES:

WILLY JOSE SANDOYA MEDRANO
MARCO EMANUEL MENA LEDESMA

TUTOR:

ING. LENIN ESTUARDO CEVALLOS ROBALINO, PhD.

Guayaquil-Ecuador

2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Willy Jose Sandoya Medrano con documento de identificación N° 1206017905 y; Marco Emanuel Mena Ledesma con documento de identificación N° 0923104566; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 1 de septiembre del año 2021

Atentamente,



Firmado electrónicamente por:
**MARCO EMANUEL
MENA LEDESMA**

Marco Emanuel Mena Ledesma
0923104566

Willy Jose Sandoya Medrano
1206017905

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

Nosotros, Willy Jose Sandoya Medrano con documento de identificación No. 1206017905 y Marco Emanuel Mena Ledesma con documento de identificación No. 0923104566, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: diseño e implementación de un prototipo de monitoreo de temperatura, humedad y presión con comunicación vía radio e internet para mejorar la producción agrícola, el cual ha sido desarrollado para optar por el título de: Ingeniero electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 1 de septiembre del año 2021

Atentamente,



Firmado electrónicamente por:
**MARCO EMANUEL
MENA LEDESMA**

Marco Emanuel Mena Ledesma
0923104566

Willy Jose Sandoya Medrano
1206017905

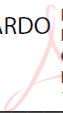
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Lenin Estuardo Cevallos Robalino con documento de identificación N° 0603274226, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: diseño e implementación de un prototipo de monitoreo de temperatura, humedad y presión con comunicación vía radio e internet para mejorar la producción agrícola, realizado por Marco Emanuel Mena Ledesma con documento de identificación N° 0923104566 y por Willy Jose Sandoya Medrano con documento de identificación N° 1206017905, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 1 de septiembre del año 2021

Atentamente,

LENIN ESTUARDO
CEVALLOS
ROBALINO



Firmado digitalmente por
LENIN ESTUARDO
CEVALLOS ROBALINO
Fecha: 2021.10.01
17:44:23 +02'00'

Lenin Estuardo Cevallos Robalino
0603274226

Dedicatoria

Dedico este proyecto a mi madre Mónica Ledesma y a mi padre Marco Mena, por siempre apoyarme en las situaciones más difíciles, a mis hermanas que siempre están cuando más las necesito.

Marco Emanuel Mena Ledesma

Dedico este proyecto a mi familia, el apoyo incondicional que me brindan siempre me da las energías necesarias para continuar este maravilloso viaje al que llamamos vida.

Willy Sandoya

Agradecimiento

Agradezco a la Universidad Politécnica Salesiana y a sus docentes por impartir conocimientos importantes para nuestro desarrollo como ingenieros electrónicos.

Agradecimientos especiales:

- Pablo Luis Sánchez Stahlschmidt, instructor de UdeMy, por impartir los conocimientos necesarios para completar el proyecto.
- Ing. Jose Walter Sandoya Villafuerte M.Sc, por colaborar en el desarrollo del código y la construcción del prototipo.
- Ing. Alexis Steven Cedeño Hernandez, por colaborar en el desarrollo de la página web.
- Ing. Denys Alava, por colaborar en el desarrollo de la base de datos.

Willy Sandoya

RESUMEN

El proyecto propuesto es diseñar e implementar un prototipo de monitoreo de temperatura, humedad y presión con comunicación vía radio e internet para mejorar la producción agrícola el cual se recopila información con respecto a los sensores que son compatibles con la unidad de procesamiento principal de un microcontrolador.

El prototipo mide tres variables fundamentales en el cuidado agrícola (humedad, temperatura y presión atmosférica), el cual se usa como una herramienta para el profesional o el usuario que trabaja en las actividades agrícolas, donde siempre y cuando se mantengan las condiciones adecuadas en el terreno se consigue un mejor desarrollo del cultivo en su etapa joven y su adultez que por resultados se obtiene una mejor producción.

El prototipo consiste en un set de 2 microcontroladores ESP32 LoRa que van a trabajar en conjunto, el microcontrolador transmisor (Tx) estará en el campo de trabajo mostrando los datos obtenidos en la pantalla incorporada mientras que el microcontrolador receptor (Rx) recibirá la información obtenida reflejando la misma en la pantalla incorporada para verificar los datos sin la necesidad de estar en el campo de trabajo, además se tiene la posibilidad de ver los datos en cualquier parte del mundo porque los datos estarán reflejados también en la página web y además en el registro de la base de datos. Todo esto se puede lograr gracias a que los microcontroladores poseen módulos de Wi-Fi y radiofrecuencia.

El protocolo MQTT se encarga de mantener una sesión entre el usuario y el dispositivo mediante credenciales para que el usuario pueda visualizar por una interfaz gráfica los datos obtenidos por los sensores, el usuario al utilizar este servicio sólo va a poder recibir la información a la que está suscrito, en este caso los datos de temperatura, humedad y presión.

Adicional, se implementó al prototipo el servicio de una base de datos, por medio del software Heidi SQL, al cual se debe acceder a través de credenciales que darán comunicación con el servidor y se puede subir dicha información del servidor a la base de datos, el presente proyecto fue diseñado de tal manera que la única inversión sea en el hardware, tanto

dispositivos como sensores, sin embargo, el costo de inversión en software es totalmente gratuito.

ABSTRACT

The proposed project is to design and implement a temperature, humidity and pressure monitoring prototype with communication via radio and internet to improve agricultural production which information it's collected regarding sensors that are compatible with the main processing unit of a microcontroller.

The prototype measures three fundamental variables in agricultural care, which is used as a tool for the professional or the user who works in agricultural activities, where as long as the appropriate conditions are maintained in the field, a better development of the crop is achieved in its young stage and adulthood, which results in a better production.

The prototype consists of a set of two ESP32 LoRa microcontrollers that will work together, the transmitter microcontroller (Tx) will be in the work field showing the obtained data on the built-in screen while the receiving microcontroller (Rx) will receive the obtained information reflecting the same on the built-in screen to verify the data without the necessity to be in the work field, in addition you have the possibility of viewing the data anywhere in the world because the data will also be reflected on the web page and also in the database record. All this can be achieved thanks to the fact that the microcontrollers have Wi-Fi and radio frequency modules.

The MQTT protocol is responsible for maintaining a session between the user and the device through credentials so that the user can view the data obtained by the sensors through a graphical interface, the user by using this service, will only be able to receive the information to which he is subscribed, in this case the data of temperature, humidity and pressure.

Additionally, a database service was implemented to the prototype, via Heidi SQL software, which must be accessed through credentials that will give us communication with the server and that information can be uploaded from the server to the database, this project was designed in such a way that the only investment is in the hardware, both devices and sensors, however, the investment in software is completely free.

Índice general

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN	1
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA	2
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN	3
Dedicatoria	4
Agradecimiento	5
RESUMEN	6
Índice general.....	10
Índice de figuras.....	13
Índice de tablas	14
I. INTRODUCCIÓN	15
Problema.....	16
Antecedentes.....	17
Delimitación	18
Explicación del problema	19
Grupo objetivo	19
Objetivo general	19
Objetivo específico	19
II. Marco Teórico	20
1 Factores Influyentes en la Agricultura Moderna	20
1.1 Factores Climáticos	20
1.2 Clima y Actividad Agraria.....	20
1.3 Productos Cultivados en la región Costa	21
1.4 Productos Cultivados en la región Sierra	21
1.5 Productos Cultivados en la Región Amazónica.....	21
1.6 Ecosistema Tropical (Costa).....	21
1.7 Ecosistema Andino (Sierra)	22

1.8	Ecosistema Amazónico (Amazonía)	22
1.9	Agricultura Inteligente	23
1.10	Automatización de la Agricultura	24
2	Microcontroladores y Sensores	24
2.1	Microcontroladores	24
2.2	Aplicaciones de Microcontroladores	24
2.3	Sensores	25
2.4	Algunos Tipos de Sensores	25
2.5	Aplicaciones de Sensores	25
3	Protocolos de comunicación (Microcontroladores)	26
3.1	Protocolos de Comunicación	26
4	Protocolos de Comunicación (Redes)	27
4.1	Protocolos de Comunicación	27
5	Comunicación MQTT	30
5.1	Protocolo MQTT	30
5.2	MQTT y sensores	33
6	ESP32: Fundamentos y alcances	33
6.1	ESP32	33
6.2	ESP32 LoRa	35
6.3	Bases de Datos	35
6.4	HeidiSQL	35
6.5	EMQX	36
	Fundamentación teórica	36
	III. Marco metodológico	37
	Estructura del Proyecto	38
7	Comunicación entre Sensores & ESP32	39
8	ESP32 & Servidor MQTT	39
9	Registro en Base de Datos	39

Funcionamiento del Proyecto.....	40
10 Diseño del circuito	40
Página Web y Comunicación Online.....	58
21 Vesta Control Panel.....	59
22 Putty	59
23 Plantilla para Desarrollo Web.....	59
24 Funcionamiento de la Base de Datos	65
IV. Resultados.....	68
Experimento 1: Obtención de datos entre 2 ambientes.	68
Experimento 2: Pruebas en campo.	75
7 Temperatura	79
8 Humedad	82
9 Presión atmosférica.....	84
Análisis.....	87
V. Conclusiones	88
VI. Recomendaciones	88
Referencias bibliográficas	89
Bibliography	89
Anexos	94
25 Diseño placa.....	94
26 Placa terminada.....	96
27 Placas con dispositivos.....	99
28 Proyecto en Funcionamiento	100
29 Código Tx	101
30 Código Rx.....	107

Índice de figuras

Figura 1. Broker MQTT.....	32
Figura 2. Tópicos MQTT.....	32
Figura 3. Comunicación Broker MQTT	34
Figura 4. diagrama de pines para Heltec LoRa Wi-Fi (V2).....	44
Figura 5. Diagrama de pines para Heltec LoRa Wi-Fi (V2)	44
Figura 6. Diseño del circuito para el sensor BME 280.	47
Figura 7. Diseño del circuito para el sensor DS18B20.....	48
Figura 8. Diseño del circuito completo.	48
Figura 9. Librerías incluidas en la codificación del transmisor (Tx).....	51
Figura 10. Codificación para el ESP32 Transmisor (Tx).....	51
Figura 11. Codificación del transmisor (Tx), llamado de librerías.....	51
Figura 12. Codificación del transmisor (Tx), Función para identificar BME280.....	52
Figura 13. Codificación del transmisor (Tx), Función para obtener lecturas BME280	52
Figura 14. Codificación del transmisor (Tx), Llamado de función de identificación BME280. 53	
Figura 15. Codificación del transmisor (Tx), Llamado de función obtener lecturas DS18b20 53	
Figura 16. Codificación del transmisor (Tx), Llamado de función “getReadings” BME280	53
Figura 17. Codificación del transmisor (Tx), Solicitar lecturas del sensor “getReadings” DS18b20	53
Figura 18. Codificación del transmisor (Tx), Conversión de datos a cadena de caracteres DS18b20	53
Figura 19. Codificación del transmisor (Tx), Creación del paquete para comunicación LoRa	54
Figura 20. Librerías incluidas en la codificación del Recibidor (Rx)	55
Figura 21. Credenciales para dar acceso al Recibidor (Rx) a una red WiFi y al servidor MQTT.....	56
Figura 22. Identificación espClient para el acceso del recibidor (Rx)	56
Figura 23. Función” LoRaData” para el recibidor (Rx).....	57
Figura 24. Función “cbk” para el recibidor (Rx)	57
Figura 25. Función “cbk” para el recibidor (Rx)	58
Figura 26. Función “setup_wifi” para el recibidor (Rx).....	59
Figura 27. Temas de Flatkit, (Flatkit, 2020).....	61
Figura 28. Demos de Flatkit, (Flatkit, 2020).	62

Figura 29. Página principal en la web del Proyecto, (Flatkit, 2020).	62
Figura 30. Página Login del proyecto, (Flatkit, 2020).	63
Figura 31. Página Login Introducción de credenciales, (Flatkit, 2020).	63
Figura 32. Página Dashboar, (Flatkit, 2020).	64
Figura 33. Página Dashboard, (Flatkit, 2020).	64
Figura 34. Página Dashboard, actualización de datos, (Flatkit, 2020).	65
Figura 35. Página Dashboard, estabilidad de datos, (Flatkit, 2020).	65
Figura 36. Página Dashboard, estabilidad de datos, (Flatkit, 2020).	66
Figura 37. Ventana HeidiSQL, acceso al servidor	67
Figura 38. Ventana HeidiSQL, ventana principal del servidor	67
Figura 39, Ventana HeidiSQL, tabla de datos.	68
Figura 40. Ventana PuTTY, registro de datos.	68
Figura 41. Ventana HeidiSQL, registro en las tablas	69
Figura 42. Plantas de prueba	70
Figura 43. Sujeto 1 en el día 1	71
Figura 44. Sujeto 2 en el día 1	71
Figura 45. Sujeto 1 en el día 2	72
Figura 46. Sujeto 2 en el día 2	72
Figura 47. Sujeto 1 en el día 3	73
Figura 48. Sujeto 2 en el día 3	73
Figura 49. Sujeto 1 en el día 4	73
Figura 50. Sujeto 2 en el día 4	74
Figura 51. Sujeto 1 en el día 5	74
Figura 52. Sujeto 2 en el día 5	75
Figura 53. Monitoreo de la siembra de Arroz	76
Figura 54. Dueño de hacienda “La Lucha”	77
Figura 55. Cultivos de cacao	77
Figura 56. Receptor mostrando en pantalla las variables del transmisor en campo	78
Figura 57. Temperatura Siembra de Arroz Día 1.	80
Figura 58. Temperatura Siembra de Cacao Día 1	81
Figura 59. Temperatura Siembra de Arroz Día 2	81
Figura 60. Temperatura de Cacao Día 2	82
Figura 61. Humedad Siembra de Arroz Día 1	83

Figura 62. Humedad Siembra de Arroz Día 2.....	83
Figura 63. Humedad Siembra de Cacao Día 2.....	84
Figura 64. Humedad Siembra de Cacao Día 1.....	84
Figura 65. Presion Atmosférica Siembra de Arroz Día 1	85
Figura 66. Presión Atmosférica Siembra de Arroz Día 2	85
Figura 67. Presión Atmosférica Siembra de Cacao Día 1	86
Figura 68. Presión Atmosférica Siembra de Cacao Día 2	86
Figura 69, Placa tx.....	95
Figura 70, Placa Tx conductors.....	95
Figura 71, Placa Rx.....	96
Figura 72, Placa Rx conductores	97
Figura 73, Placa terminada	97
Figura 74, placa terminada tx	98
Figura 75, Parte posterior placa tx.....	98
Figura 76, Placa rx terminada.....	99
Figura 77, parte posterior placa rx.....	100
Figura 78, Placa tx con microcontrolador y sensores.....	100
Figura 79, Placa rx con microcontrolador	101
Figura 80, Placa tx en funcionamiento	102
Figura 81, Placa rx en funcionamiento.....	102

Índice de tablas

Tabla 1. Características eléctricas del controlador, (Heltec, 2020).	46
Tabla 2. Especificaciones de conexión adjunta a la figura 4.	47
Tabla 3. Especificaciones de conexión adjunta a la figura 5.	48
Tabla 4. Especificaciones de conexión adjunta a la figura 6.	49
Tabla 5. Librerías para la codificación del transmisor (Tx).....	50
Tabla 6. Librerías para la codificación del Recibidor (Rx)	55
Tabla 1. Efectos de la temperatura sobre el crecimiento y desarrollo del arroz.....	78

I. INTRODUCCIÓN

La productividad agrícola es muy importante ya que representa el 8% del PBI del país, de ella el 65% de la población en Asia & África y el 35% de Hispano América obtienen los alimentos que se consumen, además de otras fuentes como la pesca, la ganadería, etc. Se debe considerar que la calidad de los productos influye tanto en la alimentación de las personas como en su salud, esta calidad depende de varios factores ambientales entre los cuales tenemos 3 que son los que se van a considerar: temperatura, humedad, y presión, (Latham, 2021), (Andrade, 2021).

El prototipo se encarga de medir las variables para luego subirlos a una página web para luego ser guardados en una base de datos, para lograr esto se comunican los módulos ESP32 a través de tecnología LoRa, la cual utiliza radiofrecuencia, y finalmente se utiliza tecnología Wifi-incorporada en los módulos para que se pueda subir la información a internet.

A través del protocolo MQTT podremos visualizar dicho contenido en la página web en tiempo real ya que los sensores estarán en constante funcionamiento para poder visualizar y analizar los cambios que se den con mayor precisión, ya con el acceso a la información a través de la base de datos se pueden decidir qué acciones tomar para mitigar los aspectos que estén perjudicando la producción.

Problema

En el sector agrícola se tiende a sufrir de muchas pérdidas por falta de supervisión en los terrenos ya que los volúmenes de los cultivos pueden ser grandes, hay variables que no se pueden ver a simple vista, el terreno no estaba en condiciones locales o la temporada no era la adecuada para cultivar específicos productos.

El monitoreo de la temperatura, humedad y presión van a servir para poder mejorar la productividad, gracias a la adquisición de los datos se podrá predecir o determinar si hace falta aplicar cambios respectivos lo cual afectará de forma positiva al terreno. Por lo que se requiere de un dispositivo capaz de no solo monitorear sino también almacenar y mostrar la información donde y cuando se requiera, debido a que este tipo de herramientas con las características descritas no siempre están disponibles en el mercado local.

El internet de las cosas (IoT) ofrece la facilidad de acceder a la información sin la necesidad de estar presente en el campo de trabajo y no siempre es aprovechada ni utilizada por la falta de conocimiento técnico.

El microcontrolador que se desea implementar posee la particularidad de que es capaz de comunicarse a través de tecnología LoRa (vía radio) y Wi-Fi lo cual mantendrá comunicado al prototipo de monitoreo tomando la lectura de los sensores para luego subir dicha información al portal web del usuario y actualizar el dashboard.

Antecedentes

Se está sufriendo de pérdidas en los cultivos por falta de supervisión y no tener lecturas claras y medibles, donde si no se toman las respectivas medidas correspondientes se continúa teniendo pérdidas. Por este motivo se desarrolla una herramienta para los ingenieros agrónomos o usuarios dedicados a la agricultura.

Para temperaturas ambientales de hasta 40°C las plantas/los cultivos adaptados a altas temperaturas abastecidas adecuadamente de productos agroquímicos y agua podrán seguir transpirando y continuar su ciclo sin complicaciones, tales como: la lechuga, la acelga, repollo, entre otros, pasando los 40°C puede provocar que se marchite parte de la planta/cultivo o que incluso este muera (FAO, 2020).

Se considera de referencia los cultivos de invernadero, por ejemplo, tomate, pepino, pimiento, etc, cuando se tiene grandes porcentajes de humedad, se puede aprovechar especialmente al inicio del ciclo de vida de la planta donde se requiere entre 50-85% de humedad, y durante la etapa final de crecimiento ir disminuyendo la humedad a la que está expuesta hasta un porcentaje adecuado el cual sería de 60% como máximo, (Plagron, 2020), (Nutricontrol, 2020).

La poca presión atmosférica la planta no va a poder crecer rápido ni desarrollarse porque no va a tener suficiente intercambio de gases, las plantas están expuestas a 101kPa en la atmósfera terrestre, las plantas jóvenes necesitan estar expuestas al oxígeno entre 15-20kPa, estos son los requisitos mínimos de presión atmosférica (Geniolandia, 2020).

Delimitación

Académica: Orientado a la ingeniería electrónica, industrial, telecomunicaciones y a la ingeniería en agronomía. Se desarrollarán procesos que necesitan de conceptos básicos de redes, programación, electrónica y procesos industriales.

Espacial: Este proyecto se realiza en el cantón Baba, Recinto: Arenal, Hacienda: La Lucha.

Temporal: Este proyecto se realiza en los meses de octubre 2020 hasta septiembre 2021.

Explicación del problema

En base a la experimentación en la hacienda “La Lucha” se pudo identificar lo siguiente, la experiencia del trabajador en el campo logra buenos resultados ya que se tiene el conocimiento empírico que resulta de la prueba y error, pero si complementamos la experiencia con el prototipo se puede identificar de una manera más precisa los factores que están perjudicando al terreno. La falta de supervisión en los terrenos con piso lodoso, arenoso, etc, es muy complicada, incluso si no se detecta un aumento en la humedad, temperatura y presión se pueden sufrir pérdidas debido a distintos aspectos tales como: enfermedades en las plantas, deshidratación, altura, calor o cualquier otro aspecto negativo producido por no tomar medidas a tiempo que dificulta el crecimiento y desarrollo de los cultivos.

Grupo objetivo

Personas que posean terrenos dedicados a la agricultura, además investigadores que desean realizar estudios sobre el tema y para los estudiantes de la Universidad Politécnica Salesiana para que puedan aplicar sus conocimientos en laboratorios a través de las prácticas.

Objetivo general

Diseñar e implementar un dispositivo de monitoreo de temperatura, humedad y presión con comunicación vía radio e internet para mejorar la producción agrícola.

Objetivo específico

- Medir la temperatura, humedad y presión usando sensores y microcontroladores en un determinado terreno.
- Registrar las medidas obtenidas de los sensores en la base de datos.
- Mostrar a los usuarios la información histórica y actual del dispositivo a través de una página web.
- Posibilitar que otros profesionales realicen cálculos de los datos almacenados y descargados de la página web.

II. Marco Teórico

1 Factores Influyentes en la Agricultura Moderna

1.1 Factores Climáticos

Son aquellos que determinan el tipo de clima que existe en un lugar determinado, entre estos factores tenemos: Temperatura, humedad, presión atmosférica, viento y nubosidad, (S. Quiroga-Gómez, 2020).

1.2 Clima y Actividad Agraria

La actividad agraria depende mucho del clima, es determinante conocer las condiciones climáticas para poder trabajar el suelo y determinar su viabilidad. Los agricultores no pueden manipular el clima ni predecir con precisión las condiciones a la que está expuesto el terreno lo cual dificulta sus actividades y se enfrentan a la posibilidad de poca productividad, (Navarra, 2020).

1.3 Productos Cultivados en la región Costa

Algunos ejemplos de los productos que se cultivan en la Costa, (Elyex, 2020), (Delcorp, 2020):

Palma aceitera. banano, caña de azúcar, mango, café, sandía, arroz, papaya, maíz, cacao.

1.4 Productos Cultivados en la región Sierra

Algunos ejemplos de los productos que se cultivan en la Sierra, (Lifeder, 2020):

Papa, cebada, fresa, cebolla, zanahoria, manzana, café, quinua, trigo, haba.

1.5 Productos Cultivados en la Región Amazónica

Algunos ejemplos de los productos que se cultivan en la Amazonía, (Lifeder, 2021), (Delamazonas, 2021) :

Yuca, guaraná, pitahaya, lúcuma, guayusa, acaí, papa china, camu-camu, palma africana, naranjilla.

1.6 Ecosistema Tropical (Costa)

Los ecosistemas tropicales se caracterizan por tener temperaturas “altas” (>18 °C), lo cual produce gran evaporación de agua y estos estados se mantienen durante gran parte del año. Estas zonas son idóneas para el crecimiento de vegetación y cultivos (Características, 2021), (Ecosistema, 2021), (Ecología Verde, 2021).

1.7 Ecosistema Andino (Sierra)

Ecosistemas caracterizados por tener temperaturas “bajas” (7 °C & 21 °C), pero esto varía dependiendo de la ubicación:

Temperado seco: Zonas de hasta 1200 m de altura (20 °C & 25 °C.).

Temperado semi-húmedo: Zonas de entre 2500 y 3200 m de altura (12 °C & 20 °C).

Húmedo de alta montaña: Zonas de entre 2500 y 3200 m de altura (4 °C & 8 °C).

Glacial: Zonas de 4650 m de altura mínima (<0 °C).

En estas zonas se han adaptado varios tipos de vegetación, podemos resaltar ejemplares como las plantas medicinales, (Sierra Ecuador, 2021), (M. Castro, 2021).

1.8 Ecosistema Amazónico (Amazonía)

Ecosistemas muy “húmedos” con temperaturas que tienden a (26 °C & 40 °C), en estas zonas existe la mayor variedad de vegetación:

Vegetación de Tierra: Encontramos palmeras y el castaño de pará.

Valles: Especies semejantes a las de la categoría anterior.

Pantano: Encontramos especies como musgos, arbustos, enredaderas.

En estas zonas también podemos encontrar plantas medicinales y también gran variedad de plantas acuáticas y muchos tipos de flores. El suelo amazónico no es muy fértil, (Gtush, 2021).

Después de conocer todos estos aspectos presentes en Ecuador, viene la interrogante:

¿Es posible conocer los factores climáticos que afectan tanto directa como indirectamente al terreno destinado para la agricultura?

Con la necesidad de mejorar los terrenos y la productividad nacen las diferentes herramientas para adaptar a los terrenos a estos impredecibles cambios, la evolución de la actividad agrícola se vuelve más inteligente.

1.9 Agricultura Inteligente

Con las nuevas tecnologías podemos procesar información, obtener y hasta compartir datos obtenidos en el campo de trabajo. El uso de sensores, internet de las cosas (IoT), manipulación de datos, son algunos de los mejores ejemplos por los cuales podemos mejorar las condiciones del terreno, acondicionar mejor los cultivos, incluso obtener mejores resultados en la producción.

1.10 Automatización de la Agricultura

Gracias a las tecnologías de “última generación” se pueden realizar procesos de forma más eficiente, con el uso de algoritmos y análisis de datos tenemos resultados más precisos, más rápidos y con la posibilidad de señalar cuáles son los problemas que no permitan el desarrollo de los cultivos, incluso hacer todo esto de forma remota, (Agriculturers, 2021), (Agmoderna, 2021).

2 Microcontroladores y Sensores

2.1 Microcontroladores

Es un dispositivo electrónico programable capaz de realizar múltiples tareas, tiene integrado un CPU, memorias ROM, RAM, puertos INPUT & OUTPUT (Entrada & Salida), entradas analógicas, entradas digitales, salidas analógicas, salidas digitales, comunicación serial, comunicación I2C, entre otras especificaciones, (Sherlin, 2021), (Smelpro, 2021).

2.2 Aplicaciones de Microcontroladores

Las aplicaciones para los microcontroladores son muy diversas con posibilidades en crecimiento, actualmente los dispositivos digitales satisfacen muchas necesidades lo cual demanda que se hagan avances más rápidos en la tecnología y dispositivos electrónicos, los usuarios y consumidores cada vez son más exigentes y siempre están en busca de innovación.

2.2.1 Industrias con Microcontroladores

Las industrias que trabajan con microcontroladores son, (Microcontraladoressesv, 2021), (Hetpro, 2021):

Dispositivos médicos, Instrumentación, juguetes, sistemas de control, sistemas de medición, robótica, automatización, domótica, automotriz, IoT Internet of Things (Internet de las cosas).

2.3 Sensores

Son dispositivos que convierten magnitudes físicas o químicas en magnitudes eléctricas, es decir, detectan cambios en el campo para después captar esa información y la transforman en un impulso eléctrico que luego pasará a ser manipulado para la aplicación designada.

2.4 Algunos Tipos de Sensores

Los tipos de sensores pueden ser, (Ingeniería Mecafenix, 2021), (Dewesoft, 2021), (Recursostic, 2021):

De contacto, óptico, térmico, humedad, magnéticos, infrarrojos, micrófonos, aceleración.

2.5 Aplicaciones de Sensores

Las aplicaciones de sensores van a depender de la tarea que se requiera realizar, teniendo un sensor de temperatura se puede medir calor, con un sensor de proximidad se pueden medir distancias, con un sensor PIR se puede medir la radiación de luz infrarroja, etc. Gracias a esto se descubren más posibilidades e incluso llegar a acoplar diferentes sensores para cumplir múltiples tareas.

3 Protocolos de comunicación (Microcontroladores)

3.1 Protocolos de Comunicación

3.1.1 Comunicación en Paralelo

Transmite datos bit por bit a través de canales, es decir, un dato para cada hilo, para lograr esta comunicación se requiere el mismo número de entradas como de salidas.

3.1.2 Comunicación en Serie o Serial

Tiene la ventaja de enviar datos a través de un solo canal ("bus" de datos), el método más aplicado es la conexión USB. Gracias a esta comunicación se pueden enviar datos a mayores distancias utilizando menos recursos.

3.1.3 Comunicación I2C (Inter-Integrated Circuit)

Protocolo de comunicación síncrono que funciona mediante el método Maestro & Esclavo cuyo bus de datos va a ser transmitido por un solo canal de modo bidireccional, los dispositivos I2C utilizan 2 canales para transmitir la información y 1 canal para tierra.

3.1.4 Comunicación SPI

Protocolo de comunicación síncrono con la capacidad de enviar y recibir datos por el mismo canal, esta comunicación Maestro & Esclavo se caracteriza por tener un canal llamado SS (Slave Select) el cual se encarga de informar al esclavo cuando debe transmitir la información, (Tapia, 2021), (Concepción, 2021).

4 Protocolos de Comunicación (Redes)

4.1 TCP/IP

Establece la conexión entre los clientes y los servidores después de haber realizado varias solicitudes y se permita el acceso al servidor, para finalizar una conexión se debe enviar un mensaje de Salida o final para cerrar el socket destinado para establecer la conexión, en algunos casos puede existir una sesión abierta por parte del cliente, pero el socket para recibir esté cerrado esto implica que la información va a seguir enviándose y viceversa.

Debido a que TCP sirve a una gran cantidad de protocolos de la capa de aplicación, es fundamental que los datos lleguen correctamente al destinatario, sin errores, y, en orden. Si en la transmisión de los segmentos, se corrompieren o perdiesen, automáticamente el protocolo TCP inicia la retransmisión, sin intervención de la capa de aplicación. De esta manera, se garantiza que los datos llegan al destinatario sin errores, ya que este protocolo se encarga de solucionar cualquier tipo de problema (De Luz, 2021).

4.2 HTTP

Protocolo de comunicación entre cliente web y servidor web, el cliente solicita acceso al servidor a través del buscador para luego mostrar la página principal con la que podrá interactuar con la información de cabecera.

Para establecer una conexión HTTP se realiza desde un navegador con la respectiva solicitud HTTP y por supuesto se debe esperar una respuesta HTTP. Existen varios lenguajes para desarrollo web, pero la ventaja es que todos van a estar basados en HTTP, por lo tanto, las solicitudes se codifican para que la respuesta sea la requerida (López Jurado, 2021).

4.3 FTP

Protocolo de comunicación que permite compartir archivos entre computadoras o entre cliente web y servidor, del mismo modo se establece una conexión entre el cliente y el servidor y con dicha información se permite el acceso a la descarga de archivos.

4.4 SMTP

Protocolo de comunicación que se encarga del servicio de correo electrónico, para lograr dicha comunicación se establece conexión entre el cliente y el servidor SMTP a través del puerto 25 para permitir el servicio de mensajería entre los 2 puntos.

4.5 UDP

El protocolo UDP permite el envío de datagramas sin necesidad de establecer previamente una conexión, pero con la desventaja de que no se asegura de que la conexión sea completa, por ello se puede perder parte del paquete, tan solo es necesario tener abierto un socket en el destino para que acepte los datagramas del origen.

UDP es un protocolo no orientado a conexión, es decir, no ocurre como en TCP donde hay una fase de establecimiento de la conexión, aquí directamente se envían sin establecimiento previo.

Este protocolo no proporciona ningún tipo de control de flujo, si un equipo es más rápido que otro y envía información, es muy posible que se pierda información debido a que colapsará al más lento, y tendremos que proceder al reenvío de la información. Un detalle importante es que la gestión de reenvío de los datagramas la realiza la capa de transporte, ya que UDP es muy simple y no dispone de mecanismos de control de reenvío de datagramas por haberse perdido.

Se utiliza para sesiones que no requieren de verificación y si se desea obtener la información en tiempo real (Vestertraining, 2021), (Fernandez, 2021), (Mastermoviles, 2021).

4.6 IoT (Internet of Things)

La definición de IoT podría ser la agrupación e interconexión de dispositivos y objetos a través de una red, donde todos ellos podrían ser visibles e interactuar. Respecto al tipo de objetos o dispositivos podrían ser cualquiera, desde sensores y dispositivos mecánicos hasta objetos cotidianos, cualquier cosa que se pueda imaginar podría ser conectada a internet e

interaccionar sin necesidad de la intervención humana, el objetivo por tanto es una interacción de máquina a máquina, o lo que se conoce como una interacción M2M (machine to machine).

Si se piensa en aplicaciones en el sector productivo, es usado ya en muchas plantas donde los dispositivos y sensores conectados a la red permiten analizar los datos y generar alarmas y mensajes que son enviados a los distintos usuarios para que tomen las acciones necesarias o incluso iniciar protocolos de actuación de forma automática, sin interacción humana, para corregir o tratar dichas alarmas.

En aplicación del sector ganadero dónde la monitorización biométrica y la geolocalización es un factor que ayuda a los ganaderos a que sus animales estén siempre controlados.

Las aplicaciones que se le dan son muy diversas mientras se requiera obtener o transferir datos, estos datos luego estarán disponibles para que el usuario los manipule como le parezca ya sea usándolos en una base de datos, usarlos para controlar otros dispositivos incluso de forma remota. El sector ganadero dónde se realiza la monitorización biométrica y la geolocalización es un factor que ayuda a los ganaderos a que sus animales estén siempre controlados (Gracia, s.f.).

5 Comunicación MQTT

5.1 Protocolo MQTT

Es un protocolo de comunicación M2M (Machine to machine) se basa en el modelo cliente/servidor para enviar y recibir mensajes. Los clientes se conectan a un nodo central llamado "Broker" el cual contiene la información requerida.

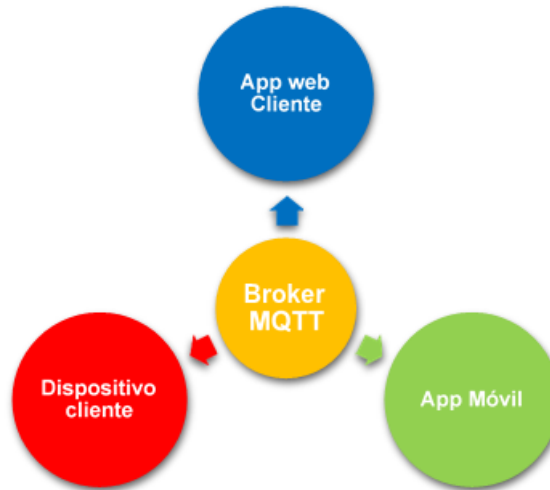


Figura 1. Broker MQTT

Este sistema funciona con jerarquías, gracias a esta estructura se puede filtrar la información que el usuario (cliente o dispositivo) desea recibir. El usuario se suscribe a un “topic” (tema) para poder recibir el mensaje, es decir, que dará prioridad una vez que el usuario se suscribe. Los usuarios se conectan al broker, esta sesión se mantiene abierta hasta que el usuario la cierra o el tiempo de sesión abierta se agote (Llamas, 2020).



Figura 2. Tópicos MQTT

En cada grupo tenemos diferentes servicios (topics) en los cuales se requiere mantener el control, entonces cada piso va a disponer de un dispositivo diferente que se encargue de monitorear cada servicio, por supuesto se puede disponer de un mismo dispositivo que pueda controlar todos los servicios a la vez. Lo que se busca es obtener y almacenar los datos producidos por el uso de dispositivos para su posterior análisis.

Un usuario se suscribe al tópico, por ejemplo, usuario 1 se suscribe a grupo 1, porque ese usuario requiere de datos que pertenecen a ese grupo, luego deberá estar suscrito a un subtópico, por ejemplo, “seguridad” y finalmente se suscribe a “Ingreso a nivel”

El broker MQTT publica el mensaje en la dirección:

Tópicos/Grupo1/Seguridad/Ingreso_a_nivel

Se debe ser preciso al asignar o suscribirse a la información que se desea recibir ya que pueden existir confusiones tales como:

Caso 1: Estar suscrito a “seguridad” y recibir tanto “ingreso a nivel” como “cámaras de seguridad”.

Caso 2: Estar suscrito a todo el grupo 1 y sólo requerir la información de un solo tópico.

Caso 3: Estar suscrito a todo el grupo 2 y recibir también la información de Grupo 1, ya que se reciben los mismos servicios.

Caso 4: Recibir información de todo el nivel de “seguridad” en ambos grupos tanto “ingreso a nivel” como “cámaras de seguridad”.

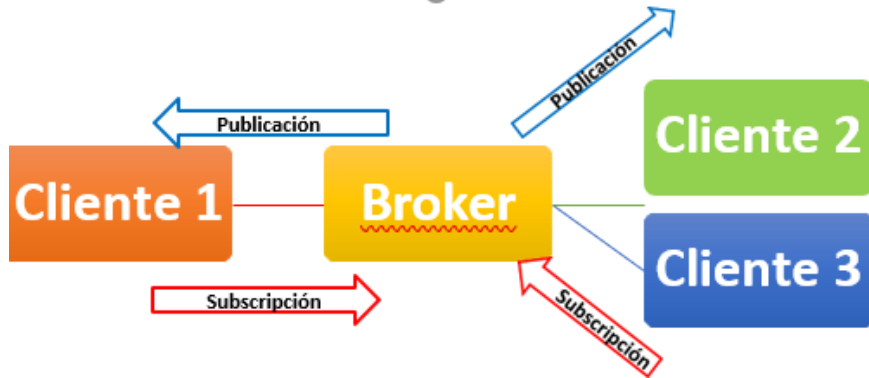


Figura 3. Comunicación Broker MQTT

5.2 MQTT y sensores

Un sensor no tiene la capacidad de computacional para enviar datos a una red o disparar alertas, este enfoque no es ocupación de los sensores ya que por lo general son dispositivos pequeños. El protocolo MQTT complementa las capacidades limitadas de los sensores, incluso de los más avanzados, para poder transmitir la data obtenida por los mismos e incluso llegar a automatizar procesos industriales de forma remota, (Valerie Lampkin, 2012).

Gracias al protocolo MQTT podemos alcanzar niveles de automatización muy avanzados como para acoplar bases de datos y otras herramientas de automatización que facilitan la obtención, guardado y consulta de información.

6 ESP32: Fundamentos y alcances

6.1 ESP32

Placa electrónica de desarrollo de grandes capacidades, el ESP32 ofrece la posibilidad de trabajar con WiFi, Bluetooth, BLE (Low Energy Bluetooth). Internamente posee varios periféricos: sensores táctiles, sensor de efecto Hall, amplificadores de bajo ruido, interfaz para tarjetas SD, Ethernet, SPI de alta velocidad, I2S e I2C, (Naylampmechatronics, 2020).

El ESP32 supera al ARDUINO UNO y a su antecesor el ESP8266 ya que viene a ser una versión más avanzada en la familia de microcontroladores Espressif Systems, con el ESP32 se busca desarrollar proyectos aún más complejos y poder aprovechar mejor las capacidades que se implementaron a este nuevo dispositivo (programarfacil, 2021).

6.1 LoRa

Tecnología inalámbrica de enlace de radio que utiliza modulación en radiofrecuencia. Esta tecnología también conocida como CSS (Chirp Spread Spectrum) se utiliza en la comunicación militar ya que se pueden aprovechar las largas distancias (por lo general kilómetros) para comunicarse.

Algunas ventajas de la tecnología LoRa:

- Alta tolerancia a las interferencias
- Alta sensibilidad para recibir datos (-168dB)
- Bajo Consumo (hasta 10 años con una batería*)
- Largo alcance 10 a 20km
- Baja transferencia de datos (hasta 255 bytes)
- Conexión punto a punto
- Frecuencias de trabajo: 868 Mhz en Europa, 915 Mhz en América, y 433 Mhz en Asia

Esta tecnología también trabaja en conjunto con las redes IoT para explotar las utilidades de muchos dispositivos que tengan la capacidad de comunicarse de manera remota (alfaiot, 2021).

6.2 ESP32 LoRa

Es una versión del ESP32 que contiene un módulo LoRa además de las previas mejoras que se implementaron, esto permite tener más posibilidades de transmisión y recepción de información entre dispositivos que se comuniquen de manera remota y/o entre varios ESP32.

Para aplicaciones IoT se puede utilizar el manejo de información del ESP32 y utilizar el módulo LoRa para la transmisión de dicha información la cual luego podrá ser utilizada para otras aplicaciones o guardar dicha información en una nube.

Teniendo la tecnología LoRa se pueden implementar otros métodos de comunicación para no depender únicamente de internet para realizar múltiples tareas aprovechando los demás módulos implementados en el ESP32.

6.3 Bases de Datos

Se pueden definir como gestores de información recopilada y ordenada electrónicamente dicha información es captada y administrada por sistemas computacionales, una base de datos puede almacenar cualquier tipo de valor ya sea numérico, de tipo cadena, de tiempo, etc. Como la información es almacenada puede ser reutilizada para su posterior manipulación, (Olaya, 2021), (Valdés., 2021).

6.4 HeidiSQL

Es un software de código abierto que permite gestionar información con una amplia gama de herramientas como crear tablas, relacionar tablas, conectar múltiples servidores, comunicación SSH, otorgar acceso a usuarios, etc. Una importante característica de esta base de datos es que te permite editar la información guardada e incluso otorgar etiquetas para diferenciar variables, usuarios, datos, entre otros elementos, (Heidisql, 2021), (Aguilar., 2021).

6.5 EMQX

Es un motor de servicio MQTT escalable en tiempo real que permite la administración de miles de usuarios, tiene múltiples herramientas para administrar puertos de comunicación, dar acceso a usuarios y establecer comunicación entre cliente MQTT y el servidor web. En el panel principal podemos configurar a los puertos para siempre estén en modo escucha y estén pendientes de recibir y responder a las alertas o notificaciones de parte de los servicios que requieren comunicarse al servidor MQTT, (Chaparro., 2021), (Docs Emqx, 2021).

Los alcances que puede llegar a tener el ESP32 dependerán de las tareas que se desean realizar, la ventaja de utilizar el ESP32 es que permite realizar múltiples tareas y además se pueden implementar múltiples servicios para trabajar en conjunto tales como: obtener datos de un sensor, guardar información, mostrar información, hacer cálculos, conectar a servidores, entre otros y de ser necesario tener la posibilidad de realizar todo al mismo tiempo.

Fundamentación teórica

Los problemas que han investigado son las complicaciones que hay en supervisión de un terreno agrícola, dado las dificultades por las condiciones del suelo, no tomar medidas correspondientes porque las variables que necesitan ser atendidas no tienen una apreciación cuantitativa más la tienen cualitativa, por este motivo el usuario o trabajador agrícola no puede tomar medidas correctivas a tiempo, como aumentar la cantidad de riego, o saber si sus plantaciones tienden a enfermar.

Por este motivo se desarrolló un prototipo que pueda hacer las mediciones a una distancia de 20m (campo abierto) y de esa manera soluciona la problemática de ir a un lugar difícil de llegar a más de una ocasión y desgastar al usuario por gusto, también se le implementó una medición cuantitativa que se puede verificar en una página web con sus respectivas variables y de esa manera el usuario sabrá las acciones a realizar tanto para prevenir pérdidas como para tener una muestra cuantitativa y ver en la base de datos cómo va el desarrollo de su cultivo.

III. Marco metodológico

Para este proyecto se aplicó la metodología de Scrum, la cual consiste en trabajar en equipo sin llevar una planificación secuencial montando diferentes tareas para completar la meta final. Se planifica según los avances con muchas incertidumbres con una gran cantidad de cambios a última hora.

Para la investigación respectiva se aplicaron los siguientes métodos:

Método analítico

Estudio basado en la experimentación y la lógica empírica, descompone en elementos básicos los fenómenos que se estudian.

Método experimental

Conlleva la observación, manipulación y registro de variables que intervienen en el fenómeno de estudio. El objetivo de este método es descubrir las causas de dicho fenómeno.

Estructura del Proyecto

El proyecto trabaja en 4 niveles:

Nivel 1 Sensores & ESP32: Los módulos ESP32 se van a encargar de enviar y recibir la información, el módulo transmisor capta la información de los sensores y la muestra en su

pantalla integrada para luego transmitir esa información al módulo receptor, el módulo receptor muestra la información recibida en su pantalla para luego subir dicha información a la página web.

Nivel 2 comunicación web: El módulo receptor utiliza la función WiFi para conectarse a una red de internet y con las credenciales MQTT sube y muestra la información obtenida por los sensores en tiempo real en la página web.

Nivel 3 aplicación web: Desde cualquier buscador se accede al dominio, con las credenciales de usuario accedemos al “dashboard” donde se muestran las lecturas de los sensores. Se añadieron gráficos para poder visualizar mejor los cambios que se den en las lecturas. Con instrucciones PHP vamos a permitir el acceso al usuario a al dashboard.

Nivel 4 base de datos: En la programación de la página web está establecido el mensaje que se va a publicar en la base de datos. Las lecturas obtenidas por los sensores se van a guardar en HeidiSQL, específicamente en una tabla, le damos acceso a Heidi para que el servidor pueda almacenar la información. Con instrucciones PHP vamos a permitir la comunicación entre la base de datos y lo mostrado en la página web.

Comunicación entre Sensores & ESP32

Los sensores se comunican a través de I2C, en la programación de los ESP32 se definen qué pines van a ser utilizados para dicha comunicación ya que puede haber conflictos porque la pantalla integrada en el mismo también utiliza comunicación I2C, también se debe definir la frecuencia en la que van a trabajar los ESP32, para el territorio ecuatoriano se usa la banda de 433 MHz.

La comunicación entre Los ESP32 transmisor y receptor va a ser por tecnología LoRa, es decir ellos siempre van a estar comunicados y enviando y recibiendo datos sin depender de internet.

El ESP32 transmisor crea una cadena de caracteres para poder enviar un sólo paquete al ESP32 receptor con la información requerida, es importante establecer variables diferenciadas al momento de enviar y recibir ya que pueden ser confundidas por el código.

ESP32 & Servidor MQTT

Dentro de la programación del ESP32 vamos a encontrar las credenciales para poder acceder al servidor MQTT y poder visualizar los mensajes que llegan a la web y en el panel EMQX.

Registro en Base de Datos

Los datos subidos por el ESP32 receptor están estructurados en una línea de caracteres, este mensaje está siendo registrado en la página web para luego interpretarlo con la codificación correspondiente para ser subido desde la página web a la base de datos.

El proyecto pretende dar la facilidad de visualizar los datos que se registran en cualquier lugar y a cualquier hora con el objetivo de no descuidar los cambios que se den en el campo de trabajo y si se presenta el caso de que ocurra algún inconveniente sea más fácil la toma de decisiones para poder mitigar dichos problemas de la maera más eficiente.

Funcionamiento del Proyecto

El proyecto utiliza los recursos integrados para que sea posible la obtención, la subida, visualización y registro de la información al que está designado.

Para aclarar aún más las dudas que han surgido hasta ahora se procede a explicar con más detalle qué se encarga de qué y cómo se realiza dicha tarea.

Diseño del circuito

Para poder medir la humedad la temperatura y la presión atmosférica se utilizan los siguientes circuitos integrados:

BME280

Es un sensor de temperatura, humedad y presión generalmente utilizado en aplicaciones o en dispositivos inteligentes, mide los parámetros con gran precisión y con respuesta rápidas a los cambios en el entorno.

Rango de operaciones BME280

Los rangos de operaciones en sus diferentes variables son, (Bosch, 2021)

Temperatura: -40 – 85 °C.

Humedad: Humedad relativa: $\pm 3\%$ con tendencia $\leq 2\%$.

Presión: 300 – 1100 hPa Interface: I2C & SPI Voltaje: 1.7 – 3.6 V.

DS18B20

Sensor digital de temperatura, utiliza una función de bus de datos para la comunicación. Existen varias presentaciones, pero en este caso se va a enfocar en la versión de sonda. Posee 3 pines, uno de datos, uno para alimentación y uno para tierra, (Naylampmechatronics, 2021).

Rango de operaciones DS18B20

Temperatura: -55 – 125 °C.

Interface: 3 pines. 1 GND, 1 VCC, 1DQ.

Voltaje: 3 - 5V.

Heltec LoRa WiFi 32 V2

Placa electrónica que tiene integrado un ESP32 con pantalla OLED y un chip nodo LoRa. Posee características similares al ESP32 clásico, pero con la posibilidad de conectarse a una red WiFi y comunicarse por tecnología LoRa con otros dispositivos.

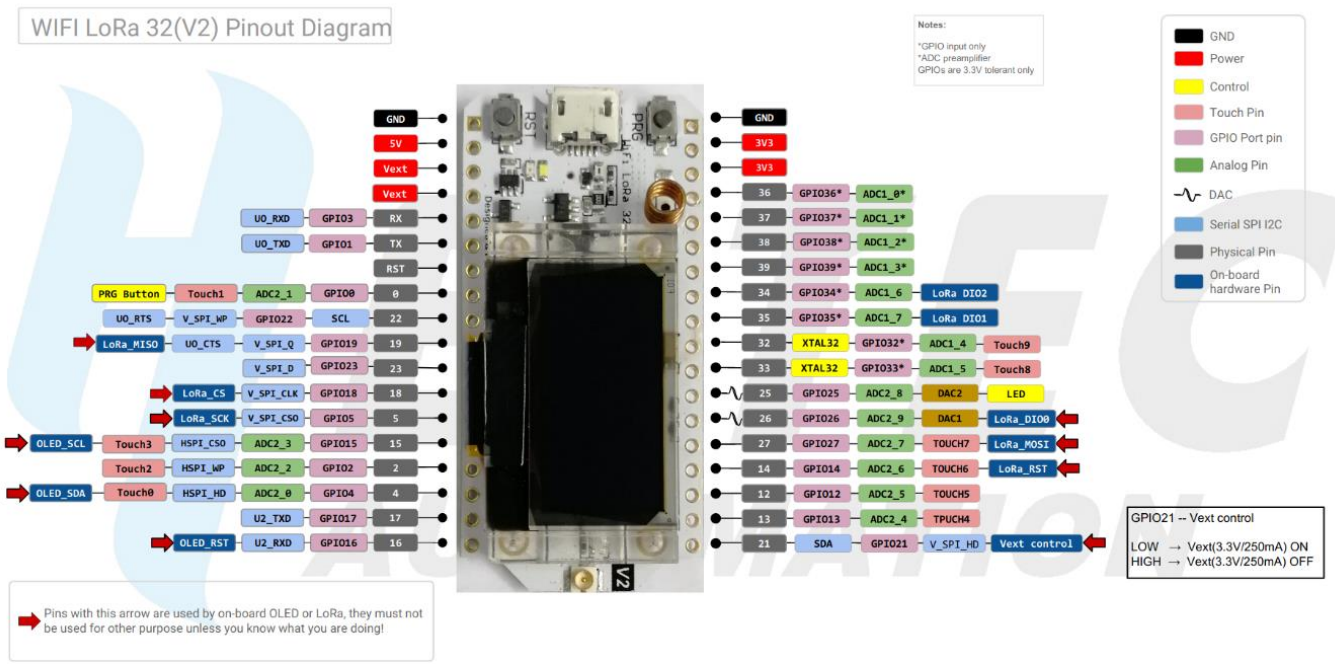


Figura 4. diagrama de pines para Heltec LoRa Wi-Fi (V2)

Fuente: <http://www.weargenius.in/wp-content/uploads/2018/12/df.png>

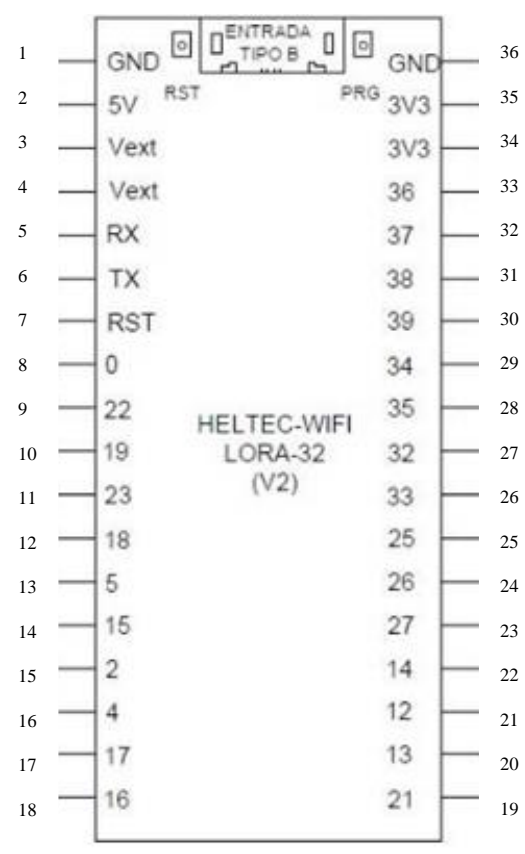


Figura 5. Diagrama de pines para Heltec LoRa Wi-Fi (V2)

¿Qué incluye?

Dependiendo del distribuidor

- Caja.
- 2 juegos de pines.
- Incluye una antena para mayor alcance.

Especificaciones

Chip maestro: ESP32 (240 MHz dual-core)

Comunicación inalámbrica:

WiFi: 802.11 b/g/n (Hasta 150Mbps).

Bluetooth: V4.2 y BLE.

LoRa: Nodo a nodo o LoRaWAN Salida de LoRa (dB): $19\text{dB} \pm 1\text{dB}$ Memoria FLASH: 8MB.

Pantalla: 0.96 pulgadas 128*64 OLED display

Memoria RAM: 520KB.

Tamaño: 50.2 x 25.5 x 9.74 mm

Tipo de batería: 3.7V Litio (SH1.25 x 2)

Interface: Micro USB x1, Antena LoRa (IPEX) x1, 36 pines.

Características eléctricas

En la siguiente tabla se demuestra las características eléctricas:

Cualidad eléctrica	Condición	Mínimo	Regulación	Máximo
Fuente de poder	USB ($\geq 500\text{mA}$)	4.7V	5	6
	Batería de litio ($\geq 250\text{mA}$)	3.3V	3.7	4.2
	Pin 3V3 ($\geq 150\text{mA}$)	2.7V	3.3	3.5
	Pin 5V ($\geq 500\text{mA}$)	4.7V	5	6
Consumo (mA)	Escáner WiFi		115	
	WiFi AP		135	
	LoRa 10dB salida		50	
	LoRa 12dB salida		60	
	LoRa 15dB salida		110	
	LoRa 20dB salida		130	
Salida	Pin 3V3			500mA
	Pin 5V (USB)		Igual a la corriente de entrada	
	Vext 3V3			500mA

Tabla 2. Características eléctricas del controlador, (Heltec, 2020).

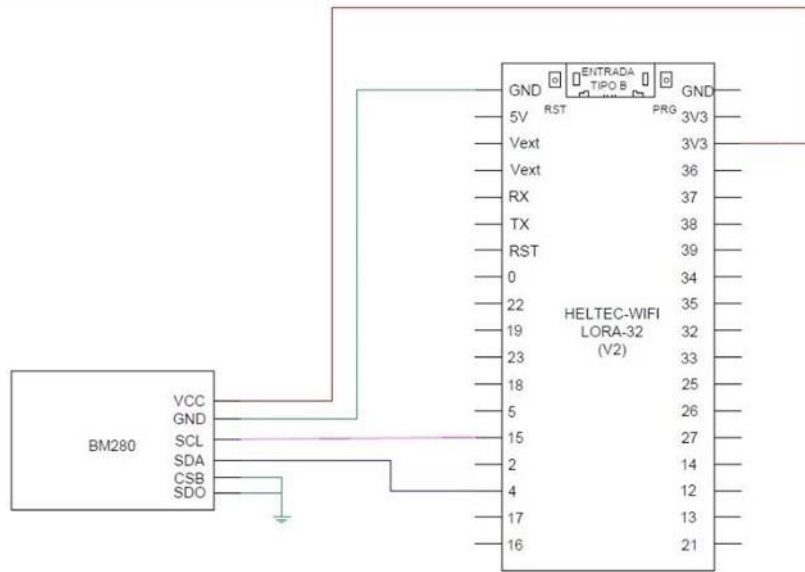


Figura 6. Diseño del circuito para el sensor BME 280.

El sensor tiene 6 pines para su conexión, 4 pines de datos y 2 para voltaje y tierra. SDA y SCL van a ser los pines de datos que se van a comunicar con el ESP32 para comunicación I2C, respectivamente se van a conectar en los pines GPIO4 y GPIO15. Este sensor se va a encargar de medir la humedad y la presión.

VCC	3V3
GND	GND
SDA	GPIO4
SCL	GPIO15

IMPORTANTE

Revisar la imagen referencial 1 y 2 para identificar pines y nomenclaturas

Tabla 3. Especificaciones de conexión adjunta a la figura 4.

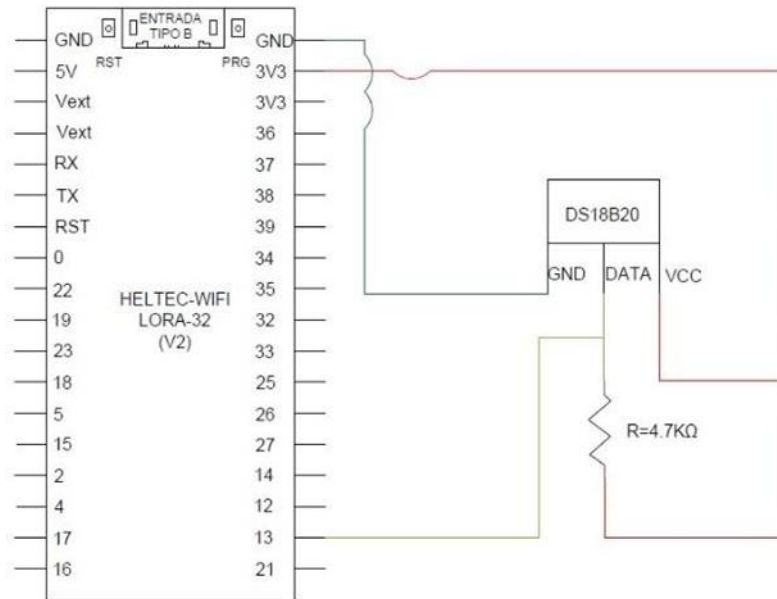


Figura 7. Diseño del circuito para el sensor DS18B20

El sensor tiene 3 Canales para su conexión, 1 VCC, 1 GND, DQ/DATA, con el ESP32 para comunicación I2C, respectivamente se van a conectar en los pines GPIO4 y GPIO15. Este sensor se va a encargar de medir la temperatura.

VCC	3V3
GND	GND
DQ/DATA	GPIO13

IMPORTANTE

Revisar la imagen referencial 1 y 2 para identificar pines y nomenclaturas

Tabla 4. Especificaciones de conexión adjunta a la figura 5.

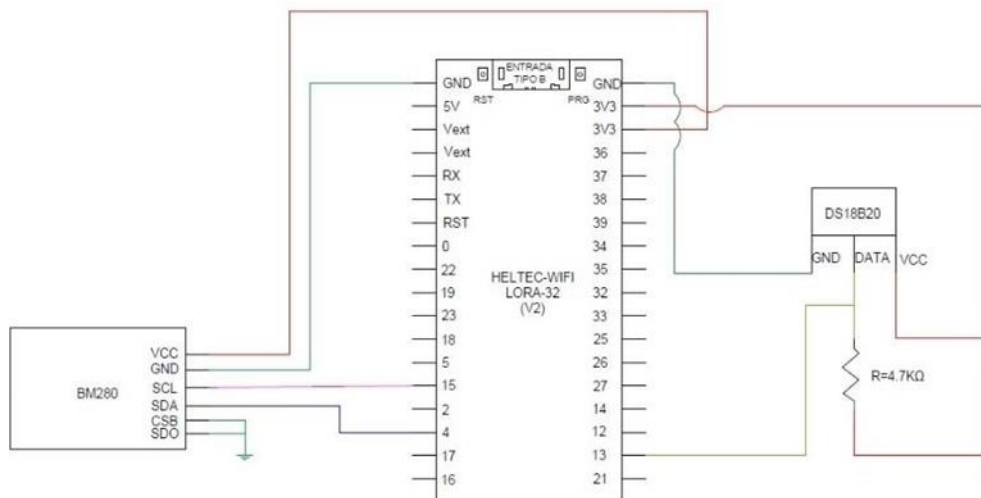


Figura 8. Diseño del circuito completo.

Finalmente quedan los 2 sensores conectados y listos para captar los parámetros, el transmisor se encargará de las lecturas y las enviará al receptor a través de tecnología LoRa. Se designaron los pines para evitar conflictos de comunicación.

BME280	Pin
VCC	3V3
GND	GND
SDA	GPIO4
SCL	GPIO15
DS18B20	Pin
VCC	3V3
GND	GND
SDA	GPIO4
SCL	GPIO15

IMPORTANTE

Revisar la imagen referencial 1 y 2 para identificar pines y nomenclaturas

Tabla 5. Especificaciones de conexión adjunta a la figura 6.

Codificación

Este proyecto va a ser codificado en lenguaje C++ utilizando el IDE de Atom, los comandos utilizados están en un lenguaje que son de tipo pseudocódigo.

Atom

Es un editor de texto que permite programar en una gran cantidad de lenguajes para desarrollo de programas y aplicaciones gráficas.

Atom fue diseñado para ser personalizable, por eso permite el desarrollo de aplicaciones en diferentes lenguajes como C, C++, Node.js, HTML, JavaScript, PHP, entre otros, (Atom, 2021).

Platformio

Es la herramienta de desarrollo específicamente utilizada para programar los ESP32, este software contiene herramientas similares a las de Arduino tales como compilar programas, subir programas a la placa, monitor serial, entre otras. Una gran ventaja de este entorno es que permite conectarse a un servidor y gestionarlo, (Platformio, 2021).

Codificación Transmisor (Tx)

Librerías

El ESP32 Tx va a utilizar las siguientes librerías:

T x	Funció n
Arduino	Lenguaje C++
stdlib	Lenguaje C
Dallas Temperature	BME280
AsyncDelay	BME280
SoftWire	BME280
Adafruit Unified Sensor	BME280
Adafruit BME280 Library	BME280
Heltec ESP32 Dev- Boards	Programación para ESP32 marca HELTEC

Tabla 6. Librerías para la codificación del transmisor (Tx)

```

1 //LIBRERIAS
2 #include <Arduino.h>
3 #include <WiFi.h>
4 #include <stdlib.h>
5 #include <OneWire.h>
6 #include <DallasTemperature.h>
7 #include <AsyncDelay.h>
8 #include <SoftWire.h>
9 #include <Adafruit_Sensor.h>
10 #include <Adafruit_BME280.h>
11 #include "heltec.h"
12 #include "images.h"

```

Figura 9. Librerías incluidas en la codificación del transmisor (Tx)

```

68 #define BAND 433E6
69 #define SDA 4
70 #define SCL 15

```

Figura 10. Codificación para el ESP32 Transmisor (Tx)

ESP32 va a trabajar con frecuencias 300MHz – 3000MHz ya que tenemos un enlace de corto alcance, pero el ESP32 es capaz de transmitir a mayores distancias hasta 2,8km en áreas abiertas, Podemos “setear” las bandas a 433, 868 y 915 MHz, a mayor frecuencia, menor distancia.

```

81 SoftWire sw(SDA, SCL);
82 OneWire oneWireObjeto(pinDato);
83 DallasTemperature sensorDS18B20(&oneWireObjeto);
84 Adafruit_BME280 bme;
85 TwoWire I2Cone = TwoWire(1); //0

```

Figura 11. Codificación del transmisor (Tx), llamado de librerías

La variable pinDato se refiere al Pin designado para datos del sensor DS18b20, fue establecido para el pin GPIO13.

```

95 void startBME(){
96   bool bme_status;
97   bme_status = bme.begin(0x76);
98   if (!bme_status) {
99     Heltec.display->clear();
100    Heltec.display->drawString( 0, 0, "No valid BME280 found");
101    Heltec.display->drawString( 0, 12, "Please check wiring!");
102    Heltec.display->display();
103    Serial.println("Could not find a valid BME280 sensor, check wiring!");
104   }
105 }

```

Figura 12. Codificación del transmisor (Tx), Función para identificar BME280

Para iniciar el BME280 necesitamos que el ESP32 lo identifique, por eso creamos una función en donde establecemos por defecto que lo detecta y que comience a medir los parámetros.

```

107 void getReadings(){
108   bme.setSampling(Adafruit_BME280::MODE_FORCED,
109                 Adafruit_BME280::SAMPLING_X16,
110                 Adafruit_BME280::SAMPLING_X1,
111                 Adafruit_BME280::SAMPLING_X1,
112                 Adafruit_BME280::FILTER_X16,
113                 Adafruit_BME280::STANDBY_MS_0_5 );
114   bme.takeForcedMeasurement();
115   tempBME = bme.readTemperature();
116   hudBME = bme.readHumidity();
117   pressBME = bme.readPressure() / 100.0F;
118
119 }

```

Figura 13. Codificación del transmisor (Tx), Función para obtener lecturas BME280

Para que los sensores midan los parámetros se debe crear una función que obtenga los datos, una vez que se active esta función se podrán ver las medidas en la pantalla del ESP32.

Dentro del "Setup" se procede a llamar la función "StartBme", se la coloca en esta sección para que el programa identifique que existe un BME280 antes de proceder con el resto del programa, es decir, si no identifica al BME280 no se muestran las lecturas. También se procede a llamar al sensor DS18B20 para que también empiece a medir, no se requiere crear una función ya que no necesita de una función para identificarlo.

```
143      startBME();
```

Figura 14. Codificación del transmisor (Tx), Llamado de función de identificación BME280

```
144      sensorDS18B20.begin();
```

Figura 15. Codificación del transmisor (Tx), Llamado de función obtener lecturas DS18b20

En la función “Loop” se va a llamar a las funciones que obtienen los valores de los sensores ya que van a estar obteniendo valores de forma continua y en tiempo real, para el caso del BME280 se llama a la función “getReadings” y para el DS18b20 se procede a solicitar un “requestTemperatures”.

```
154      getReadings();
```

Figura 16. Codificación del transmisor (Tx), Llamado de función “getReadings” BME280

```
157      sensorDS18B20.requestTemperatures();
```

Figura 17. Codificación del transmisor (Tx), Solicitar lecturas del sensor “getReadings” DS18b20

```
158      float temp=sensorDS18B20.getTempCByIndex(0);  
159      char tempstring[20];  
160      dtostrf(temp,3,1,tempstring);  
161      String temperatura(tempstring);
```

Figura 18. Codificación del transmisor (Tx), Conversión de datos a cadena de caracteres DS18b20

Para el sensor DS18B20 creamos una variable con el comando "sensorDS18B20.getTempCByIndex(0);" y también se crea un arreglo donde se va a guardar ese dato, y para finalizar se transforma en cadena de caracteres ya que vamos a enviar un mensaje con las medidas convertidas a cadena de caracteres.

```
192   LoRa.beginPacket();
193   LoRa.setTxPower(14, RF_PACONFIG_PASELECT_PABOOST);
194   LoRa.print("TempBME:! ");
195   LoRa.print(temperatura);
196   LoRa.print(" HudBME:& ");
197   LoRa.print(hudBME);
198   LoRa.print("PressBME:% ");
199   LoRa.print(pressBME);
200   LoRa.println(MensajeBME);
201   LoRa.endPacket();
```

Figura 19. Codificación del transmisor (Tx), Creación del paquete para comunicación LoRa

Para crear el paquete es necesario empezar con "LoRa.beginPacket();", luego establecemos la ganancia con la que se va a transmitir, para este caso es 14dB, pero puede ajustarse hasta un máximo de 20dB.

Codificación Recibidor (Rx)

Librerías

El ESP32 Rx va a utilizar las siguientes librerías:

Rx	Función
Arduino	Lenguaje C++
WiFi	Conectarse a una red WiFi
PubSubClient	Conectarse al servidor MQTT
Heltec ESP32 Dev-Boards	Programación para ESP32 marca HELTEC

Tabla 7. Librerías para la codificación del Recibidor (Rx)

```
1 //LIBRERIAS
2 #include <Arduino.h>
3 #include <WiFi.h>
4 #include <PubSubClient.h>
5 #include <stdlib.h>
6 #include <OneWire.h>
7 #include <DallasTemperature.h>
8 #include "heltec.h"
9 #include "images.h"
```

Figura 20. Librerías incluidas en la codificación del Recibidor (Rx)


```

14  const char* ssid      = "xxxxxxxxxx";
15  const char* password = "xxxxxxxxxx";
16
17  const char *mqtt_server = "prototypealpha.tk";
18  const int  mqtt_port = 1883;
19  const char *mqtt_user  = "xxxxxxxxxx";
20  const char *mqtt_pass  = "xxxxxxxxxx";

```

Figura 21. Credenciales para dar acceso al Recibidor (Rx) a una red WiFi y al servidor MQTT

Se debe otorgar las credenciales necesarias para que el ESP32 se pueda comunicar a través de Internet con la página web y con el servidor MQTT, para conectarse al servidor MQTT se utiliza el puerto 1883.

```

22  WiFiClient espClient;
23  PubSubClient client(espClient);
24
25  long lastMsg = 0;
26  char msg[25];

```

Figura 22. Identificación espClient para el acceso del recibidor (Rx)

Llamamos a las funciones "WiFi" y "PubSubClient" y se va a identificar al Recibidor como espCliente, además creamos un arreglo en el cual vamos a guardar 25 caracteres el cual va a ser el mensaje que vamos a ubicar en la página web y n la base de datos.

```

60 void LoRaData(){
61   Heltec.display->clear();
62   Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
63   Heltec.display->setFont(ArialMT_Plain_10);
64   Heltec.display->setFont(ArialMT_Plain_16);
65   Heltec.display->drawString(5, 5,"T = " + Tempw + " °C");
66   Heltec.display->drawString(5, 20,"H = " + hudBME + "%");
67   Heltec.display->drawString(5, 35,"P = " + pressBME + " hPa");
68   Heltec.display->setFont(ArialMT_Plain_10);
69   Heltec.display->drawString(0, 50,"Receiving Packets From Tx");
70   Heltec.display->drawString(110, 0,"(Rx)");
71   Heltec.display->display();
72 }

```

Figura 23. Función "LoRaData" para el receptor (Rx)

En el receptor crea una función similar a la del transmisor para que se muestren igual y poder notar las diferencias o que en verdad se están comunicando, si se actualizan las lecturas de Tx también deben actualizarse las Rx.

```

79 void cbk(int packetSize) {
80   packet = "";
81   packetSize = String(packetSize,DEC);
82   for (int i = 0; i < packetSize; i++) { packet += (char) LoRa.read(); }
83   rssi = "RSSI " + String(LoRa.packetRssi(), DEC) ;
84   MensajeRecibido=packet;
85   if (MensajeRecibido!=""){
86     int data1=packet.indexOf('!');
87     int data2=packet.indexOf('&');
88     int data3=packet.indexOf('%');
89     int data4=packet.indexOf('#');
90     Tempw=packet.substring(data1+1,data1+7);
91     hudBME=packet.substring(data2+1,data2+7);
92     pressBME=packet.substring(data3+1,data3+8);
93     tempBME=packet.substring(data4+1,data4+8);
94     LoRaData();
95   }
96   else{
97     Tempw="";
98     hudBME="";
99     pressBME="";
100    tempBME="";
101    LoRaData();
102  }
103  }
104  Serial.println(MensajeRecibido);
105 }

```

Figura 24. Función "cbk" para el receptor (Rx)

La función "cbk" se encarga de identificar el mensaje, también se encarga de ordenar el mensaje. El mensaje está separado por símbolos, "!", "&", "%", "#", es decir, el mensaje va a

ubicar las lecturas sobre esos símbolos, luego se construyen nuevas variables en la que se identificará cuántos espacios o cuántos caracteres se van a escribir antes de pasar al siguiente dato para poder visualizar el mensaje “completo”.

```
148 void loop(){
149     if (!client.connected()) {
150         reconnect();
151     }
152     client.loop();
153     long now = millis();
154     if (now - lastMsg > 500){
155         lastMsg = now;
156
157         Serial.println("RSSI ");
158         Serial.println(rssi);
159         String to_send = String(Tempw) + "," + String(hudBME) + "," + String(pressBME);
160         to_send.toCharArray(msg, 25);
161         Serial.print("Publicamos mensaje -> ");
162         Serial.println(msg);
163         client.publish("values", msg);
164         Serial.println(String(packet));
165         Serial.println(Tempw);
166         Serial.println(hudBME);
167         Serial.println(pressBME);
168     }
169     int packetSize = LoRa.parsePacket();
170     if (packetSize) { cbk(packetSize); }
171     delay(10);
172 }
```

Figura 25. Función “cbk” para el receptor (Rx)

En la función “Loop” establecemos que el cliente se reconecte en caso de fallas en, luego procederemos a subir las cadenas de caracteres en el lugar designado para mostrar ese mensaje además de visualizar la llegada del mensaje en el servidor MQTT.

```

186 void setup_wifi(){
187     delay(10);
188     // Nos conectamos a nuestra red Wifi
189     Serial.println();
190     Serial.print("Conectando a ");
191     Serial.println(ssid);
192
193     WiFi.begin(ssid, password);
194
195     while (WiFi.status() != WL_CONNECTED) {
196         delay(500);
197         Serial.print(".");
198     }
199
200     Serial.println("");
201     Serial.println("Conectado a red WiFi!");
202     Serial.println("Dirección IP: ");
203     Serial.println(WiFi.localIP());
204 }

```

Figura 26. Función “setup_wifi” para el receptor (Rx)

La función “setup_wifi” se encarga de conectarse a una red wifi para tener conexión a internet y tener la posibilidad de comunicarse con la página web, el servidor MQTT y con la base de datos.

Página Web y Comunicación Online

Para la creación de la página web se utilizan varios servicios para poder habilitar las características que posee.

Amazon Web Service (AWS)

Es una plataforma de servicios en la nube los cuales pueden ser utilizados para desarrollo de páginas web, aplicaciones, etc. La ventaja de utilizar este servicio es que se no se debe comprar ni poseer centros de datos ni tampoco de servidores físicos y se puede seleccionar el servicio que mejor se adapte al proyecto que se desea realizar, entre las opciones tenemos

servicios de base de datos, almacenamiento informático, inteligencia artificial, análisis de datos, IOT, etc, (Amazon, 2021).

El VPS (Servidor Virtual Privado) del proyecto está alojado en AWS, gracias a esto es posible establecer reglas de entrada y reglas de salida lo cual permite otorgar permisos especiales a usuarios y permitir que se conecten desde cualquier parte del mundo.

Vesta Control Panel

Vesta ofrece servicios de seguridad, servicios de email, servicios de base de datos e incluso servicios de FTP (File Transfer Protocol) permitiendo subir o descargar archivos del servidor, (Vestacp, 2021).

Vesta es el encargado de que el servidor tenga servicios de transferencia de archivos, firewall, certificados SSL o Secure Sockets Layer (capa de sockets seguros) el cual permite la autenticación, encriptación y desencriptación de datos enviados a través de Internet. Incluso se encarga de instalar el motor para que el servidor pueda utilizar una base de datos.

Putty

Permite el acceso a servidores a través de un cliente SSH en modo remoto o serial, se puede aprovechar esta característica para gestionar o administrar un servidor o una máquina remota, (Putty, 2021).

Para acceder al servidor del proyecto se utiliza Putty y con ello se puede administrar el servidor de manera remota o incluso añadir al servidor más herramientas para expandir las utilidades del proyecto.

Plantilla para Desarrollo Web

Para entender cómo funciona la página web se utilizaron plantillas y opciones de Flatkit, Flatkit ofrece una gran variedad de diseños dinámicos y servicios preestablecidos para configurar y personalizar una página web.

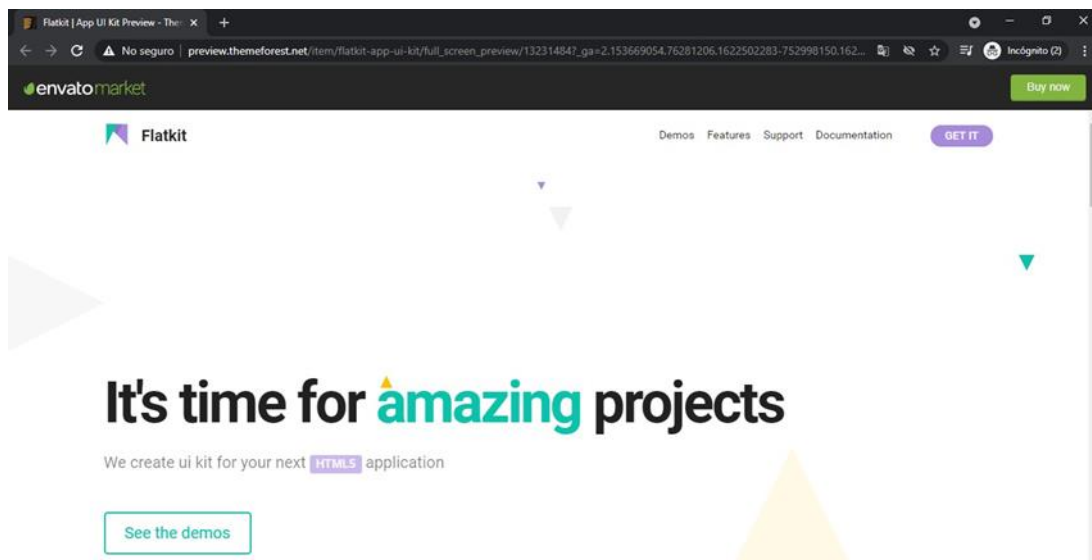


Figura 27. Temas de Flatkit, (Flatkit, 2020).

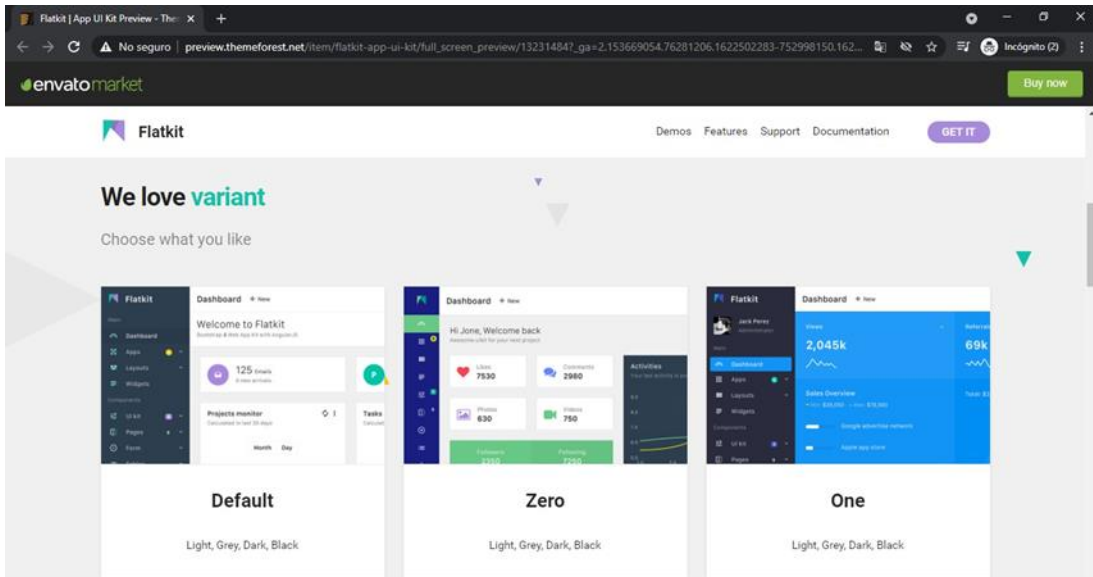


Figura 28. Demos de Flatkit, (Flatkit, 2020).

Se adquirió un paquete de flatkit para construir la página web porque los recursos que flatkit ofrece son muy didácticos y agradables a la vista, también es un entorno bastante amigable y relativamente sencillo de reprogramar.

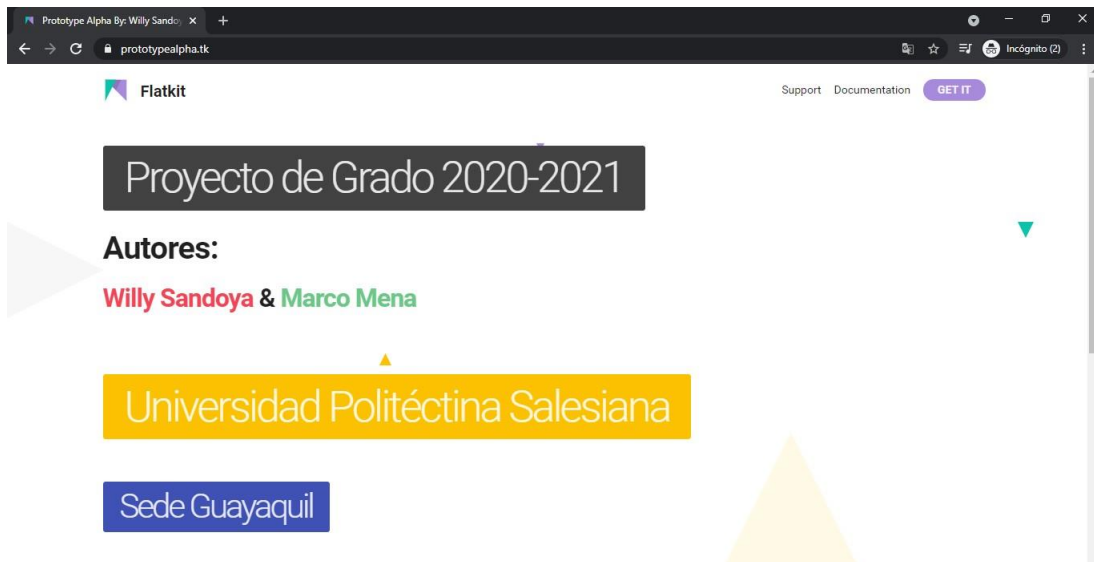


Figura 29. Página principal en la web del Proyecto, (Flatkit, 2020).

En la página principal se colocaron los botones para iniciar sesión y registrarse ya que el proyecto es un prototipo de servicio podemos acceder a nuestro dashboard o panel donde vamos a encontrar la información que se solicita al prototipo.

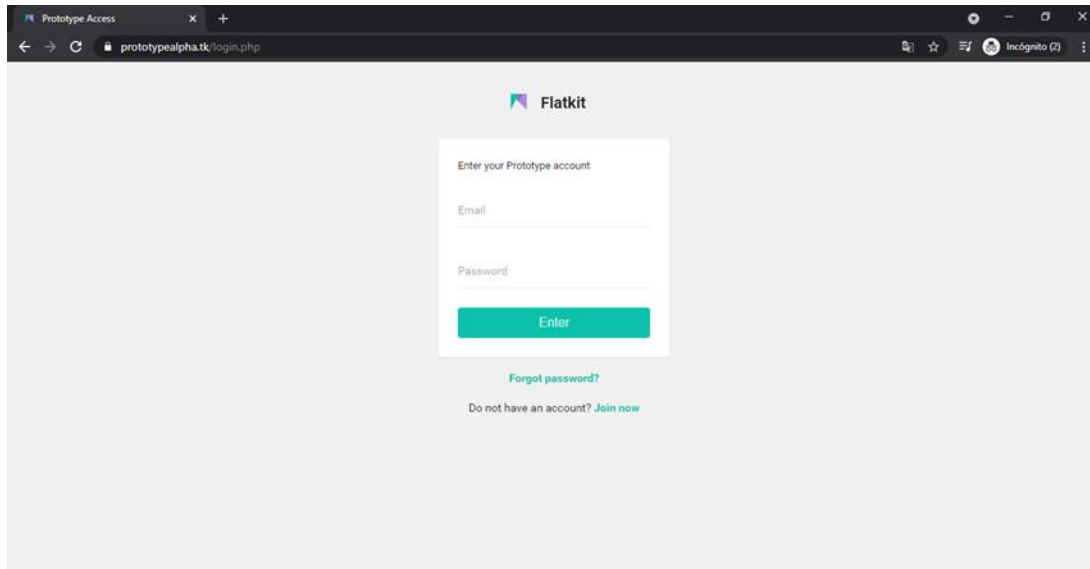


Figura 30. Página Login del proyecto, (Flatkit, 2020).

En la sección “Login” encontramos un formulario donde se va a introducir las credenciales del usuario para que tenga acceso al dashboard.

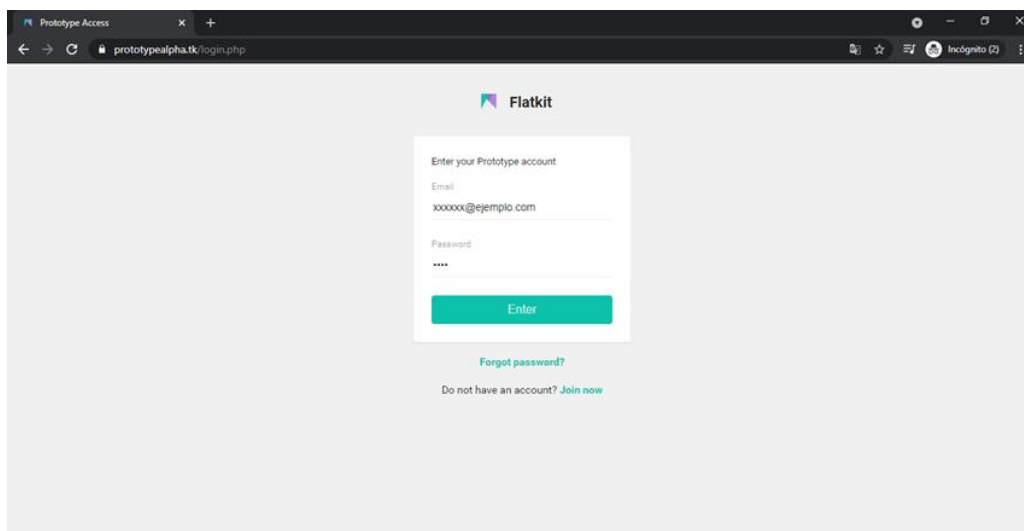


Figura 31. Página Login Introducción de credenciales, (Flatkit, 2020).

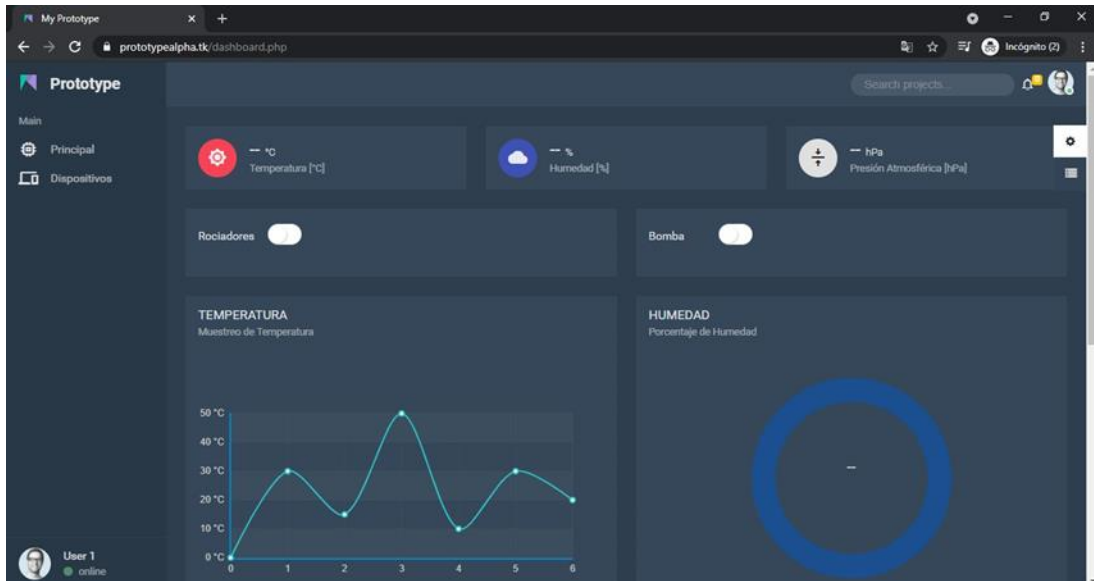


Figura 32. Página Dashboar, (Flatkit, 2020).

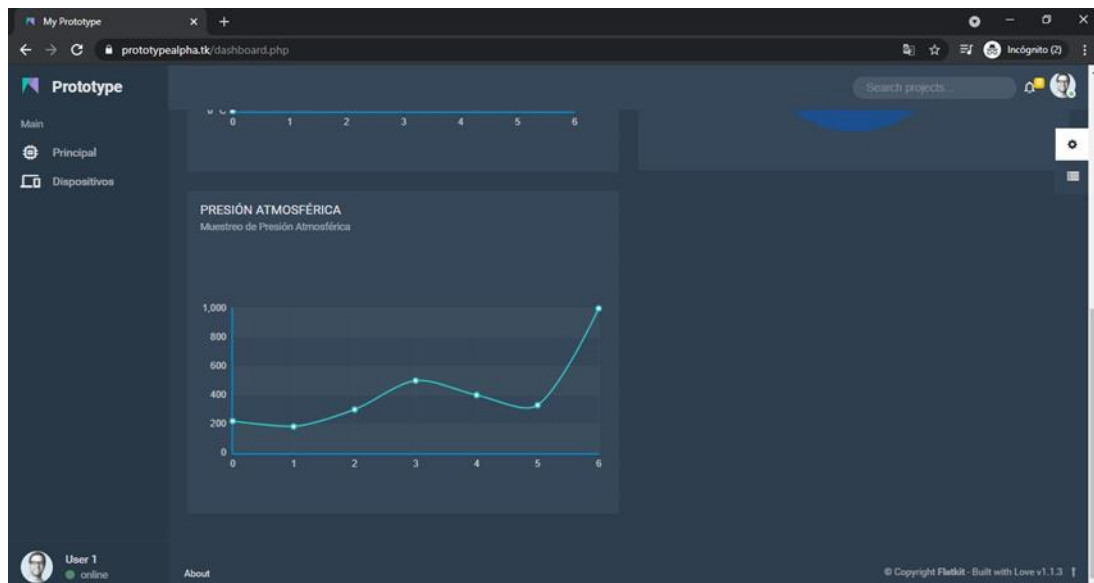


Figura 33. Página Dashboard, (Flatkit, 2020).

En el dashboard se optó por dejar una apariencia predeterminada para poder visualizar cómo se ven los diferentes objetos.

Cuando los ESP32 empiecen a funcionar los datos que se visualizan cambiarán por los datos obtenidos en tiempo real.

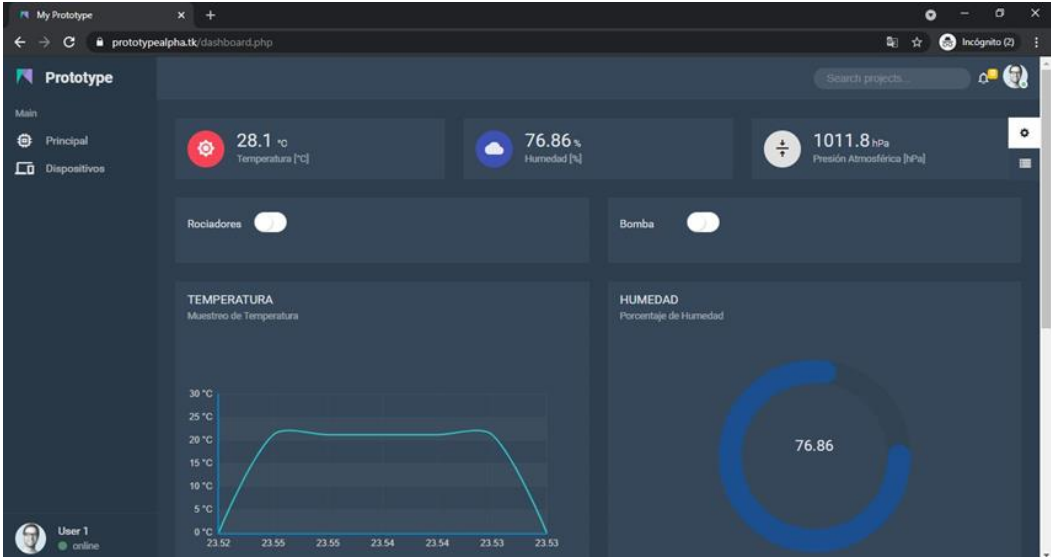


Figura 34. Página Dashboard, actualización de datos, (Flatkit, 2020).

Cuando el receptor empieza a tomar lecturas del transmisor el gráfico predeterminado desaparece y se actualiza según los parámetros que actualmente se están tomando.

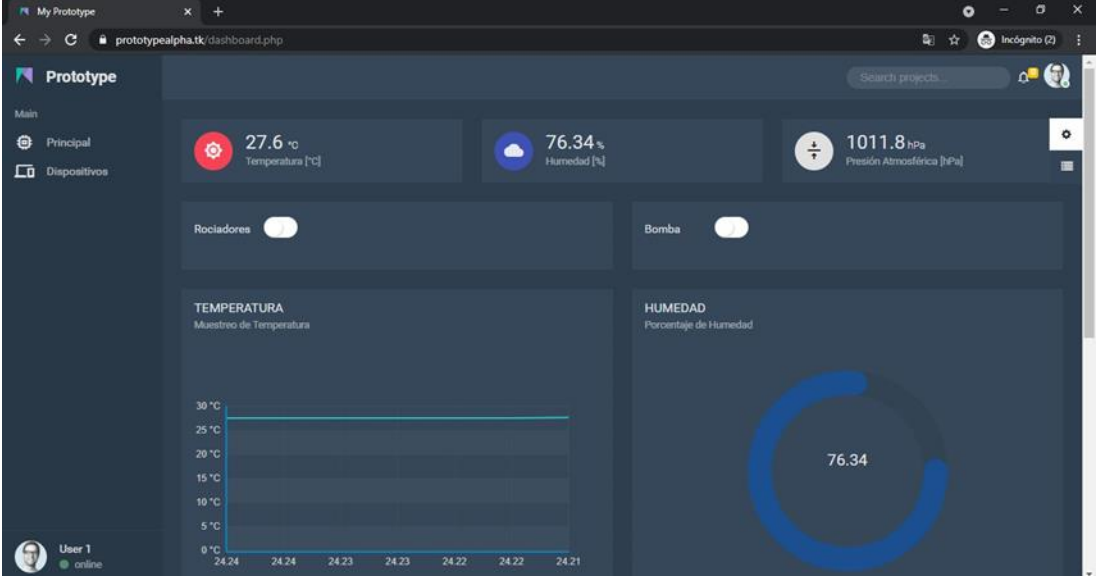


Figura 35. Página Dashboard, estabilidad de datos, (Flatkit, 2020).

Ya que los cambios son muy leves ya que se está hablando de décimas, el gráfico se queda como una función constante.

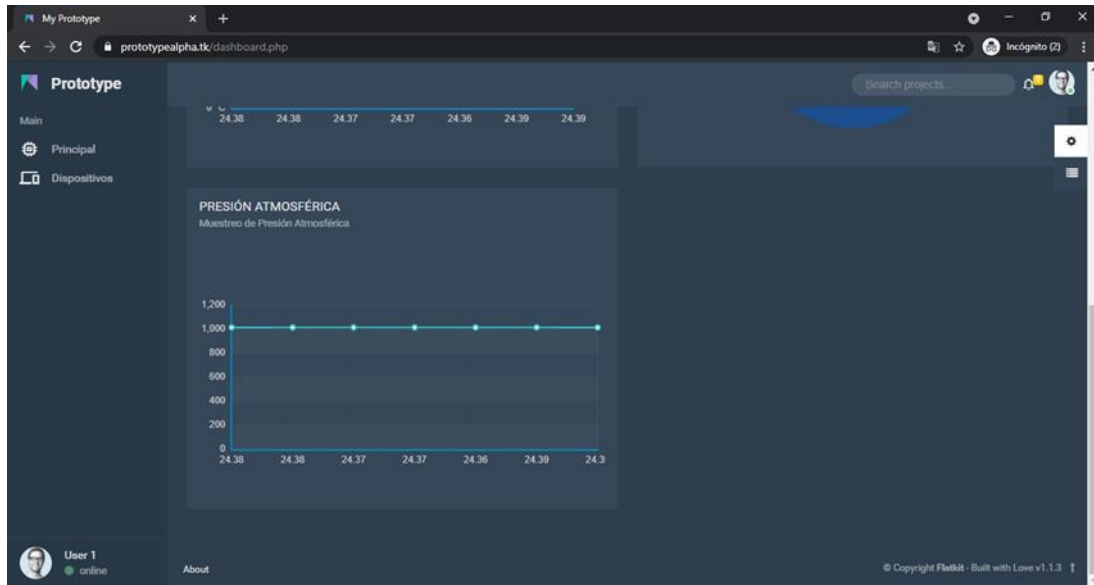


Figura 36. Página Dashboard, estabilidad de datos, (Flatkit, 2020).

Funcionamiento de la Base de Datos

Para tener acceso a la base de datos se llena el formulario con los datos respectivos del servidor.

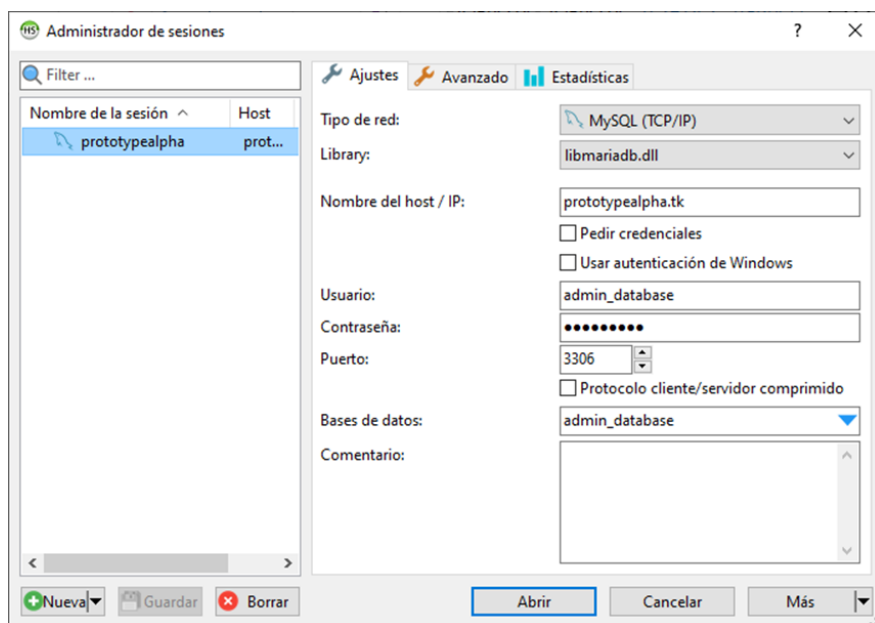


Figura 37. Ventana HeidiSQL, acceso al servidor

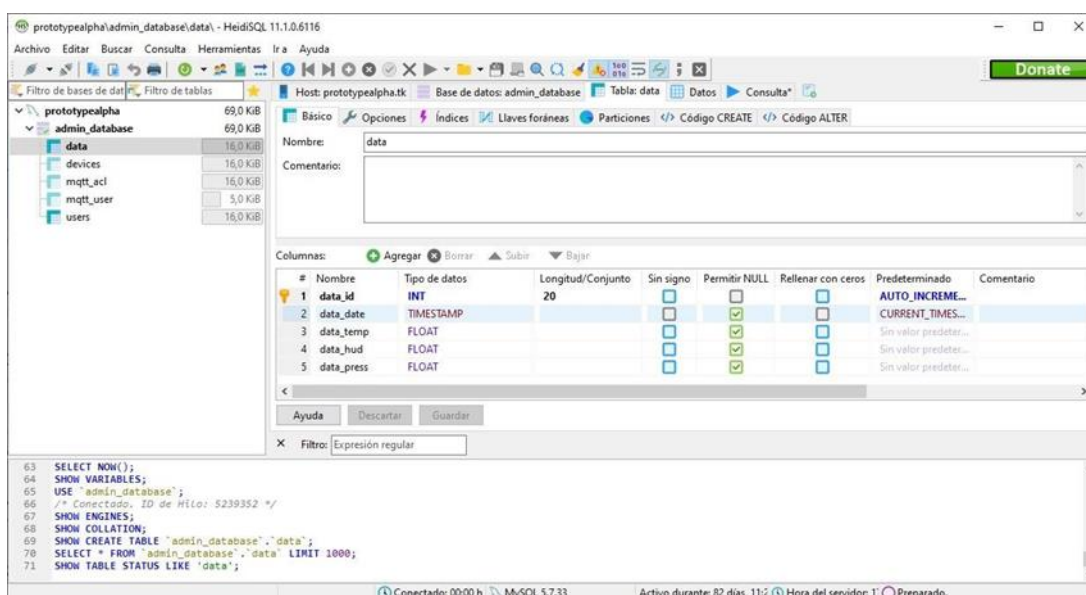


Figura 38. Ventana HeidiSQL, ventana principal del servidor

Después de acceder, aparece nuestro servidor con todas las tablas que se crearon para gestionar el proyecto, tablas para usuarios, administrador MQTT, usuario MQTT, entre otros, cabe mencionar que también está la tabla en la que se va a llenar con los datos que se obtienen de los sensores.

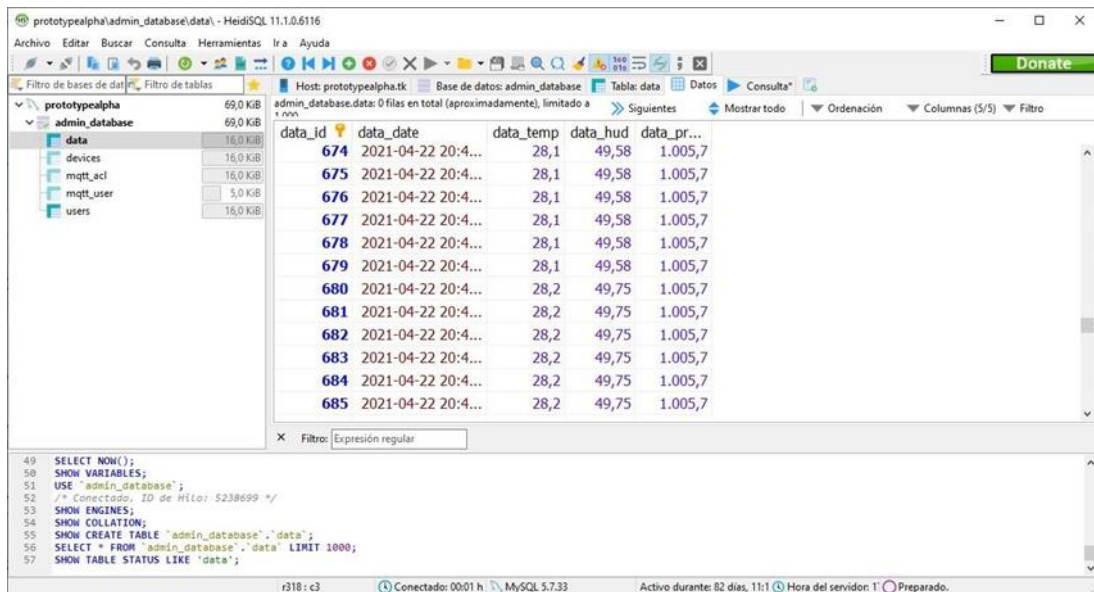


Figura 41. Ventana HeidiSQL, registro en las tablas

Se rellena la tabla, con esto se puede determinar cuándo se realizaron los cambios y determinar las siguientes acciones para mejorar la productividad.

IV. Resultados

Se realizaron varios experimentos que demuestran la premisa de que si se obtienen datos reales de los factores climáticos que afectan a un terreno destinado para la actividad agrícola se puede mejorar la producción.

Experimento 1: Obtención de datos entre 2 ambientes.

Se ha realizado la siguiente práctica la cual permite ver los cambios que sufre una planta de orégano que no mantiene un riego diario, se utilizó el sensor de humedad implementado en el proyecto para realizar las mediciones las cuales son anotadas de manera manual, el procedimiento consiste en dejar sin riego 2 plantas de orégano durante 5 días las cuales serán sujeto 1 y sujeto 2, el sujeto 1 estará dentro de un ambiente cerrado es decir dentro del hogar y el sujeto 2 estará dentro de un ambiente abierto es decir un patio, la toma de medidas fue a las 22h00.



Figura 42. Plantas de prueba

Dado que el procedimiento tarda 5 días se procedió dar una descripción por día, se colocan las plantas respectivamente a los ambientes designados entonces se procedió a dar seguimiento y descripciones de los observado.

Día 1:

Se obtuvo las siguientes medidas del sujeto 1 $T=27.7^{\circ}\text{C}$, $H=69.82\%$ y $P=1011\text{hpa}$.

Se obtuvo las siguientes medidas del sujeto 2 $T=25^{\circ}\text{C}$, $H=66\%$ y $P=1011\text{hpa}$.



Figura 43. Sujeto 1 en el día 1



Figura 44. Sujeto 2 en el día 1

Día 2:

Se obtuvo las siguientes medidas del sujeto 1 $T=26^{\circ}\text{C}$, $H=67\%$, y $P=1012\text{hpa}$.

Se obtuvo las siguientes medidas del sujeto 2 $T=25^{\circ}\text{C}$, $H=73\%$ y $P=1012\text{hpa}$.



Figura 45. Sujeto 1 en el día 2



Figura 46. Sujeto 2 en el día 2

Día 3:

Se obtuvo las siguientes medidas del sujeto 1 $T=26^{\circ}\text{C}$, $H=57\%$ y $P=1013\text{hpa}$.

Se obtuvo las siguientes medidas del sujeto 2 $T=23^{\circ}\text{C}$, $H=75\%$ y $P=1013\text{hpa}$.



Figura 47. Sujeto 1 en el día 3



Figura 48. Sujeto 2
en el día 3

Día 4:

Se obtuvo las siguientes medidas del sujeto 1 $T=26^{\circ}\text{C}$, $H=57\%$ y $P=1014\text{hpa}$.

Se obtuvo las siguientes medidas del sujeto 2 $T=25^{\circ}\text{C}$, $H=67\%$ y $P=1014\text{hpa}$.



Figura 49. Sujeto 1 en el día 4



Figura 50. Sujeto 2 en el día 4

Día 5:

Se obtuvo las siguientes medidas del sujeto 1 $T=26^{\circ}\text{C}$, $H=60\%$ y $P=1014\text{hpa}$.

Se obtuvo las siguientes medidas del sujeto 2 $T=26^{\circ}\text{C}$, $H=60\%$ y $P=1014\text{hpa}$.



Figura 51. Sujeto 1 en el día 5



Figura 52. Sujeto 2 en el día 5

Con los resultados obtenidos de la práctica se deduce desglosado por día:

Día 1:

Las plantas se colocaron en sus respectivos ambientes y tomamos como punto de referencia los valores iniciales.

Día 2:

Se observó que la planta que está en el exterior, es decir, sujeto 2 tiene mayor humedad que el sujeto 1.

Día 3:

Se logró observar que el porcentaje de humedad de sujeto 1 bajo mucho más donde se dedujo que sufre un proceso de deshidratación, el sujeto 2 mantiene la humedad similar del día 3.

Día 4:

Esta vez el porcentaje del sujeto 2 acabo de bajar y el sujeto 1 se mantiene al mismo estado de deshidratación.

Día 5:

Los 2 sujetos tienen un porcentaje de humedad del 60% lo cual se dedujo que están en un estado de deshidratación, por inspección visual se ve que el sujeto 1 tiene un aspecto en deterioro y el sujeto 2 tiene un aspecto estético pero que empiezan a decaer sus hojas.

Experimento 2: Pruebas en campo.

Este experimento fue en un terreno destinado para la actividad agrícola, donde se realizaron las medidas en una cosecha de arroz y una de cacao. El objetivo es demostrar que el dispositivo funciona en una situación real en donde se requiere que las plantas se desarrollen.

El prototipo fue ubicado en el campo de trabajo, a una distancia sugerida por el dueño de la hacienda.



Figura 53. Monitoreo de la siembra de Arroz

Ya con el prototipo receptor en manos del dueño se hizo una pequeña comparación, el conocimiento empírico vs el prototipo. El dueño indicaba que con la experiencia él ya sabía la temperatura y la humedad que necesita el arroz para desarrollarse, pero gracias al prototipo

se puede ser un poco más preciso, ya que el dueño da aproximaciones y el prototipo daba información exacta.



Figura 54. Dueño de hacienda “La Lucha”

Gracias a la combinación del conocimiento empírico y el prototipo se va a mejorar la producción porque se pueden realizar mejoras y mitigar problemas de manera más precisa. Se dejó al prototipo trabajar por 2 días, el dueño sugirió 2 días porque así él pudo comparar su conocimiento y experiencia contra los datos que el dispositivo mostraba.

El dueño de la hacienda nos dio un recorrido por sus terrenos y nos llevó a la siembra de cacao. Con esto se nos ocurrió hacer una segunda prueba en la siembra de cacao.



Figura 55. Cultivos de cacao

El prototipo fue ubicado en el campo de trabajo, a una distancia sugerida por el dueño de la hacienda, para ser más específico fue a 20m de la casa. El dueño indicó que mientras se esté en constante gestión de condiciones o en constante localización y resolución de problemas

en el terreno, el producto va a desarrollarse mejor obteniendo productos de mejor calidad y por supuesto poder comercializarlos a mejor precio.



Figura 56. Receptor mostrando en pantalla las variables del transmisor en campo

Para conocer un poco más las condiciones en las que se desarrolla mejor el arroz, se muestra en la siguiente tabla:

Efecto de la temperatura (°C) sobre el crecimiento y el desarrollo de la planta de arroz

Crecimiento y desarrollo de la planta	Baja temperatura		Alta temperatura		Temp. óptima
	Rango	Efecto	Rango	Efecto	
Germinación	10	<i>Inhibición</i>	45	-	20-35
Emergencia de la plántula	12-13	<i>Demorada</i>	35	-	25-30
Enraizamiento	16	<i>Raquitismo</i>	35	-	25-28
Hoja	7-12	<i>Decoloración de la hoja, raquitismo</i>	45	<i>Punta blanca, bandas cloróticas y manchas</i>	31
Macollaje	9-16	<i>Reducido</i>	33	<i>Reducido</i>	25-31
Iniciación de la panoja	15	<i>Demorada</i>	-	<i>Panoja blanca</i>	-
Diferenciación de la panoja	15-20	<i>Degeneración del ápice de la panoja, alta esterilidad de la espiguilla</i>	38	<i>Número reducido de espiguillas</i>	-
Exerción de la panoja	22	<i>Exerción incompleta, floración demorada</i>	35	<i>Esterilidad</i>	30-33
Grano	12-18	<i>Madurez irregular</i>	30	<i>Menor llenado del grano</i>	20-25

Tabla 8. Efectos de la temperatura sobre el crecimiento y desarrollo del arroz.

Fuente: <https://www.fao.org/3/y2778s/y2778s04.htm>

Para conocer un poco más las condiciones en las que se desarrolla mejor el cacao, se realizó la siguiente investigación.

Clima ideal

Los factores climáticos críticos para el desarrollo del cacao son la temperatura y la lluvia. A estos se le unen el viento y la luz o radiación solar. El cacao es una planta que se desarrolla bajo sombra. La humedad relativa también es importante ya que puede contribuir a la propagación de algunas enfermedades del fruto. Estas exigencias climáticas han hecho que el cultivo de cacao se concentre en las tierras bajas tropicales.

Los factores climáticos que afectan directamente al cacao son la temperatura y la lluvia que por su puesto esto se puede traducir a que el cacao necesita de una temperatura mínima de 21°C, y estar expuesto entre 70% y 80% de humedad.

Factores que afectan al desarrollo del cacao

Agua: El cacao debe recibir 1500 y 2500 mm de agua (zonas cálidas) y 1200 y 1500 mm (zonas frescas), menos de esto baja las probabilidades de supervivencia del cacao.

Un milímetro (mm) de agua equivale a vaciar un litro de líquido en un espacio de un metro cuadrado.

Viento: Existen zonas donde la velocidad del viento es entre 14 y 15km/h lo cual provoca que las plantas pierdan hojas de manera prematura en el caso del cacao si es necesario se debe proteger a la planta con un cortaviento.

Entre otros factores debemos considerar también, la zona, la latitud, la altitud y considerar que tan adecuado es el suelo.

A continuación, se crearon unos gráficos indicando cómo fueron variando los datos del prototipo:

Temperatura

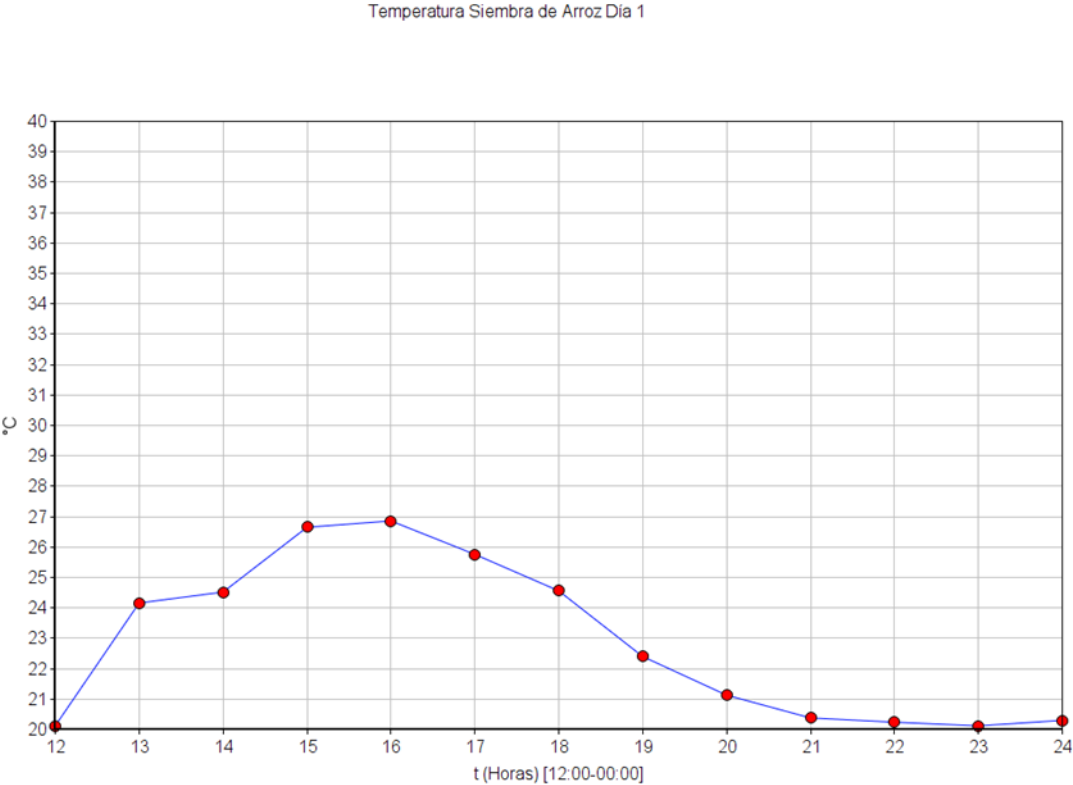


Figura 57. Temperatura Siembra de Arroz Día 1.

Temperatura Siembra de Arroz Día 2

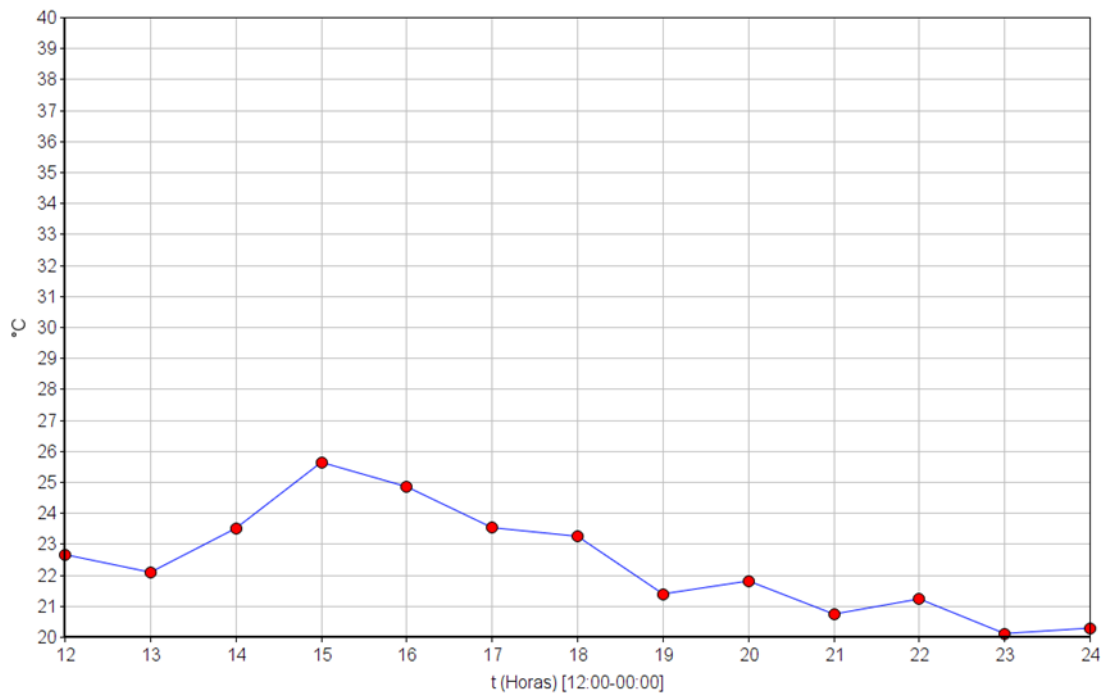


Figura 59. Temperatura Siembra de Arroz Día 2

Temperatura Siembra de Cacao Día 1

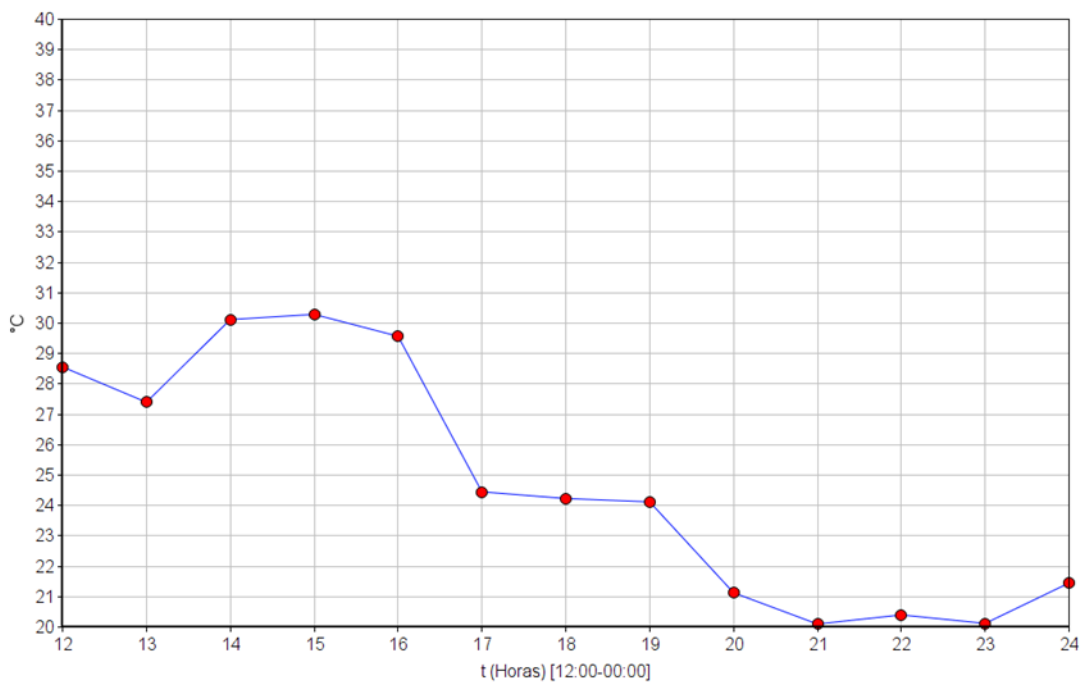


Figura 58. Temperatura Siembra de Cacao Día 1

Temperatura Siembra de Cacao Día 2

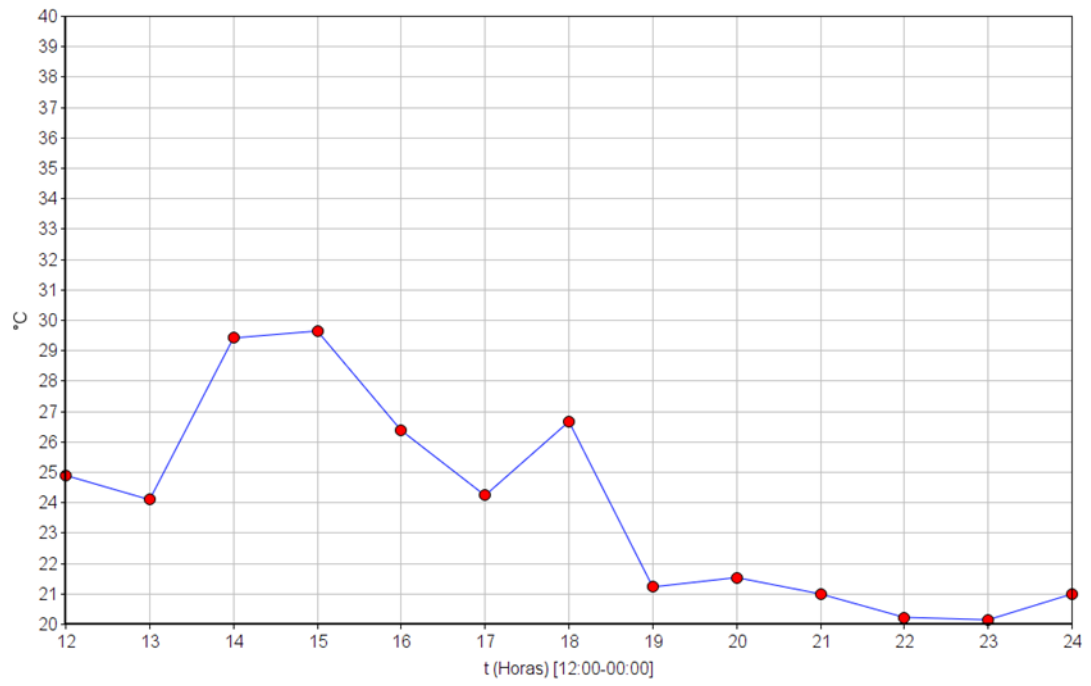


Figura 60. Temperatura de Cacao Día 2

Humedad

Humedad Siembra de Arroz Día 1

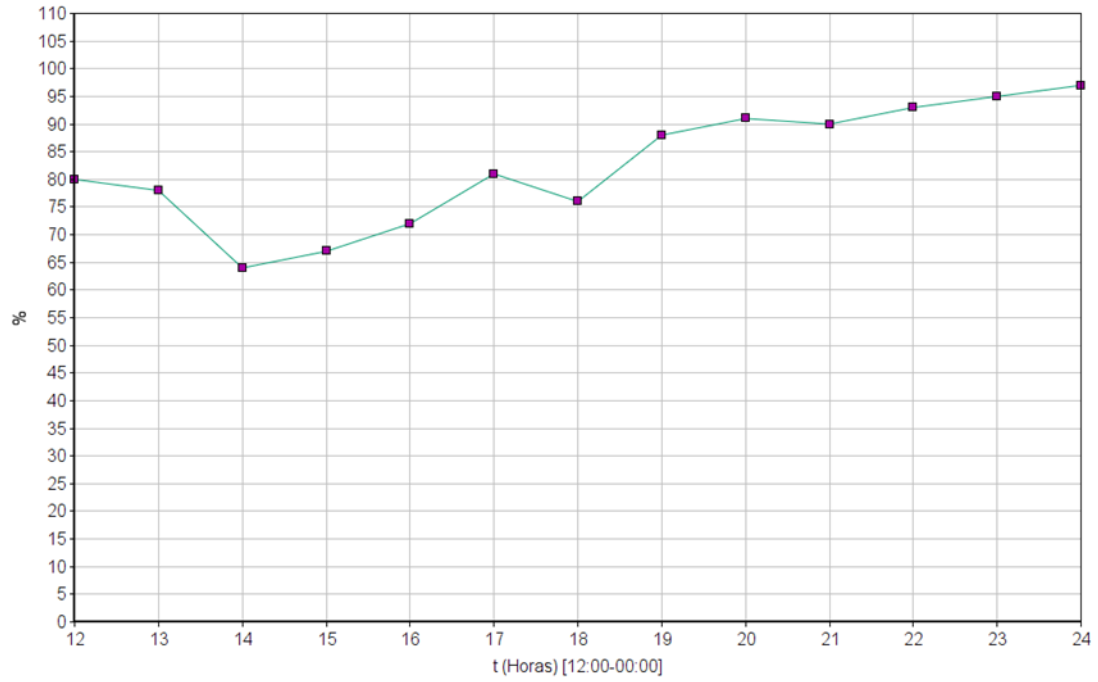


Figura 61. Humedad Siembra de Arroz Día 1

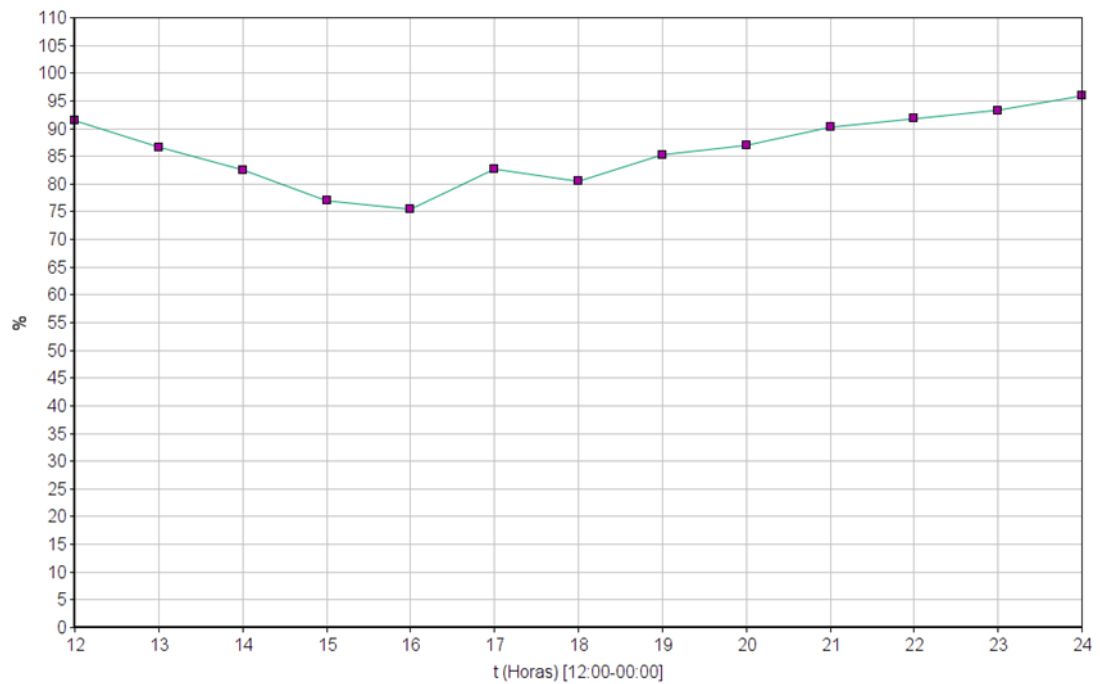


Figura 62. Humedad Siembra de Arroz Día 2

Humedad Siembra de Cacao Día 1

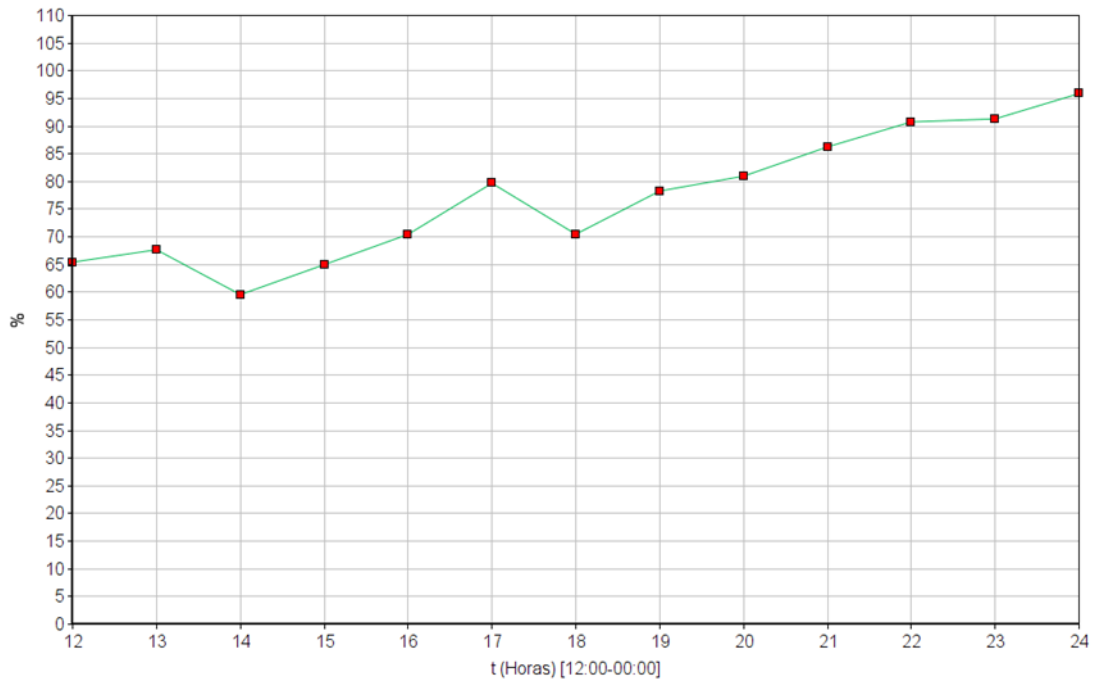


Figura 64. Humedad Siembra de Cacao Día 1

Humedad Siembra de Cacao Día 2

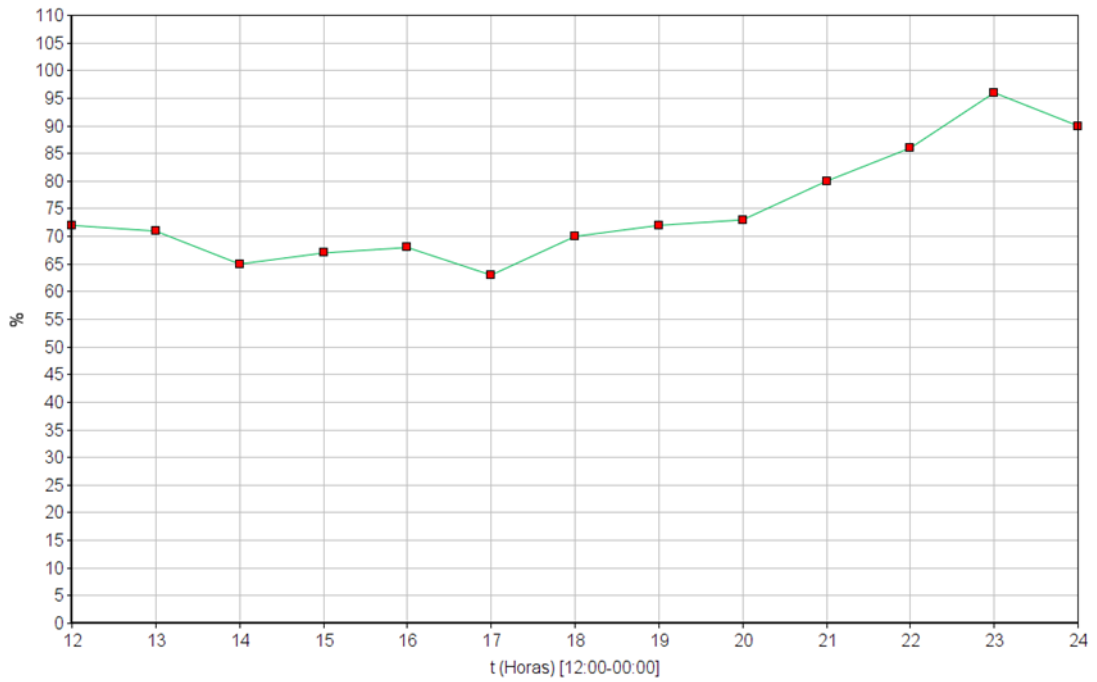


Figura 63. Humedad Siembra de Cacao Día 2

Presión atmosférica

Presión Atmosférica Siembra de Arroz Día 1

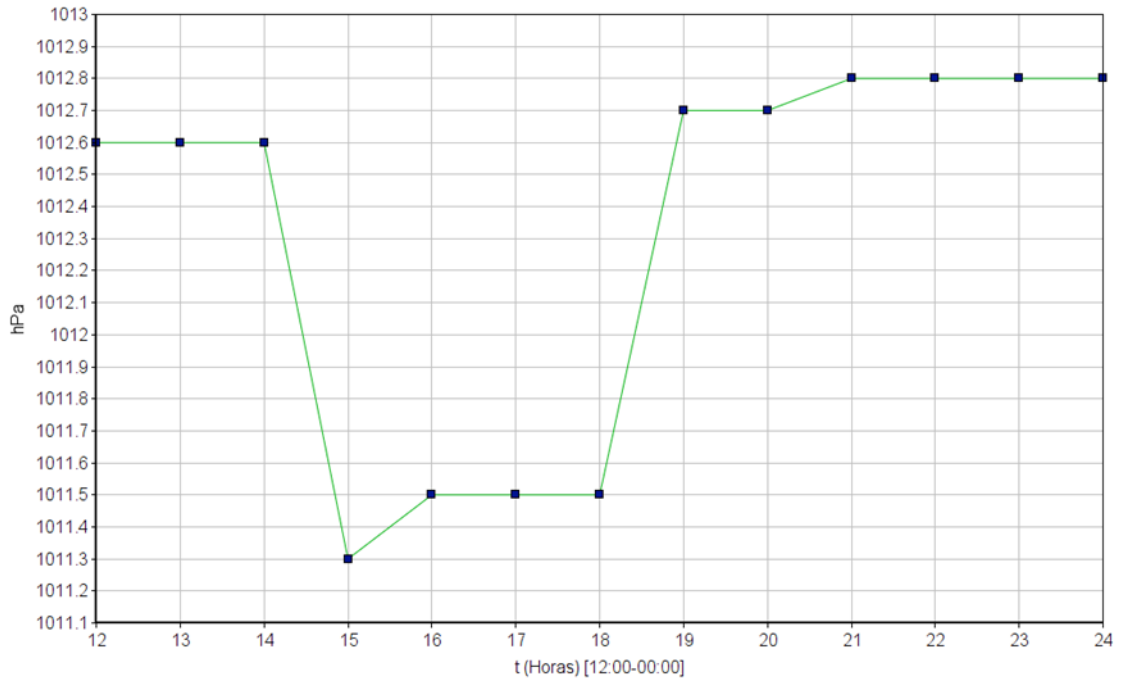


Figura 65. Presion Atmosférica Siembra de Arroz Día 1

Presión Atmosférica Siembra de Arroz Día 2

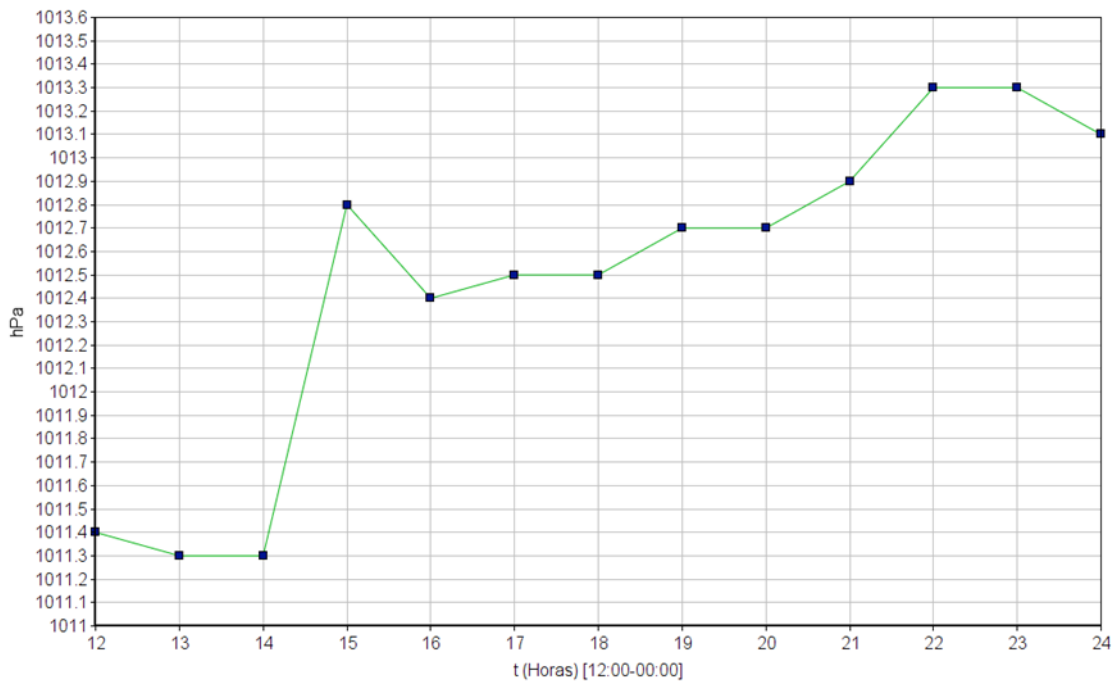


Figura 66. Presión Atmosférica Siembra de Arroz Día 2

Presión Atmosférica Siembra de Cacao Día 1

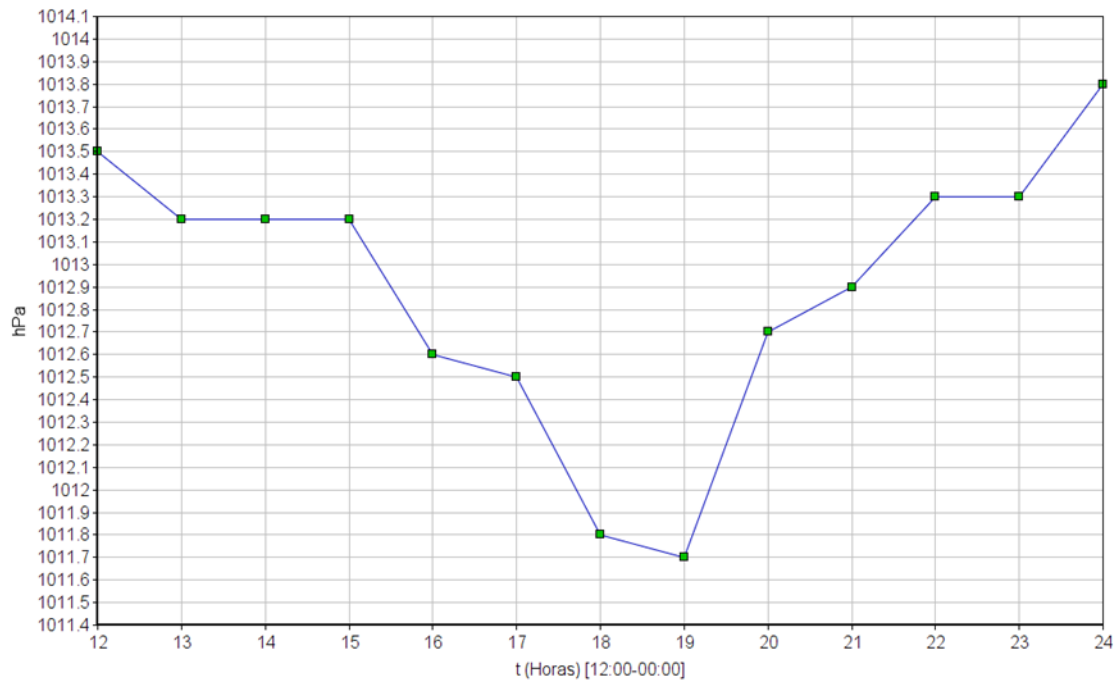


Figura 67. Presión Atmosférica Siembra de Cacao Día 1

Presión Atmosférica Siembra de Cacao Día 2

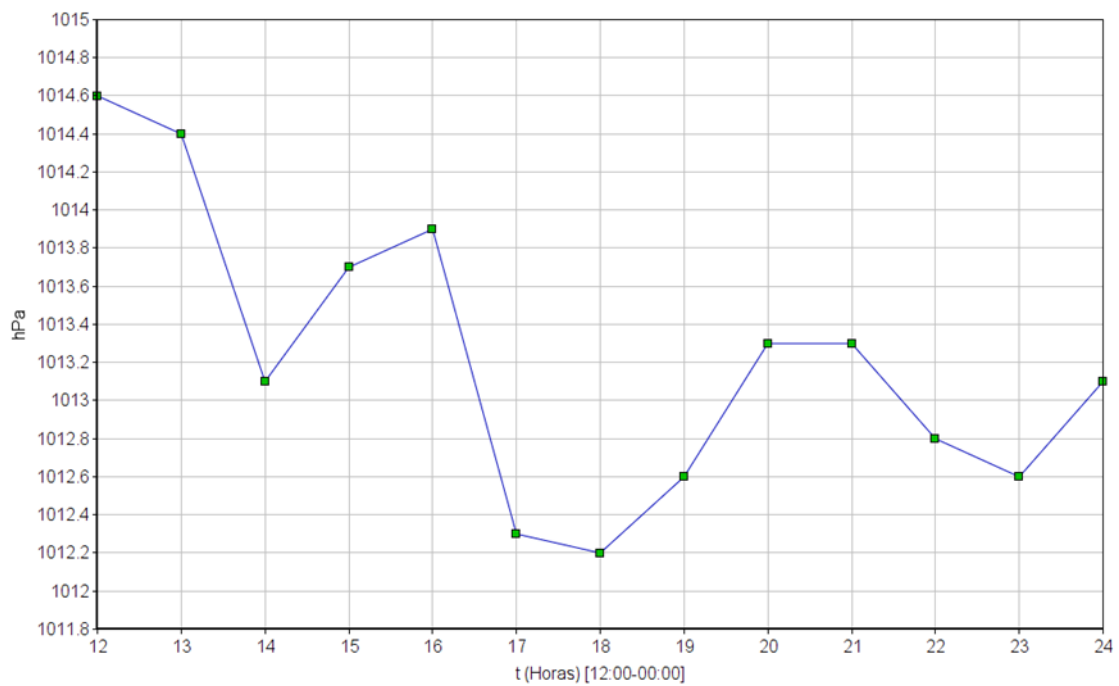


Figura 68. Presión Atmosférica Siembra de Cacao Día 2

Análisis

Experimento 1: Obtención de datos entre 2 ambientes.

Se observó de manera cualitativa y por inspección visual que el sujeto número 2 está en mejor estado de salud, dado que la humedad en la parte externa de un ambiente va ayudar mucho a la sobrevivencia de la planta, el sujeto 1 en cambio estaba en un peor estado de decadencia dado que la humedad en un ambiente cerrado como el ambiente interno de un hogar le dificultó los parámetros de sobrevivencia, y las observaciones de manera cuantitativa, los porcentajes de sujeto 2 en humedad están con mejores lecturas y los parámetros de sujeto 1 están en decadencia.

Experimento 2: Pruebas en campo.

Una vez obtenidos los factores medidos por el prototipo, el profesional agrícola tendrá la información necesaria para justificar las decisiones que van a influir en el terreno para mejorar la producción agrícola. En el caso del experimento anteriormente descrito se podrá determinar si los cultivos necesitan, una zona es adecuada, si la temporada es adecuada, entre otros factores para asegurarse que la producción sea mayor y de mejor calidad.

V. Conclusiones

- El prototipo pudo cumplir con los objetivos propuestos y gracias a la respectiva investigación se descubrieron otros aspectos que se pueden cubrir tales como: enfermedades que afecten a los productos, control del personal en el terreno, cambios significativos en las temporadas del año.
- El prototipo permite a los clientes finales y profesionales agrícolas a tomar mejores

decisiones en el campo de trabajo, ya que los datos obtenidos por el prototipo dan la facilidad de identificar o predecir anomalías que se puedan presentar durante el desarrollo de los productos.

- El usuario y el profesional agrícola podrán realizar cálculos y censos gracias a que el prototipo puede guardar la información de los sensores en una base de datos y estar al pendiente de los cambios ambientales del terreno al poder acceder a las mediciones de los sensores en tiempo real a través de la página web.
- La tecnología implementada permite que el prototipo cubra mayores distancias porque la tecnología LoRa permite comunicaciones entre 3 y 50km logrando constante comunicación entre los dispositivos LoRa.
- Este prototipo puede escalar:
 - Al implementar antenas con más ganancia se pueden cubrir mayores distancias.
 - El bajo consumo de los componentes permite que se puedan utilizar distintas fuentes de energía tales como: paneles solares, baterías, entre otros.
 - Es un sistema manejable, sólo se requiere ubicar los dos dispositivos LoRa donde corresponda y conexión a internet para el receptor (el que sube los datos).
 - Podemos automatizar procesos a través de un panel virtual, incluso, complementar utilizando un panel físico en el campo de trabajo.

VI. Recomendaciones

- El dispositivo debe estar protegido en el campo de trabajo ya que puede sufrir daños por varios factores tales como: agua, polvo, entre otros.

- Durante el desarrollo se debe procurar que la información que se necesita sea verificada ya que se podrían obtener valores imprecisos.
- Para la visualización de datos se debe ser muy preciso al momento de asignar nomenclaturas ya que puede haber conflictos al identificar mensajes.
- Al tener varios usuarios y contraseñas porque se manejan varios servicios, se debe fortalecer la seguridad guardando las credenciales de acceso y no compartirlas.

Referencias bibliográficas

- Agmoderna. (30 de enero de 2021). Automatización: una forma de producir y conservar. Obtenido de <https://www.agmoderna.com.ar/tecnologia-en-el-campo/automatizacion-una-forma-de-producir-y-conservar/#:~:text=En%20el%20marco%20de%20una,malezas%20e%20incluso%20el%20diagn%C3%B3stico>
- Agriculturers. (30 de enero de 2021). Qué es la agricultura inteligente. Obtenido de <https://agriculture.com/que-es-la-agricultura-inteligente/>
- Aguilar., J. (21 de febrero de 2021). HeidiSQL para la gestión de bases de datos grandes. Obtenido de <https://www.jose-aguilar.com/blog/heidisql/>
- Amazon. (31 de mayo de 2021). What is cloud computing? Obtenido de https://aws.amazon.com/es/what-is-cloud-computing/?nc2=h_ql_le_int_cc
- Andrade, N. (27 de mayo de 2021). La importancia de la agricultura en nuestro país. Obtenido de <https://www.utn.edu.ec/ficaya/carreras/agropecuaria/?p=1091#:~:text=Los%20productos%20agr%C3%ADcolas%20como%20banano,las%20importaciones%20se%20reducen%20considerablemente.>
- Atom. (30 de mayo de 2021). Atom. Obtenido de <https://atom.io/>
- Bosch. (30 de mayo de 2021). Humidity sensor BME280. Obtenido de <https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/>
- Características. (20 de enero de 2021). Clima tropical. Obtenido de <https://www.caracteristicas.co/clima-tropical/#ixzz6w6kLx4GE>
- Casanova-Gonzales, M. (25 de agosto de 2020). ¿Qué es LoRa? Una Tecnología LPWAN para IoT. Obtenido de <https://alfaiot.com/blog/ultimas-noticias-2/post/que-es-lora-2>
- Chaparro., M. (2 de marzo de 2021). Instalación Broker MQTT emqx . Obtenido de https://unelectronica.github.io/emqx_docker/
- Concepción, R. (10 de Febrero de 2021). Protocolos de comunicación usados en microcontroladores. Obtenido de <https://www.rjconcepcion.com/podcast/protocolos-de-comunicacion-usados-en-microcontroladores/>

Delamazonas. (15 de enero de 2021). Productos del amazonas. Obtenido de <https://delamazonas.com/productos/>

Delcorp. (10 de diciembre de 2020). Cultivos Región Costa. Obtenido de <http://www.delcorp.com.ec/index.php/cultivos/costa#:~:text=La%20regi%C3%B3n%20litoral%20posee%20una,%2C%20sand%C3%ADa%2C%20mango%20y%20m%20C3%A1s.>

Desarrolloweb. (15 de febrero de 2021). Protocolos de comunicaciones. Obtenido de <https://desarrolloweb.com/articulos/1617.php#:~:text=Un%20protocolo%20de%20comunicaciones%20es,ca%C3%B3tica%20y%20por%20tanto%20imposible.>

Dewesoft. (10 de febrero de 2021). what is a sensor? Obtenido de <https://dewesoft.com/daq/what-is-a-sensor>

Docs Emqx. (2 de marzo de 2021). EMQ x Broker. Obtenido de <https://docs.emqx.io/en/broker/v4.3/>

Ecología Verde. (20 de enero de 2021). Qué es el clima tropical y sus características. Obtenido de <https://www.ecologiaverde.com/que-es-el-clima-tropical-y-sus-caracteristicas-1434.html>

Ecosistema. (20 de enero de 2021). El ecosistema tropical. Obtenido de <https://www.ecosistema.top/el-ecosistema-tropical/>

Elyex. (10 de diciembre de 2020). Productos que se cultivan en la costa de Ecuador. Obtenido de <https://elyex.com/productos-que-se-cultivan-en-la-costa-de-ecuador/>

FAO. (24 de agosto de 2020). Factores Ambientales. Obtenido de [http://www.fao.org/3/x8234s/x8234s08.htm#:~:text=Los%20cultivos%20con%20sufi%20ciento%20agua,agua%20cerrando%20sus%20estomas%20\(p%C3%A1g.](http://www.fao.org/3/x8234s/x8234s08.htm#:~:text=Los%20cultivos%20con%20sufi%20ciento%20agua,agua%20cerrando%20sus%20estomas%20(p%C3%A1g.)

Fernandez, L. (15 de febrero de 2021). Protocolos de redes: la guía completa con todos los protocolos básicos. Obtenido de <https://www.redeszone.net/tutoriales/internet/protocolos-basicos-redes/>

Flatkit. (2020). Obtenido de http://preview.themeforest.net/item/flatkit-app-ui-kit/full_screen_preview/13231484?_ga=2.153669054.76281206.1622502283-752998150.1622501896

Fractal. (15 de febrero de 2021). Las 9 aplicaciones más importantes del Internet de las Cosas. Obtenido de <https://www.fractal.com/es/blog/2018/10/10/9-aplicaciones-importantes-iot>

Geniolandia. (24 de agosto de 2020). Los efectos de la presión atmosférica en las plantas. Obtenido de <https://www.geniolandia.com/13108647/los-efectos-de-la-presion-atmosferica-en-las-plantas>

Grupomarcservice. (10 de febrero de 2021). Sensores aplicaciones y electrodomésticos. Obtenido de <https://grupomarcservice.com/sensores-aplicaciones-y-electrodomesticos/>

Gtush. (23 de enero de 2021). Amazonas. Obtenido de <https://www.gtush.com/amazonas/>

Heidisql. (21 de Febrero de 2021). What's this? Obtenido de <https://www.heidisql.com/>

Heltec. (2020). Heltec. Obtenido de <https://heltec.org/project/wifi-lora-32>

Hetpro. (7 de febrero de 2021). Microcontrolador que es y para que sirve. Obtenido de <https://hetpro-store.com/TUTORIALES/microcontrolador/>

Ingeniería Mecafenix. (10 de febrero de 2021). tipo de sensores y sus características. Obtenido de <https://www.ingmecafenix.com/automatizacion/sensores/>

Latham, M. C. (27 de mayo de 2021). nutrición humana en el mundo en desarrollo. Obtenido de <http://www.fao.org/3/w0073s/w0073s06.htm>

Lifeder. (15 de diciembre de 2020). Los 12 productos de la Sierra más comunes. Obtenido de <https://www.lifeder.com/productos-sierra-ecuatoriana/>

Lifeder. (15 de enero de 2021). 10 productos del la amazonía ecuatoriana. Obtenido de <https://www.lifeder.com/productos-amazonia-ecuatoriana/>

Llamas, L. (24 de agosto de 2020). ¿Qué es MQTT? Su Importancia Como Protocolo IoT. Obtenido de <https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/>

M. Castro. (23 de enero de 2021). Región Sierra. Obtenido de <https://www.lifeder.com/region-interandina-ecuador/>

Mastermoviles. (15 de febrero de 2021). Protocolos de comunicación en red. Obtenido de <https://mastermoviles.gitbook.io/tecnologias2/protocolos-de-comunicacion-en-red>

Microcontroladoresesv. (7 de febrero de 2021). Los microcontroladores de hoy en día. Obtenido de <https://microcontroladoresesv.wordpress.com/los-microcontroladores-de-hoy-en-dia/>

Navarra. (31 de octubre de 2020). Elementos y factores del clima. Obtenido de <http://meteo.navarra.es/definiciones/elementosFactores.cfm#A>

Naylampmechatronics. (24 de agosto de 2020). Placa de desarrollo para ESP32. Obtenido de <https://naylampmechatronics.com/espressif-esp/384-placa-de-desarrollo-para-esp32-nodemcu-32.html>

Naylampmechatronics. (30 de mayo de 2021). TUTORIAL SENSOR DIGITAL DE TEMPERATURA DS18B20. Obtenido de https://naylampmechatronics.com/blog/46_tutorial-sensor-digital-de-temperatura-ds18b20.html

Nutricontrol. (24 de agosto de 2020). la humedad relativa en invernadero. Obtenido de <https://nutricontrol.com/es/la-humedad-relativa-en-%20invernadero/#:~:text=Dependiendo%20de%20la%20especie%2C%20los,el%20r end%20imiento%20de%20los%20cultivos>

Olaya, V. (21 de febrero de 2021). Sistemas de Información Geográfica. Obtenido de https://volaya.github.io/libro-sig/chapters/Bases_datos.html

Plagron. (24 de agosto de 2020). Humedad. Obtenido de <https://www.plagron.com/es/temas/humedad>

Platformio. (30 de mayo de 2021). The most loved IDE solution for Microsoft Visual Studio Code. Obtenido de <https://platformio.org/>

Putty. (31 de mayo de 2021). Download PuTTY. Obtenido de <https://www.putty.org/>

Recursostic. (10 de febrero de 2021). Sensores. Obtenido de http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena11/4quincena11_contenidos_3a.htm

Redhat. (15 de febrero de 2021). EL INTERNET DE LAS COSAS. Obtenido de <https://www.redhat.com/es/topics/internet-of-things/what-is-iot>

S. Quiroga-Gómez, A. I. (31 de octubre de 2020). Infoagro. Obtenido de https://www.infoagro.com/hortalizas/relacion_clima_cultivo.htm

- Sap. (15 de febrero de 2021). ¿Qué es internet de las cosas (IoT)? Obtenido de <https://www.sap.com/latinamerica/insights/what-is-iot-internet-of-things.html>
- Sherlin. (2 de febrero de 2021). Qué es un microcontrolador? Obtenido de <http://sherlin.xbot.es/microcontroladores/introduccion-a-los-microcontroladores/que-es-un-microcontrolador>
- Sierra Ecuador. (23 de enero de 2021). La Sierra Ecuatoriana. Obtenido de <https://sierraecuador.wordpress.com/clima/#:~:text=La%20temperatura%20promedio%20est%C3%A1%20entre%2012%20y%2020%20%C2%BAC.&text=Comprende>
- Smelpro. (2 de febrero de 2021). Qué es un microcontrolador? Obtenido de <https://smelpro.com/blog/que-es-un-microcontrolador/>
- Tapia, J. (10 de febrero de 2021). Comunicación en microcontroladores PIC. Obtenido de <https://johntapia.wordpress.com/2008/11/21/comunicacion-en-microcontroladores-pic/>
- Valdés., D. P. (21 de febrero de 2021). ¿Qué son las bases de datos? Obtenido de <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>
- Valerie Lampkin, W. T.-L. (2012). Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry. Estados Unidos: Vervante.
- Vestacp. (31 de mayo de 2021). Simple & Clever Hosting Control Panel. Obtenido de <https://vestacp.com/>
- Vestertraining. (15 de febrero de 2021). Lo que necesitas saber sobre los protocolos de red. Obtenido de <https://vestertraining.com/blog/protocolos-red/>

Anexos

Diseño placa

Tx

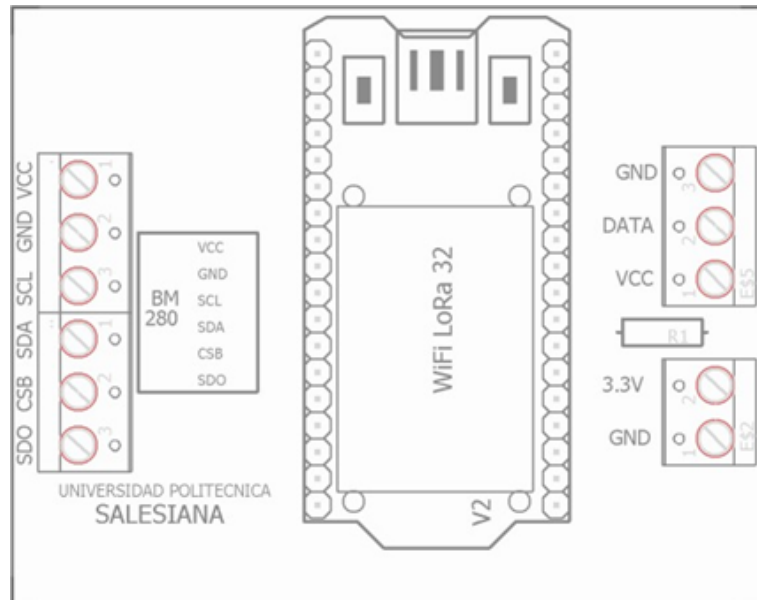


Figura 69, Placa tx

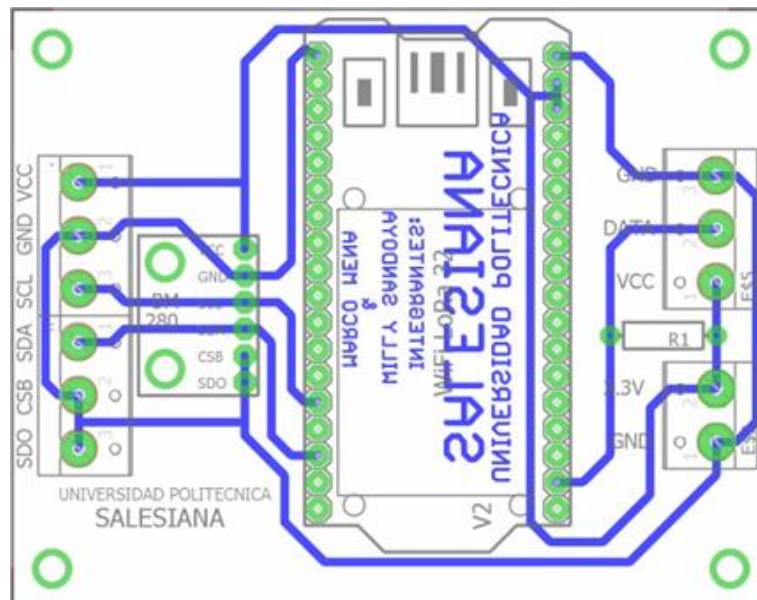


Figura 70, Placa Tx conductors

Rx

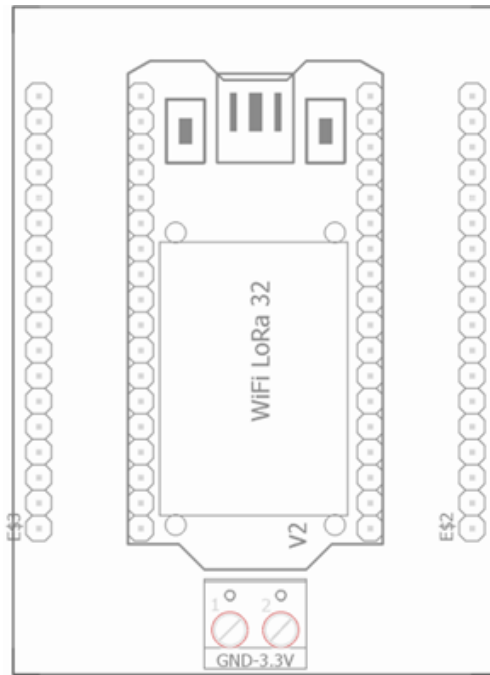


Figura 71, Placa Rx

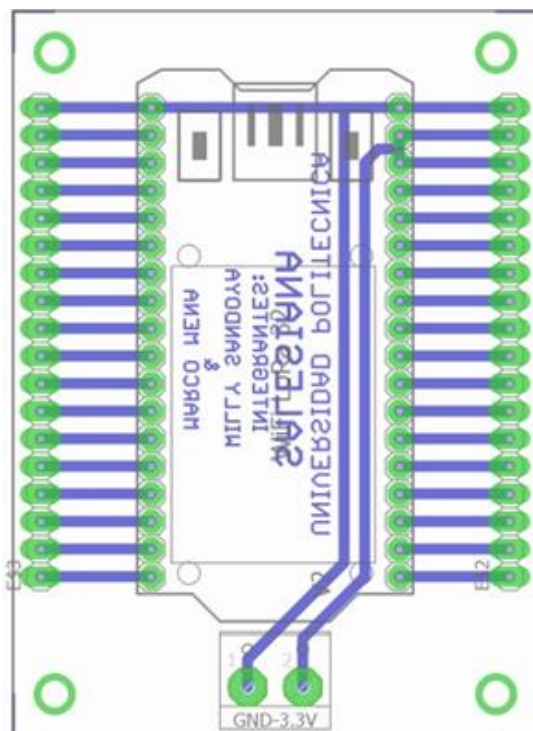


Figura 72, Placa Rx conductores

Placa terminada



Figura 73, Placa terminada

Tx

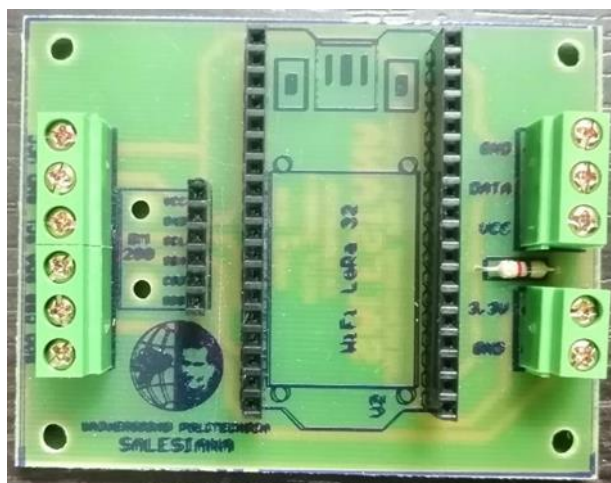


Figura 74, placa terminada tx



Figura 75, Parte posterior placa tx

Rx

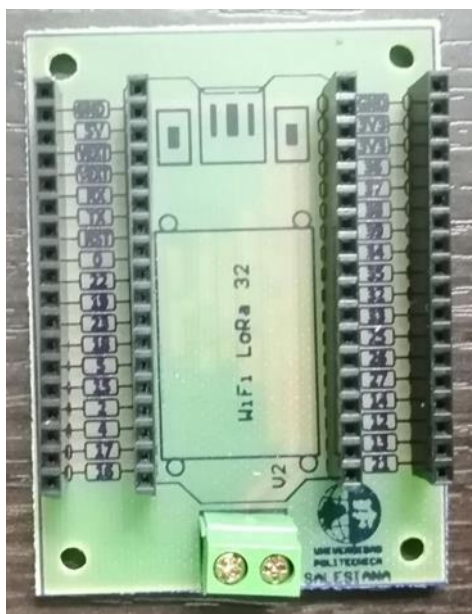


Figura 76, Placa rx terminada



Figura 77, parte posterior placa rx

Placas con dispositivos

Tx



Figura 78, Placa tx con microcontrolador y sensores

Rx

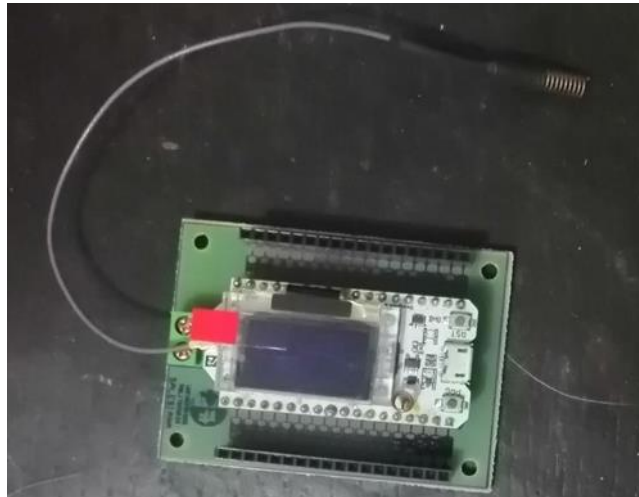


Figura 79, Placa rx con microcontrolador

Proyecto en Funcionamiento

Tx

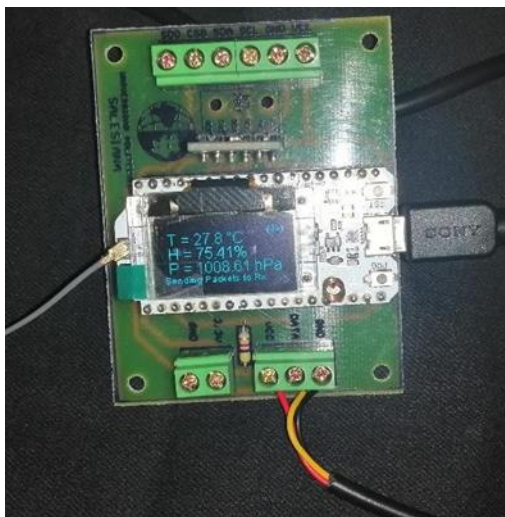


Figura 80, Placa tx en funcionamiento

Rx



Figura 81, Placa rx en funcionamiento

Código Tx

```
//LIBRERIAS
```

```
#include <Arduino.h> #include <WiFi.h> #include <stdlib.h> #include <OneWire.h>
```

```
#include <DallasTemperature.h> #include <AsyncDelay.h> #include <SoftWire.h>
```

```
#include <Adafruit_Sensor.h> #include <Adafruit_BME280.h> #include "heltec.h"
```

```
#include "images.h"
```

```
//DEFINIR BANDA/PINES
```

```
#define BAND 433E6 #define SDA 4
```

```
#define SCL 15
```

```
//DECLARACIÓN DE VARIABLES
```

```
const int pinDato = 13; unsigned int counter = 0; String rssi = "RSSI --"; String packSize = "--";  
String packet;
```

```
float tempBME;//NUEVO float hudBME;
```

```
float pressBME; String MensajeBME;
```

```
//ACTIVAR LIBRERIAS
```

```
SoftWire sw(SDA, SCL);
```

```
OneWire oneWireObjeto(pinDato); DallasTemperature sensorDS18B20(&oneWireObjeto);  
Adafruit_BME280 bme;
```



```
TwoWire I2Cone = TwoWire(1);//0
```

```
//LOGO
```

```
void logo()
```

```
{
```

```
Heltec.display->clear(); Heltec.display->
```

```
drawXbm(0,5,logo_width,logo_height,logo_bits); Heltec.display->display();
```

```
}
```

```
//ACTIVAR BME280
```

```
void startBME(){ bool bme_status;
```

```
bme_status = bme.begin(0x76); if (!bme_status) {
```

```
Heltec.display->clear();
```

```
Heltec.display->drawString( 0, 0, "No valid BME280 found");
```

```
Heltec.display->drawString( 0, 12, "Please check wiring!");
```

```
Heltec.display->display();
```

```
Serial.println("Could not find a valid BME280 sensor, check wiring!");
```

```
}
```

```
}
```

```
//CAPTURAR LECTURAS BME
```

```
void getReadings(){ bme.setSampling(Adafruit_BME280::MODE_FORCED,
```

```
Adafruit_BME280::SAMPLING_X16,
```

```
Adafruit_BME280::SAMPLING_X1,
```

```
Adafruit_BME280::SAMPLING_X1,
```

```
Adafruit_BME280::FILTER_X16, Adafruit_BME280::STANDBY_MS_0_5 );
```

```
bme.takeForcedMeasurement(); tempBME = bme.readTemperature(); hudBME =  
bme.readHumidity();
```

```
pressBME = bme.readPressure() / 100.0F;
```

```
}
```

```
//SETUP
```

```
void setup()
```

```
{
```

```
Serial.begin(115200);
```

```
Heltec.begin(true /*DisplayEnable Enable*/, true
/*Heltec.Heltec.Heltec.LoRa Disable*/, true /*Serial Enable*/, true /*PABOOST Enable*/,
BAND /*long BAND*/);

Heltec.display->init();

Heltec.display->flipScreenVertically(); Heltec.display->setFont(ArialMT_Plain_24); logo();

delay(2000); Heltec.display->clear();

Heltec.display->drawString(0, 0, "Heltec.LoRa");

Heltec.display->drawString(0, 25, "Initialized!"); Heltec.display->display();

delay(1000); startBME(); sensorDS18B20.begin();}
```

```
//LOOP
```

```
void loop()
```

```
{
```

```
digitalWrite(16, LOW); // set GPIO16 low to reset OLED delay(1);//50
```

```
digitalWrite(16, HIGH); delay(1); Heltec.display->init();
```

```
getReadings();
```

```
sensorDS18B20.requestTemperatures();
```

```
float temp=sensorDS18B20.getTempCByIndex(0);
```

```
char tempstring[20];//20 dtostrf(temp,3,1,tempstring);//temp
```

```
String temperatura(tempstring);
```

```
Heltec.display->clear();
```

```
Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT);
```

```
Heltec.display->setFont(ArialMT_Plain_16);//15 o 24 Heltec.display->drawString(5, 5,"T = " +  
String(temperatura)
```

```
+ " °C");
```

```
Heltec.display->drawString(5, 20,"H = " + String(hudBME) + "%");
```

```
Heltec.display->drawString(5, 35,"P = " + String(pressBME) + " hPa");
```

```
Heltec.display->setFont(ArialMT_Plain_10);//15 o 24 Heltec.display->drawString(0,  
50,"Sending Packets to Rx");
```

```
Heltec.display->drawString(110, 0,"Tx"); Heltec.display->display();
```

```
//CREAR PAQUETE
```

```
LoRa.beginPacket(); LoRa.setTxPower(14,RF_PACONFIG_PASELECT_PABOOST);
```

```
LoRa.print("TempBME:! "); LoRa.print(temperatura);
```

```
LoRa.print(" HudBME:& "); //envía la palabra cont("\n: Cont:* ")
```

```
LoRa.print(hudBME);      LoRa.print("PressBME:%      ");      LoRa.print(pressBME);
```

```
LoRa.println(MensajeBME); LoRa.endPacket();
```

```
//FIN PAQUETE
```

```
digitalWrite(LED, HIGH); delay(2500); digitalWrite(LED, LOW); delay(2500);
```

```
}
```

Código Rx

```
//LIBRERIAS

#include <Arduino.h> #include <WiFi.h> #include <PubSubClient.h> #include <stdlib.h>
#include <OneWire.h>

#include <DallasTemperature.h> #include "heltec.h"

#include "images.h"

//VARIABLES: WIFI & MQTT

const char* ssid      = "xxxxxxxxxxxxx"; const char* password = "xxxxxxxxxxxxx";

const char *mqtt_server = "xxxxxxxxxxxxx"; const int mqtt_port = 1883;
const char *mqtt_user = "xxxxxxxxxxxxx"; const char *mqtt_pass = "xxxxxxxxxxxxx";

//ACTIVAR LIBRERIAS WIFI & MQTT
WiFiClient espClient; PubSubClient client(espClient);

long lastMsg = 0; char msg[25];

//VARIABLES LoRa

#define BAND 433E6 String rssi = "RSSI --"; String packSize = "--"; String packet;

String MensajeRecibido; String Tempw;

String Contw; String LoRaDataw; String tempBME; String hudBME; String pressBME; String
counter; String temperatura;
```

```

//LOGO

void logo(){ Heltec.display->clear();

Heltec.display->

>drawXbm(0,5,logo_width,logo_height,logo_bits); Heltec.display->display();

}

//MOSTRAR LO QUE SE RECIBE

void LoRaData(){ Heltec.display->clear();

Heltec.display->setTextAlignment(TEXT_ALIGN_LEFT); Heltec.display->
>setFont(ArialMT_Plain_10); Heltec.display->setFont(ArialMT_Plain_16); Heltec.display->
>drawString(5, 5,"T = " + Tempw + " °C"); Heltec.display->drawString(5, 20,"H = " + hudBME
+ "%");

Heltec.display->drawString(5, 35,"P = " + pressBME + " hPa");

Heltec.display->setFont(ArialMT_Plain_10);//15 o 24 Heltec.display->drawString(0,
50,"Receiving Packets From

Tx");

Heltec.display->drawString(110, 0,"(Rx)"); Heltec.display->display();

}

//RECIBIR PAQUETE

void cbk(int packetSize) { packet ="";

packSize = String(packetSize,DEC);

for (int i = 0; i < packetSize; i++) { packet += (char) LoRa.read(); }

```

```

rssi = "RSSI " + String(LoRa.packetRssi(), DEC) ;
MensajeRecibido=packet;//LoRa.readString(); if (MensajeRecibido!=""){

int data1=packet.indexOf('!'); int data2=packet.indexOf('&'); int data3=packet.indexOf('%'); int
data4=packet.indexOf('#');

Tempw=packet.substring(data1+1,data1+7); hudBME=packet.substring(data2+1,data2+7);
pressBME=packet.substring(data3+1,data3+8);
tempBME=packet.substring(data4+1,data4+8); LoRaData();

}

else{

Tempw=""; hudBME=""; pressBME=""; tempBME=""; LoRaData();

}

Serial.println(MensajeRecibido);

}

void setup_wifi();

void callback(char* topic, byte* payload, unsigned int length);

void reconnect();

//SETUP

void setup() {

Heltec.begin(true /*DisplayEnable Enable*/, true

/*Heltec.Heltec.Heltec.LoRa Disable*/, true /*Serial Enable*/, true /*PABOOST Enable*/,
BAND /*long BAND*/);

```



```

Heltec.display->init();

Heltec.display->flipScreenVertically(); Heltec.display->setFont(ArialMT_Plain_10); logo();

delay(1500); Heltec.display->clear();

Heltec.display->setFont(ArialMT_Plain_16); Heltec.display->drawString(0, 0, "Heltec.LoRa");
Heltec.display->drawString(0, 15, "Initial success!"); Heltec.display->drawString(0, 30, "Wait
for Tx data..."); Heltec.display->display();

delay(1000); LoRa.receive(); Serial.begin(115200);

pinMode(BUILTIN_LED, OUTPUT);

Serial.begin(115200); randomSeed(micros()); setup_wifi();

client.setServer(mqtt_server, mqtt_port); client.setCallback(callback);
}

//LOOP

void loop(){

if (!client.connected()) {
reconnect();
}

client.loop();

```

```

long now = millis();

if (now - lastMsg > 500){ lastMsg = now;

//temperatura++;

temp2++; volts++;

Serial.println("RSSI "); Serial.println(rssi);

String to_send = String(Tempw) + "," +
String(hudBME) + "," + String(pressBME); //Tempw, Contw to_send.toCharArray(msg, 25);
Serial.print("Publicamos mensaje -> "); Serial.println(msg); client.publish("values", msg);

Serial.println(String(packet)); Serial.println(Tempw); Serial.println(hudBME);
Serial.println(pressBME);

}

int packetSize = LoRa.parsePacket(); if (packetSize) { cbk(packetSize); } delay(10);}

//CONEXION WIFI

void setup_wifi(){

delay(10); Serial.println();

Serial.print("Conectando a "); Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) { delay(500);

Serial.print(".");

}

```

```
Serial.println(""); Serial.println("Conectado a red WiFi!"); Serial.println("Dirección IP: ");  
Serial.println(WiFi.localIP());  
}
```

```
//CALLBACK
```

```
void callback(char* topic, byte* payload, unsigned int length){
```

```
String incoming = "";
```

```
Serial.print("Mensaje recibido desde -> "); Serial.print(topic);
```

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

```
for (int i = 0; i < length; i++) { incoming += (char)payload[i];
```

```
}
```

```
incoming.trim();
```

```
Serial.println("Mensaje -> " + incoming);
```

```
if ( incoming == "on") { digitalWrite(BUILTIN_LED, HIGH);
```

```
} else {
```

```
digitalWrite(BUILTIN_LED, LOW);
```

```
}
```

```
}
```

```
//RECONECTAR

void reconnect() {

while (!client.connected()) { Serial.print("Intentando conexión Mqtt...");

String clientId = "esp32_";

clientId += String(random(0xffff), HEX);

if (client.connect(clientId.c_str(),mqtt_user,mqtt_pass)) {

Serial.println("Conectado!");

client.subscribe("led1"); client.subscribe("led2");

} else {

Serial.print("falló :( con error -> ");

Serial.print(client.state()); Serial.println(" Intentamos de nuevo en 5

segundos");

delay(5000);

}

}

}
```