

**Construcción de un Prototipo Electrónico Para el Monitoreo Climatológico en Cultivos
de Aguacate en el Municipio de Tona, Basado en Tecnología IoT.**

Julián Darío Luna Patiño

Trabajo de grado para optar al título de Ingeniero de Telecomunicaciones

Director:

Ing. Faver Adrián Amorocho

Universidad Nacional Abierta y a Distancia - UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería

Pregrado en Ingeniería de Telecomunicaciones

Santander

2020

Dedicatoria

Dedico este proyecto a mis padres, mi hermano, mi hija y mi abuelo, este logro es resultado de un esfuerzo colectivo y del acompañamiento de cada uno en este proceso que he decidido emprender y que hace parte de la formación de un mejor futuro.

Agradecimientos

Agradezco a mis padres, quien con esfuerzo me han dado la posibilidad de formarme y sacar adelante mi carrera profesional, de la misma manera agradezco a mi director de proyecto de grado, por su interés en buscar siempre la forma de llegar a los resultados esperados y la manera de consolidar el proyecto con el objetivo de que pueda llegar a alcanzar el objetivo.

Resumen

El proceso de monitoreo de un cultivo de aguacate requiere de atención especial, ya que los aspectos climáticos son claves para tener éxito; bien sea para alcanzar la producción esperada, como para poder controlar plagas, obtener estándares de calidad o tener una trazabilidad, a partir de estos aspectos se pretende generar valor agregado con la implementación de este proyecto que contribuye en la solución frente a problemas como: el desconocimiento y atención temprana de enfermedades en el cultivo, ahogamiento de árboles o por el contrario, la deshidratación, y en general falta de trazabilidad de los parámetros ambientales en los cuales se dio el cultivo.

Estas necesidades se buscan resolver a partir del presente proyecto aplicado que tiene por objetivo construir un prototipo electrónico para el monitoreo climatológico aplicado en cultivos de aguacate en el municipio de Tona Santander, basado en tecnología IoT. Las variables elegidas a monitorear son temperatura, humedad relativa, radiación solar y presión barométrica en cultivos de aguacate ubicados en el municipio de Tona, Santander.

Además de la integración de los sensores climatológicos y el monitoreo de las variables ambientales, un aporte que se busca destacar en este trabajo es la implementación de tecnología LoRA para optimizar la toma de datos de los sensores y la centralización de los mismos en un dispositivo con conexión a internet. Entendiendo que la conectividad a internet en las zonas rurales es limitada, la tecnología LoRA es un tipo de red LPWAN (Low-Power Wide Area Network) que surge como una opción innovadora y aún poco investigada para abrir campo a la implementación de soluciones IoT en el sector del agro.

Palabras clave. Monitoreo, lora, estación meteorológica, iot, agro 4.0, internet de las cosas.

Abstract

The process of monitoring an avocado crop requires special attention, since climatic aspects are key to being successful; either to achieve the expected production, or to be able to control pests, obtain quality standards or have traceability, from these aspects it is intended to generate added value with the implementation of this project that contributes to the solution to problems such as: ignorance and early attention to diseases in the crop, drowning of trees or, on the contrary, dehydration, and in general lack of traceability of the environmental parameters in which the crop occurred.

These needs are sought to be resolved from the present applied project that aims to build an electronic prototype for the climatological monitoring applied in avocado crops in the municipality of Tona Santander, based on IoT technology. The variables chosen to monitor are temperature, relative humidity, solar radiation and barometric pressure in avocado crops located in the municipality of Tona, Santander.

In addition to the integration of the weather sensors and the monitoring of environmental variables, a contribution that is sought to be highlighted in this work is the implementation of LoRA technology to optimize the data collection of the sensors and the centralization of the same in a device with Internet connection. Understanding that internet connectivity in rural areas is limited, LoRA technology is a type of LPWAN (Low-Power Wide Area Network) network that emerges as an innovative and still little-researched option to open up the field for the implementation of IoT solutions in the agricultural sector.

The project to date is in the consolidation phase of results, it shows in the first tests a communication capacity of 300 meters on the land where it has been tested.

Keywords. Monitoring, lora, weather station, iot, agro 4.0, internet of things.

Contenido

	Pág.
1. Introducción	14
2. Planteamiento del Problema	16
3. Objetivos	19
3.1 Objetivo General	19
3.2 Objetivos específicos	19
4. Justificación	20
5. Marco Referencial.....	23
5.1 Proceso de producción del aguacate	24
5.1.1 Preparación del suelo	26
5.1.2 Eliminación de hierbas malas	26
5.1.3 Poda.....	27
5.1.4 Propagación.....	28
5.1.5 Recolección.....	29
5.1.6 Marco de plantación.....	31
5.1.7 Riego	31
5.1.8 Fertilización	32
5.1.9 Plagas	33
5.1.10 Enfermedades.....	36
5.1.11 Cosecha	39
5.1.11.1 Calidad	39
5.1.11.2 Temperatura óptima	40
5.1.11.3 Humedad relativa	40
5.1.11.4 Tasa de respiración	40
5.1.11.5 Tasa de producción de etileno.....	41
5.2 Meteorología.....	41

5.2.1 Estación meteorológica.....	42
5.2.2 Variables ambientales	43
5.3 Marco conceptual.....	44
5.3.1 Internet de las cosas – IoT	44
5.1.2 Aplicación.....	44
5.1.3 Sensórica.....	45
5.1.4 Cloud Computing.....	45
5.1.5 Procesamiento y almacenamiento de datos.....	46
5.1.6 Hypertext Transfer Protocol (HTTP).....	46
5.1.7 MQ Telemetry Transport (MQTT)	46
5.1.8 Microcontrolador	47
5.1.9 Sensor de temperatura.....	47
5.1.10 Sensor de humedad	47
5.1.11 Sensor de lluvia.....	48
5.1.12 JavaScript Object Notation (JSON)	48
6. Metodología	49
6.1 Selección de los sensores	50
6.2 Adquisición de los sensores para el monitoreo de las variables climatológicas	51
6.3 Implementación e integración de los sensores adquiridos.....	51
6.4 Implementación del protocolo de comunicación entre los dispositivos	51
6.5 Construcción del prototipo hardware.....	51
6.6 Medición de las variables monitoreadas y análisis estadístico de las variables monitoreadas.....	52
6.7 Consolidación del informe final.....	52
7. Resultados	53
7.1 Selección de los sensores	53
7.2 Adquisición de los sensores para el monitoreo de las variables climatológicas	64

7.3	Implementación e integración de los sensores adquiridos	65
7.3.1	Sensor de temperatura y humedad relativa – DTH11	65
7.3.2	Sensor de humedad en tierra – YI100.....	66
7.3.3	Sensor de lluvia – YC69	68
7.3.4	Sensor de intensidad lumínica ambiental.....	69
7.3.5	Sensor de presión barométrica - BMP280	71
7.3.6	Módulo lector de microSD.....	73
7.3.7	Panel solar.....	76
7.3.8	Módulo de comunicación LoRA02.....	78
7.4	Implementación del protocolo de comunicación entre los dispositivos	81
7.4.1	LoRa.....	81
7.4.2	MQTT	83
7.5	Construcción del prototipo hardware.....	85
7.5.1	Dispositivo nodo	85
7.5.2	Dispositivo Base	87
7.6	Medición de las variables monitoreadas	89
7.7	Consolidación del informe final.....	101
8.	Conclusiones	104
9.	Recomendaciones	105
10.	Referencias Bibliográficas	106

Lista de tablas

	Pág.
Tabla 1 Variedades de aguacate aptas para el cultivo en las diferentes altitudes.	26
Tabla 2 Tasa de respiración del cultivo de aguacate.....	41
Tabla 3 Presupuesto general	102

Lista de figuras

	Pág.
Figura 1 Parque principal de Tona, Santander.....	17
Figura 2 Ubicación de la finca y la distancia de los dispositivos dónde en el marcado rojo indica la ubicación del dispositivo nodo.....	18
Figura 3 Aguacate producido en una de las fincas de Tona.	24
Figura 4 Árbol de aguacate evidenciando los primeros frutos.	25
Figura 5 Marchitamiento del aguacate.....	27
Figura 6 Árbol de aguacate joven podado	27
Figura 7 Ejemplo de un injerto de aguacate.....	29
Figura 8 Recolección de aguacate.....	30
Figura 9 Recolección de aguacate.....	30
Figura 10 Marco de plantación	31
Figura 11 Sistema de riego del aguacate.....	32
Figura 12 Proceso de fertilización de árboles de aguacate	33
Figura 13 <i>Copturomimus perseae</i> Gunthe	34
Figura 14 <i>Heilipus pittieri</i> (Barber)	35
Figura 15 Trip del aguacate <i>Heliothrips haemorrhoidalis</i>	35
Figura 16 Daño causado por Trip del aguacate <i>Heliothrips haemorrhoidalis</i>	36
Figura 17 <i>Stenomoma catenifer</i>	36
Figura 18 Evidencia del daño causado por pudrición de la raíz o marchitez del aguacate.....	37
Figura 19 Presencia de cercospora.....	38
Figura 20 Daño causado por polvillo o muldiu <i>oidium sp.</i>	39
Figura 21 Aguacate de calidad.....	40
Figura 22 Estación metereológica.....	42

Figura 23 Diagrama de la solución que se aborda con el proyecto.	53
Figura 24 Sensor DHT-11.....	54
Figura 25 Sensor de humedad en tierra.....	55
Figura 26 Sensor de lluvia YC69.....	56
Figura 27 Módulo de comunicación LoRA 02	57
Figura 28 Módulo de carga micro-USB para batería de litio.....	59
Figura 29 Módulo ESP8266 - ESP12F	60
Figura 30 Sensor BMP280.....	61
Figura 31 Sensor de intensidad lumínica ambiental	62
Figura 32 Módulo ESP32.....	63
Figura 33 Módulo lector de microSD	64
Figura 34 Diagrama de conexión del sensor DHT11.....	65
Figura 35 Código sensado de humedad y temperatura relativa utilizando el sensor DHT11..	66
Figura 36 Diagrama de conexión del sensor YI100.....	67
Figura 37 Código de lectura del sensor YI100 Arduino	67
Figura 38 Diagrama de conexión del sensor YC69	68
Figura 39 Código para realizar la lectura del sensor de lluvia YC69	69
Figura 40 Esquemático de la conexión del sensor	70
Figura 41 Código correspondiente a la prueba del sensor de intensidad lumínica ambiental..	70
Figura 42 Esquemático de conexión del sensor BMP280	71
Figura 43 Código de prueba del sensor BMP280	72
Figura 44 Diagrama de conexión del módulo de lectora de microSD	73
Figura 45 Código correspondiente a la lectura de los registros en la microSD.....	74
Figura 46 Código de escritura de registro en microSD.....	75

Figura 47 Diagrama de conexión del panel solar, la batería y el módulo de carga micro-USB	76
Figura 48 Prueba realizada cargando la batería de litio con el panel solar.....	76
Figura 49 Prueba realizada cargando la batería de litio con el panel solar.....	77
Figura 50 Prueba realizada cargando la batería de litio con el panel solar.....	77
Figura 51 Diagrama de conexión LoRa 02.....	78
Figura 52 Código del transmisor LoRa02.....	79
Figura 53 Código del receptor LoRa02.	80
Figura 54 Prueba de comunicación entre módulos LoRa	82
Figura 55 Log de los registros de la comunicación en el dispositivo Base	82
Figura 56 Prueba de envío de datos del dispositivo base a servidor vía MQTT	83
Figura 57 Código de prueba envío de datos a Google Sheet - parte 1.....	84
Figura 58 Código de prueba envío de datos a Google Sheet - parte 2.....	85
Figura 59 Esquemático del circuito correspondiente al dispositivo Nodo	86
Figura 60 Esquemático del circuito correspondiente al dispositivo Base.....	87
Figura 61 Dispositivo Nodo fuera del case.....	89
Figura 62 Dispositivo Nodo en case	90
Figura 63 Panel solar del dispositivo Nodo.	90
Figura 64 Dispositivo Base.....	91
Figura 65 Dispositivo Base en case	91
Figura 66 Datos emitidos por el dispositivo base y almacenados en una hoja de Google Sheet.....	92
Figura 67 Prueba de alcance realizada, llegando a una distancia de 318 m.	92
Figura 68 En una segunda prueba se tenía una expectativa de alcance de 776 m (línea amarilla), y el dispositivo alcanzó una distancia de 348m (línea roja)	93
Figura 69 Tramas recibidas vía MQTT enviadas por el dispositivo Base luego de ser enviadas a este por el dispositivo Nodo vía LoRa	93

Figura 70 Código fuente correspondiente a dispositivo Nodo - parte 1	94
Figura 71 Código fuente correspondiente a dispositivo Nodo - parte 2	94
Figura 72 Código fuente correspondiente a dispositivo Nodo - parte 3	95
Figura 73 Código fuente correspondiente a dispositivo Nodo - parte 4	95
Figura 74 Código fuente correspondiente a dispositivo Base - parte 1	96
Figura 75 Código fuente correspondiente a dispositivo Base - parte 2	96
Figura 76 Código fuente correspondiente a dispositivo Base - parte 3	97
Figura 77 Código fuente correspondiente a dispositivo Base - parte 4	97
Figura 78 Código fuente correspondiente a dispositivo Base - parte 5	98
Figura 79 Código fuente correspondiente a dispositivo Base - parte 6	98
Figura 80 Código fuente correspondiente a dispositivo Base - parte 7	99
Figura 81 Código fuente correspondiente a dispositivo Base - parte 8	99
Figura 82 Código fuente correspondiente a dispositivo Base - parte 9	100
Figura 83 Código fuente correspondiente a dispositivo Base - parte 10	100
Figura 84 Código fuente correspondiente a dispositivo Base - parte 11	101

1. Introducción

La implementación de tecnología dentro de los procesos que se llevan hoy en día en las diversas áreas se ha transformado en un sinónimo de eficiencia, competitividad, crecimiento en capacidad e incremento económico, y es así como los procesos que se empiezan a migrar a llevar en un porcentaje alto un componente de tecnología que facilite y optimice la obtención de los resultados y la aplicación de los procesos óptimos desde hace varios años ha venido transformando el ecosistema industrial y del agro.

Dentro de los mencionados procesos la posibilidad de generar soluciones intuitivas y asequibles forman parte de la democratización de la tecnología y como esta puede beneficiar a muchas personas desde el punto de vista de procesos como en la ejecución de tareas que dignifica la calidad laboral de muchas personas moviendo lo que hasta ahora se ha apalancado en un esfuerzo físico a un esfuerzo mental y procedimental que con lleva a dar los mejores resultados y presentando una vía a desarrollar la capacidad de generar “trabajo inteligente”.

El país tiene un potencial fuerte en el crecimiento en el sector agrícola por la diversidad de condiciones climáticas que existen en las regiones del país, esto sumado a la latente necesidad de productos referencia a que la adopción de tecnologías dentro de estos procesos permita acelerar el crecimiento de este sector facilitando con la tecnología la precisión y la generación formas tecnológicas que permiten llevar la trazabilidad de los procesos, así como visualizar datos que hasta la el día de hoy han sido invisibles.

En este punto es cuando la aparición de la tecnología denominada “internet de las cosas” se concibe como una herramienta que permite la integración de agricultura de precisión con sistemas avanzados de captación, procesamiento, predicción y generación de estadísticas que facilitan al agricultor la toma de decisiones como la implementación de nuevas tecnologías en un menor esfuerzo y con un mayor índice de probabilidad de éxito.

A través de este proyecto se busca desarrollar un sistema de medición de variables climatológicas que le faciliten y optimicen los procesos manuales que al día de hoy están presentes en los cultivadores de aguacate en el municipio de Tona, Santander, siendo este un punto de referencia para llevar este alcance a otros cultivos convirtiendo este proyecto en una herramienta para quienes están en el sector del agro y buscan alternativas para optimizar sus procesos, mejorando su rentabilidad y alcanzando índices como calidad y capacidad de producción.

2. Planteamiento del Problema

Los cultivos agrícolas comúnmente son de sumo cuidado, la presencia de enfermedades, la variación del clima, las condiciones del suelo y el suministro efectivo de agua son garantes de un producto de calidad y el éxito de una siembra productiva, de esta manera tener una siembra de manera tradicional y sin herramientas ni conocimiento incrementen el riesgo en la ejecución de lo planeado.

Para esto se detectan los siguientes problemas relacionados con el proceso de cultivo de aguacate:

1. Desconocimiento y detección temprana de enfermedades en los cultivos.
2. Suministro de agua efectivo en los cultivos.
3. Ausencia de métricas de los cultivos.

Para este proyecto se ha decidido trabajar en el tercer problema, ausencia de métricas de los cultivos, esto debido a la falta de sistemas de monitoreo efectivo hace que la obtención de las métricas sea manuales, intuitivas e imprecisas impactando directamente en los resultados del cultivo en calidad y capacidad de producción.

En algunos casos en los cultivos periódicamente se tiene la práctica de tomar mediciones climáticas y del suelo para tener una idea de control de plagas, disminución de riesgos y suministrar adecuadamente el fertilizante o la hidratación en cada árbol, este proceso al ser manual es un proceso con altas probabilidades de ser impreciso y poco efectivo.

Para tener una idea de las exigentes características del proyecto se citan los siguientes datos claves en el proceso de la siembra de aguacate Hass:

El suelo debe tener un pH de 5.5 a 7, la necesidad latente de agua durante todo el periodo y especialmente en el primer año de crecimiento con un requerimiento de lugares en donde las precipitaciones anuales sean de entre 1.500 mm y 1.800 mm; A pesar de que el aguacate Hass se puede dar en alturas de entre los 0 y 2.500 msnm, el ideal se encuentra entre los 1.800 y 2.500 msnm a una temperatura ideal de entre 10° C y 25°, según el tipo de suelo de la región se requerirá realizar 2 riegos anuales en épocas de sequía, también es importante realizar según lo requiera el suelo, cada 45 días realizar un proceso de fertilización en la gotera del árbol. (Bancolombia, 2018)

Con base en lo anterior los datos en el municipio de Tona como centro de implementación del producto mínimo viable son los siguientes:

- Precipitación anual: 1.114 mm
- Altura: 1.929 msnm
- Temperatura promedio: 16.1 ° C



Figura 1 Parque principal de Tona, Santander. Fuente: Tomada de tona-santander.gov.co

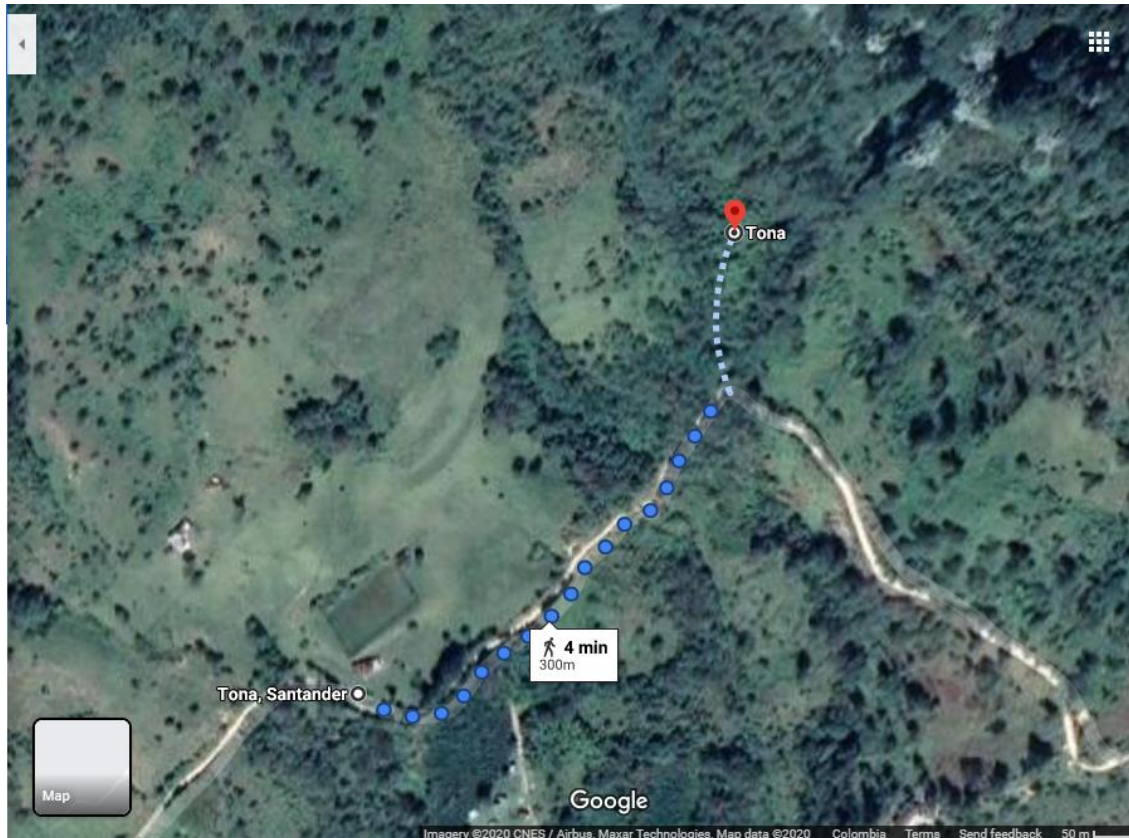


Figura 2 Ubicación de la finca y la distancia de los dispositivos dónde en el marcado rojo indica la ubicación del dispositivo nodo y el punto blanco es la ubicación del dispositivo base. Tomada de Google Maps ([7.189320, -72.996733](https://www.google.com/maps/place/7.189320,-72.996733))

La trazabilidad y las bitácoras es una práctica poco común entre los agricultores, esto impide la caracterización de la región que a futuro les permitirá a estos agricultores predecir o conocer mejor sus tierras y tomar decisiones a tiempo con miras a reducir el riesgo de sus cultivos detectado de manera temprana y efectiva características que puedan afectar el resultado de sus cultivos.

Esto lleva a plantear la pregunta: ¿Cómo a través de la tecnología se puede abordar una solución que permita llevar la trazabilidad de las mediciones climáticas de los cultivos de aguacate de una manera automatizada y que les ayude a tomar decisiones a los agricultores de manera efectiva y a tiempo?

3. Objetivos

3.1 Objetivo General

Construir un prototipo electrónico para el monitoreo climatológico aplicado en cultivos de aguacate en el municipio de Tona Santander, basado en tecnología IoT.

3.2 Objetivos específicos

- Seleccionar los sensores adecuados para el monitoreo de las variables climatológicas de temperatura, humedad relativa, radiación solar y presión barométrica.
- Implementar hardware que permita la integración de los sensores seleccionados para la toma de dichas mediciones.
- Implementar un protocolo de comunicación que permita llevar la trazabilidad de las mediciones obtenidas por el hardware implementado.
- Construir un prototipo funcional del proyecto evidenciando la toma de las mediciones de los sensores.

4. Justificación

El desarrollo de este proyecto se sustenta en la necesidad de optimizar la manera en que se realiza el monitoreo de los cultivos de aguacate y se genera una propuesta de valor agregado en la tecnificación del proceso.

La principal razón para seleccionar este cultivo es debido a que el municipio de Tona tiene un clima ideal para la siembra de cultivos de aguacate, también debido a que gracias a la demanda de este en otros países hace que surja la necesidad de tener un sistema de monitoreo sea fundamental en el éxito de este tipo de cultivos.

Mediante la implementación de este proyecto se podrá optimizar la manera en que se realiza el monitoreo de los cultivos, generando un prototipo que a futuro pueda convertirse en un equipo con el que se facilitará la adopción y crecimiento exponencial de este tipo de soluciones que día a día se alimentan del feedback de cada actor del sistema permitiendo ajustarse a la realidad de los cultivadores de aguacate.

Este tipo de proyectos abre la puerta al desarrollo de tecnología del tipo Agro 4.0 acorde a la época que beneficia a productores, proveedores y consumidores, además esto encamina de una manera ideal al productor a la digitalización y promueve la reactivación del campo como fuente de desarrollo del país mediante la implementación de tecnologías que contribuyen en la eficiencia de los cultivos y permite alcanzar los estándares de calidad requeridos, además de generar valor agregado, disminuir las pérdidas y devoluciones para los agricultores.

El sector del agro es fundamental para la economía del país y es uno de los sectores que a pesar de las situaciones que se presentan de improviso, sea un sector vigente e indispensable, lo cual generar soluciones que contribuyan al crecimiento y optimización de este importante sector para la economía nacional es contribuir con el desarrollo del país; cultivos como el del

aguacate que tiene proyección a crecer substancialmente teniendo en cuenta la demanda de “los principales compradores del aguacate colombiano están en la Unión Europea, en donde el Reino Unido y Países Bajos son los principales destinos de la producción nacional, seguido por España y Francia; entre estos cuatro países se adquieren más del 95% de las ventas externas de Aguacate Hass del país. Estos efectos se deben principalmente a la entrada en vigencia del TLC de Colombia con la Unión Europea a partir del 2013.” (PARRA RAMIREZ & SERRANO SCARPETTA, 2017, pág. 9)

En cuanto a la región de Santander el aguacate clasificado como fruta representan el 7.3% del total de los productos del sector agrícola (Camara de comercio de Bucaramanga, 2018), esto permite sustentar la orientación de este proyecto en el impacto positivo de la región, brindando una herramienta que permita la optimización de los procesos, el seguimiento y trazabilidad y generando una herramienta para la mejora de la calidad del producto en sus procesos de siembra.

La importancia de contar con tecnología que facilite y optimice los procesos en la industria agrícola pasa de ser un privilegio a una necesidad que debe estar al alcance de la mayoría de los productores, herramientas que cada día se hace más práctica, intuitiva y poderosa, brindando cada vez más y mejores herramientas a los agricultores; tecnologías como el internet de las cosas contribuye en el desarrollo de este importante sector y contribuye cada vez a resolver de manera creativa los retos.

Uno de los retos que hace parte del sector rural es la conectividad, y como bien en su nombre el internet de las cosas depende de una conexión estable y eso es algo que no en todas las zonas se puede contar con una conexión, es por esto que la utilización de tecnología LoRa en este proyecto es clave, el uso de esta tecnología brinda la posibilidad de llegar a los cultivos y crear una red “offline” que lleva a un punto central los datos y en este se cuenta con una sola

conexión a internet, esto sumado a que dicha tecnología soporta diversos rangos de alcance, los cuales van desde los 250 m. hasta los 10 Km. Siendo una alternativa bastante eficiente para facilitar el volcado de soluciones IoT para el sector agro, lo cual trae consigo la posibilidad de potencializar el sector.

La tecnología existente por su alto costo se vuelve inasequible para los agricultores, es por esto por lo que este proyecto busca construir unos dispositivos de bajo costo con los que se pueda dar una alternativa a un segmento de agricultores más amplio y de esa manera se pueda iniciar el desarrollo de tecnologías que beneficien el sector del agro, esto sumado al valor diferencial de implementar los sistemas de comunicación LoRa como optimizador de la comunicación de los dispositivos.

5. Marco Referencial

El cultivo de aguacates es uno de los cultivos que promete en crecimiento de la economía “según cifras del Gobierno, las exportaciones de aguacate Hass entre enero y septiembre de 2019 sumaron 35.700 toneladas, es decir, US\$71 millones. Esto representó un crecimiento de 45,5 % respecto al mismo periodo de 2018.” (El Espectador, 2019) , es por esto por lo que el uso de tecnología IoT en el agro se vuelve eficiente y permite tener mejores resultados en los agricultores, en los sistemas IoT se tienen “sistemas de sensores permiten monitorear en tiempo real los niveles de agua, presión y humedad de sus cultivos e, incluso, prevenir plagas y determinar el volumen de agua requerido para un terreno.” (Destino Negocio, s.f.).

En el campo de la agricultura, y en cultivos como el del aguacate el proceso en un término general el monitoreo ha venido siendo realizado en un proceso de medición con equipos análogos o de medición digital, pero sin métricas digitales, que pasan a ser un registro en un documento; “aunque hasta hace poco la barrera de entrada a este tipo de servicios era muy elevada por sus altos costos, y solo aplicable a grandes compañías agroindustriales, la irrupción de nuevos sensores y drones de bajo costo asociados al IoT hacen posible que agricultores con cultivos de tamaño medio ahora puedan acceder a estas tecnologías y sus consecuentes ventajas competitivas.” (Destino Negocio, s.f.).

El internet de las cosas en este momento es uno de los focos de desarrollo tecnológico donde se centra el esfuerzo en la búsqueda de hacer tangible la tecnología mediante la interacción con componentes físicos, esto con el objetivo de llegar a obtener una mayor precisión en los procesos que se llevan en cada área donde se implementa el IoT.

En la concepción del presente proyecto es importante recalcar conceptos que estarán demarcados y serán vitales en el desarrollo de este proceso, esto con miras a obtener los

resultados propuestos, de esta manera los conceptos importantes grosso modo de este proyecto son:

5.1 Proceso de producción del aguacate

Para hablar del proceso de cultivo de aguacate debe empezar por las características de este con la finalidad de comprender la manera en que el proyecto debe orientarse y llevar a cabo durante la ejecución y por qué Tona es una región óptima para el cultivo del aguacate.



Figura 3 Aguacate producido en una de las fincas de Tona. Fuente: Elaboración propia.

De esta manera el aguacate hace parte de la familia lauráceas y de la especie *Persea americana* de origen mexicano y con una notable expansión por las Antillas, en su presentación es un árbol que puede llegar hasta una altura de 30m.

Las exigencias del clima indican que debe cultivarse hasta una altura de 2500 msnm, pero en un escenario ideal de entre 800 y 2500 msnm esto con el objetivo de prevenir enfermedades, las cuales normalmente se sitúan en las raíces; por otra parte, el nivel de precipitación ideal es de 1200 mm anuales, lo cual indica que la clave para tener éxito en el cultivo del aguacate es evitar las sequías en los cultivos.

Otro factor sumamente importante es el nivel de pH del suelo, este debe estar entre 5.5 y 7, en cuanto al tipo de suelo se recomienda que sea en suelo de textura ligera, pero también se puede cultivar en suelo arcilloso, esto teniendo en cuenta que cuenten con buen drenaje, ya que la humedad puede provocar enfermedades en la raíz de los árboles.



Figura 4 Árbol de aguacate evidenciando los primeros frutos. Fuente: Elaboración propia.

A su vez, el aguacate puede dar en variedades, principalmente influenciadas por la altura en la que se encuentran cultivadas, dichas variaciones se exponen a continuación.

Tabla 1*Variedades de aguacate aptas para el cultivo en las diferentes altitudes.*

ALTURA		
De 0 – 1.000 msnm	De 1.000 msnm - 1.500 msnm	De 1.500 msnm – 2.500 msnm
Simmonds	Choquete	Nabal (G)
Catalina	Kahalú	Azteca
Booth 8	Hall	Fuerte
Booth 7	Simpson	Hass
Masutomi	Booth 8	Ettinger
Kahalú	Guatemala	Wurstz
	Fujikawa	
	Itzama	

Fuente: (ANACAFE, 2019)

5.1.1 Preparación del suelo.

El proceso de preparación del suelo depende de la topología y la vegetación de la región, por otra parte, si el terreno ha sido cultivado previamente será necesario únicamente hacer los hoyos con un diámetro de 60 cm. y una profundidad de entre 50 y 60 cm., por otra parte, si existen hierbas malas se debe aplicar un herbicida para eliminarla y posteriormente arar la tierra; luego de tener un terreno limpio se procede a demarcar las zonas y construir zanjas de protección que permiten contribuir en la conservación del suelo.

5.1.2 Eliminación de hierbas malas.

En este proceso se debe tener cuidado al utilizar herramientas cortantes que puedan generar heridas a los troncos de los árboles y causar el hongo *Phytophthora cinnamomi* que produce marchitez del aguacate.



Figura 5 Marchitamiento del aguacate. Fuente: Tomado de Guerrero Tejada, Juan Alejandro (http://www.avocadosource.com/WAC7/Section_01/GuerreroTejadaJesus2011.pdf)

Es importante no dejar expuesto el terreno, ya que esto generará una erosión, en cuanto a los herbicidas uno dedicado a la maleza es el Terbutizalina.

5.1.3 Poda.

La poda en los primeros 3 años no es necesario, luego de esto será necesario realizar la poda de ramas de crecimiento vertical excesivo, ramas bajas y los tallos enfermos.



Figura 6 Árbol de aguacate joven podado. Nota 1 Tomado de Intagri (<https://www.intagri.com/articulos/frutales/la-poda-en-el-cultivo-de-aguacate>)

5.1.4 Propagación.

La propagación del aguacate puede darse vía semilla o injerto, la propagación por semilla en cultivos comerciales es poco recomendada por estándares de calidad en el producto, tamaño y forma del fruto.

“Las semillas deben provenir de frutas sanas, de buen tamaño, cosechadas directamente del árbol. Su viabilidad dura hasta tres semanas después de extraída de la fruta. Es recomendable cortar la parte angosta de la semilla, en un tramo de una cuarta parte del largo total, para ayudar así a la salida del brote y para hacer una primera selección, ya que el corte permite eliminar las semillas que no presenten el color natural blanco amarillento, debido a podredumbre, lesiones o cualquier otro daño. Inmediatamente después de cortadas, se siembran en el semillero previamente preparado colocándolas sobre el extremo ancho y plano de modo que la parte cortada quede hacia arriba. Las semillas empiezan a brotar aproximadamente treinta días después de sembradas. Generalmente las plantas están listas para ser trasplantadas al vivero, a los treinta días después de la germinación.” (Info Agro, s.f.)

Este proceso de injerto se puede llevar a cabo en un invernadero o en el sitio final donde se dispondrá el árbol, en cuanto a el patrón debe tener un tallo de 1 cm de diámetro y 10 cm. de la base, esto en un tiempo estimado de aproximadamente 6 meses luego de la siembra y este procedimiento debe realizarse en un sitio fresco.

Se debe seleccionar las púas de los árboles seleccionados en la variedad deseada con las características ideales, el injerto se realiza a aproximadamente 20 cm. de altura en el patrón, luego de esto entre 22 y 30 días después se procede a quitar la parte superior del patrón, el injerto debe realizarse en un corte de tipo bisel y debe estar cubierto por una crema fungicida a base de cobre; el injerto podrá ser trasplantado al sitio definitivo cuando este tiene una altura de entre 20 y 25 cm de alto.



Figura 7 Ejemplo de un injerto de aguacate. Nota 2 Tomado de Portal frutícola (<https://www.portalfruticola.com/noticias/2016/12/06/paso-a-paso-para-hacer-un-injerto-de-corona-de-aguacate/>)

5.1.5 Recolección.

En el caso de cultivos basado en injertos el tiempo de recolección es de 5 años y la cantidad de fruto a obtener dependerá de la variedad del aguacate seleccionado en el injerto, pero en términos generales a los 5 años una planta genera 50 frutos, al sexto año genera 50 frutos, al séptimo año serán trescientos frutos y al octavo año generará ochocientos frutos, en algunas variedades como el Hass producen entre 1.000 y 1.500 frutos luego de los diez años.

Según la altura de la región del cultivo varía el momento de recolección, en zonas bajas se da entre abril y agosto, en zonas medias ocurre entre junio y septiembre y en regiones altas entre septiembre y abril.

La madurez del fruto se determina mediante la cantidad de grasa de este, el proceso de recolección es manual, debido a su actividad respiratoria es difícil alojar por mucho tiempo el aguacate y se hace necesario su procesamiento o distribución a la menor brevedad posible, adicional a esto contar con una zona de almacenamiento con condiciones óptimas contribuirá en la calidad y durabilidad del fruto luego de su recolección.



Figura 8 Recolección de aguacate. Nota 3 Tomado de Crónica del Quindío (<https://www.cronicadelquindio.com/noticias/economia/productores-de-aguacate-preocupados-por-futuros-excedentes-de-la-fruta>)



Figura 9 Recolección de aguacate. Fuente: Tomado de EFE Agro (<https://www.efeagro.com/noticia/alianzas-beneficios-produccion-organica-latam/>)

5.1.6 Marco de plantación.

Un árbol que fue sembrado hace seis meses ya se encuentra listo para ser trasplantados, acorde al tipo de suelo, topografía, variedad y condiciones ambientales se presentará un marco de plantación, en una relación dada por una mayor fertilidad o menor altura las distancias deben ser mayores.



Figura 10 Marco de plantación. Fuente: Tomada de Portal Frutícola (<https://www.portalfruticola.com/noticias/2019/05/22/tipos-de-disenos-y-marcos-de-plantacion-en-palto-hass>)

En un sistema estándar las distancias varían entre 7m x 9m a 10m x 12m, teniendo un espaciamiento de 10m entre plantas y 10m entre hileras.

5.1.7 Riego.

En el primer año un factor importante del cultivo del aguacate es el riego, contar con suficiente agua les permitirá desarrollarse de manera óptima, eso será decisivo para el futuro

del árbol, cabe recalcar que la cantidad de agua suministrada debe ser justa en su necesidad, el exceso de agua también es un problema para el árbol.



Figura 11 Sistema de riego del aguacate. Fuente: Tomado de Cultifort (<https://www.cultifort.com/cultivo-aguacate-consejos-experto/>)

5.1.8 Fertilización.

Con el fin de establecer un abono adecuado para el suelo se debe realizar un análisis de este con el fin de establecer los niveles de nutrientes con los que se cuenta, dicha labor es recomendable realizarla cada año, pero se puede realizar en un máximo de 3 años.

Para el proceso de trasplante se debe tener en cuenta utilizar en cada hoyo 250g de un fertilizante rico en fósforo como el de la fórmula 10-30-10 o triple superfosfato.

De igual manera por cada año de edad del árbol, un kilo de fertilizante rico en nitrógeno y potasio como el de la fórmula 18-5-15-6-2, repartido en tres aplicaciones, una cuando inician la época de lluvia y las otras dos cada dos meses, es de tener en cuenta que cada árbol debe

recibir máximo 12 kilos de fertilizantes para árboles de 13 años en adelante, con esto se busca lograr una producción constante.

En cuanto el árbol entra en etapa de producción la cantidad de nitrógeno requerida por el mismo será mayor, sobre todo en la etapa de floración y maduración del fruto, el uso de 1 kilo de urea 40 días posteriores a la floración.



Figura 12 Proceso de fertilización de árboles de aguacate. Nota 4 Tomado de *Dialogo con la realidad* desde la literatura (<http://dialogoconrealidadliteraria.blogspot.com/2012/02/fertilizacion-del-aguacate.html>)

5.1.9 Plagas.

En los cultivos de aguacate se pueden presentar diversos factores que pueden desviar el cultivo de su propósito inicial, dentro de este margen de imprevistos que pueden prevenirse o tratarse se encuentran las plagas, existen del tipo insectos y ácaros, las siguientes son las principales plagas que se deben tener en cuenta:

5.1.9.1 Taladrador del tronco *Copturomimus perseae* Gunthe.

Según Citri-Caldas (2017)

Esta plaga barrena el tronco y las ramas, las cuales se van secando hasta provocar la muerte del árbol; su daño se extiende incluso hasta los frutos. Se han reportado pérdidas cercanas al 85% en cultivos atacados por esta plaga. El daño se manifiesta por medio de pequeños puntos, cubiertos con una exudación blanquecina, que corresponde a la savia cristalizada, lo cual poco a poco va tomando un color negruzco muy característico sobre la superficie de la corteza. (p.1)



Figura 13 Copturomimus perseae Gunthe. Fuente: Tomado de Citricaldas (<https://www.citricaldas.com.co/barrenador-de-las-ramas-del-aguacate-copturomimus-perseae/>)

5.1.9.2 Talador de la semilla *Heilipus luari* Boh (Coleoptera: Curculionidae)

***Heilipus pittieri* (Barber).**

Estos insectos ovipositan en frutos, al emerger sus larvas se alimentan tanto de la pulpa como de la semilla, destruyéndolos en su totalidad, produciendo caída prematura. Llegan a ocasionar pérdidas hasta del 100%. (Comité Estatal de Sanidad Vegetal del Estado de México CESAVERM, 2008)



Figura 14 Heilipus pittieri (Barber). Fuente: Tomado de Dirección general de sanidad vegetal Centro Nacional de Referencia Fitosanitaria (<https://n9.cl/aslb4>)

5.1.9.3 Trip del aguacate *Heliethrips haemorrhoidalis* (Bouche) (Thysanoptera: Thripidae).

Su ataque provoca que la epidermis de los frutos y de las hojas se engrose y se agriete.

La lucha mediante insecticidas es fundamental, como: diazinon, malation, acefato, monocrotofos, o piretroides. (Info Agro, s.f.)



Figura 15 Trip del aguacate *Heliethrips haemorrhoidalis*. Fuente: Tomado de Intagri (<https://www.intagri.com/articulos/fitosanidad/manejo-trips-cultivo-aguacate>)

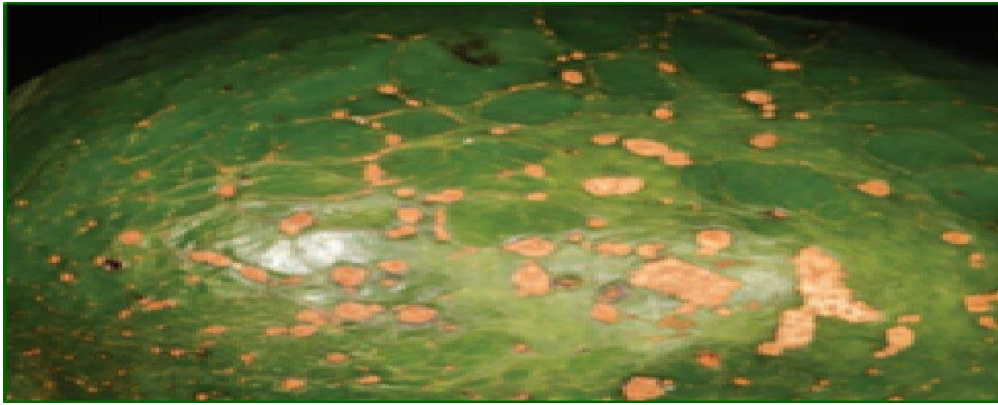


Figura 16 Daño causado por Trip del aguacate *Heliothrips haemorrhoidalis*. Fuente: Tomado de Intagri (<https://www.intagri.com/articulos/fitosanidad/manejo-trips-cultivo-aguacate>)

5.1.10 Enfermedades.

En los cultivos se presenta con normalidad una serie de enfermedades, de las cuales se presentan a continuación las principales:

La larva se introduce en el fruto cuando está en desarrollo y perfora la piel y la pulpa, para su combate, se recomiendan aplicaciones mensuales de insecticida, a partir del momento en que el fruto está recién cuajado con carbaril, también es muy importante recoger los frutos caídos, destruirlos y quemarlos. (Info Agro, s.f.)



Figura 17 *Stenoma catenifer*. Fuente: Tomado de Gobierno Mexicano (<https://n9.cl/twrg4>)

5.1.10.1 Pudrición de la raíz o marchitez del aguacate *Phytophthora cinnamomi* rands.

El daño a la raíz disminuye la capacidad de absorción de agua y nutrimentos, lo que se manifiesta en una pérdida de turgencia y marchitez de hojas. Al disminuir la producción de citoquininas y giberelinas, junto con un incremento en el suministro de etileno hacia la parte superior de la planta, se aprecia un retraso en el crecimiento de brotes, clorosis de hojas, muerte de yemas apicales y laterales, floración excesiva y producción abundante de frutos de bajo calibre. Finalmente ocasiona marchitez generalizada y defoliación del árbol, causando la muerte de la planta en un periodo corto de tiempo. (Intagri, s.f.)



Figura 18 Evidencia del daño causado por pudrición de la raíz o marchitez del aguacate Fuente: Tomado de Intagri (<https://www.intagri.com/articulos/fitosanidad/control-de-la-pudricion-de-raiz-en-aguacate>)

5.1.10.2 Mancha negra o cercospora *Cercospora purpura* Cooke.

Ataca las hojas y produce lesiones pequeñas color marrón oscuro. Cuando el ataque es severo causa su caída quedando los árboles defoliados. En los frutos produce lesiones pequeñas,

oscuras, de bordes irregulares y el resquebrajamiento de la corteza. Tanto las lesiones en las hojas como en el fruto facilitan la entrada para otros organismos como *Colletotrichum*.

Para su combate se recomiendan aspersiones con fungicidas a base de cobre, como hidróxido de cobre, oxiclورو de cobre o sulfato de cobre, ya sea solos o mezclados con otros como clorotalonil, benomilo, etc. (Hoy, s.f.)

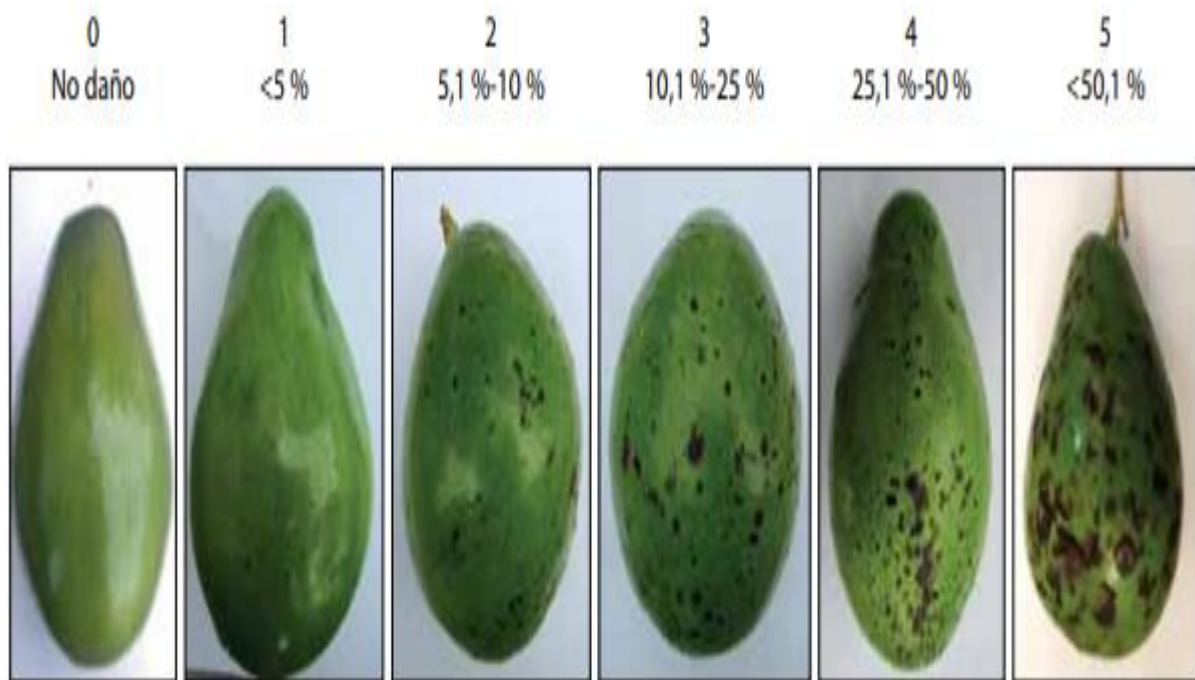


Figura 19 Presencia de cercospora. Fuente: Tomado de Scielo.org (<http://www.scielo.org.co/pdf/ccta/v16n2/v16n2a09.pdf>)

5.1.10.3 Polvillo o mildiu oidium sp.

En esta enfermedad “según se expande, las hojas comienzan a rizarse, presentando necrosis y desembocando en la caída de estas. Las partes del micelio que contienen las esporas de los hongos salen por la estoma de la planta. El mildiu es reconocible a plena luz del día por el color grisáceo o amoratado que muestra el envés de las hojas.” (Canna, s.f.)



Figura 20 Daño causado por polvillo o mildiu *oidium sp* Fuente: Tomado de Canna (<https://www.canna.es/oidio-y-mildiu-plagas-enfermedades>)

5.1.11 Cosecha.

Durante la etapa de postcosecha se presentan factores importantes como la calidad, el proceso de cosecha, temperatura, humedad, entre otros descritos a continuación:

5.1.11.1 Calidad.

La calidad en el aguacate se delimita según el consumidor, pero principalmente se basa en las siguientes características tamaño, forma, color de la cáscara y ausencia de defectos tales como heridas, manchas, quemaduras de sol, malformaciones, también cabe denotar que un almacenamiento incorrecto puede derivar en mal sabor del fruto.



Figura 21 Aguacate de calidad. Fuente: Tomado de El Confidencial (https://www.alimente.elconfidencial.com/nutricion/2019-05-02/aguacate-potasio-grasas-calidad_1517887/)

5.1.11.2 Temperatura óptima.

La temperatura ideal para un aguacate verde maduro es de entre 5° C y 13° C y depende del cultivo y de la duración de la baja temperatura, en caso de aguacates de consumo la temperatura es de 2° C a 4° C.

5.1.11.3 Humedad relativa.

Para un cultivo con condiciones ideales de la humedad es de entre el 90% y el 95%.

5.1.11.4 Tasa de respiración.

Para calcular el calor producido por cada planta se multiplica los mililitros de [CO₂](#) y se dividen en los kilogramos por 440 para obtener Btu/ton/día o por 122 para obtener kcal/ton métrico /día. (Info Agro, s.f.)

Tabla 2*Tasa de respiración del cultivo de aguacate*

Temperatura	5° C	10° C	20° C
mL CO ₂ / kg-h	10 - 25	25 - 80	40 - 150

Fuente: Elaboración propia.

5.1.11.5 Tasa de producción de etileno.

Solo hasta posterior a su cosecha empiezan a madurar y a estar en el punto de consumo ideal, esto se da en proporción de 100 uL C₂H₄/kg * h a 20° C.

La base con la que se construyó lo anterior están soportados principalmente en la fuente de Infoagro.com y la *guía completa: Cómo cultivar aguacate*. (Bancolombia, 2018) (Info Agro, s.f.)

5.2 Meteorología

La meteorología es una disciplina y técnica que se encarga de estudiar los sucesos que ocurren en la atmósfera con el fin de comprender dichos eventos y tratar de predecir la incidencia de estos en campos como la agricultura, aeronáutica, navegación, actividades militares, entre otras.

Dentro de los conceptos que se emplean en la meteorología se encuentran clima y el tiempo, determinando el tiempo como el comportamiento de la atmósfera en determinada ubicación y tiempo, por otra parte, el clima hace referencia al alcance estadísticos del tiempo en un periodo prolongado. (Metereología en red, s.f.)

5.2.1 Estación meteorológica.

Una estación meteorológica es un equipo compuesto de una diversidad de dispositivos electrónicos que se ocupan de captar diversas variaciones en el ambiente y son traducidas a valores que puedan ser trazados y sirvan como base de estudios y aplicaciones sobre diversos modelos numéricos que contribuyen al fácil entendimiento de lo que están detectando los dispositivos.

Dentro de una estación meteorológica se puede encontrar los siguientes equipos:

- Piranómetro (radiación solar)
- Higrómetro (humedad)
- Veleta (Dirección del viento)
- Anemómetro (Velocidad del viento)
- Barómetro (Presión atmosférica)
- Termómetro (Temperatura)
- Pluviómetro (Caída del agua)



Figura 22 Estación metereológica. Fuente: Tomado de Estación Meteorológica (<https://estaciondemeteorologia.com/que-es-una-estacion-meteorologica/>)

5.2.2 Variables ambientales.

Cada variable que se toma en el proceso de medición existe una serie de mediciones de las cuales se exponen a continuación algunas de las principales y más utilizadas:

5.2.2.1 Temperatura del aire.

Se encarga de determinar la temperatura en la velocidad en un contenedor o en el medio ambiente, en la mayoría de las estaciones dicha medida se mide en grados centígrados. (Inia, s.f.)

5.2.2.2 Precipitación pluvial.

Esta medida es la que determina la cantidad de gotas de agua que caen a la superficie y su medición está dada en milímetros. (Fluencecorp, 2017)

5.2.2.3 Humedad ambiental.

Representa la cantidad de vapor agua presente en el ambiente, se mide con la ayuda de un psicrómetro y se representa en porcentaje, con esta se puede predecir heladas o el desarrollo de enfermedades en los cultivos. (Colegio Americano del sur del valle, 2015)

5.2.2.4 Radiación solar.

Es la radiación de ondas electromagnéticas en diferentes frecuencias, dado en una medida en calorías por centímetro cuadrado por minuto o por día o también en watts por metro cuadrado. (Ideam, s.f.)

5.2.2.5 Velocidad del viento.

Esta variable se mide en metros/segundo o kilómetros/hora, esta se captura mediante un anemómetro. (Navarra, s.f.)

5.2.2.6 Dirección del viento.

Hace referencia al punto de horizonte de donde proviene la fuerza de este, representándolo sobre los puntos cardinales y los grados. (Navarra, s.f.)

5.3 Marco conceptual

Con el fin de establecer conceptos claves que se abordarán a lo largo del proyecto se presentan a continuación los principales que permiten generar el conocimiento necesarios.

5.3.1 Internet de las cosas – IoT.

La implementación de sistemas técnicamente complejos y de uso sencillo es una de las características de esta tecnología que permite genera interfaces de usuario técnicamente complejas, generalmente retos de ingeniería, pero en su parte de adopción generar una usabilidad demasiado sencilla y entendible en un juego de lógica simple que le permite a alguien sin conocimientos en tecnología adoptar la misma partiendo de que esta está diseñada en premisas lógicas conocidas por quien interactuará con el sistema. (Pérez, 2017)

5.1.2 Aplicación.

En este tiempo la adopción de la tecnología se ha ido recalando día tras días más y más, llegando a la mayor cantidad de personas en diferentes rangos y condiciones, la adopción de tecnología fácil de utilizar y atractiva en valores agregados es notoria en cada idea que se vuelve realidad en un crecimiento acelerado. La masificación y uso de los dispositivos móviles y el internet facilita la adopción y familiarización de nuevas interfaces de operación tecnológica, en este caso, las aplicaciones han representado y hecho tangibles los modelos de ingeniería complejos y no aptos para todo el mundo, gracias a esto hoy podemos consultar cosas que hace un par de décadas eran impensables y sólo un gran equipo de ingeniería con bastos

conocimientos podría dominar, hoy en día todo esto pasa sin que si quiera el usuario sea consciente de los procesos que está realizando y menos el grado de complejidad del mismo.

(Universidad de Sevilla, 2004)

5.1.3 Sensórica.

Cada día en nuestra cotidianidad se introducen nuevas tecnologías que permiten tener más datos de nuestro medio o del medio de un objeto en específico, de esa manera la sensórica permite el muestreo y la generación de disparadores de decisiones automatizadas o manuales de un proceso que termina en la eficiencia de un sistema complejo que se encamina en cada vez ser más eficiente, preciso y rentable. (PCE-Iberica, s.f.)

5.1.4 Cloud Computing.

Esto hace referencia al poder del procesamiento en servidores de gran capacidad que se encuentran en lugares remotos y que permiten la creación de conexiones en tiempo real, de gran escalabilidad y de robustez y sobre dimensionamiento inimaginables que permiten suplir grandes retos de ingeniería y facilitar la adopción de nuevas tecnologías que se podrán ejecutar de manera centralizada y remota, esto con miras a garantizar un acceso con alta disponibilidad, además de resguardar la información importante de los procesos y llevar a otro nivel mediante el tratamiento y procesamiento de información que una solución eficiente como la tecnología de computación en la nube nos brinda hoy en día y que se encuentra en un punto de madurez ideal en el que se podrá apalancar infinidad de soluciones que trascenderán en este y otros mercados. (Hurwitz & Kirsch, 2020)

5.1.5 Procesamiento y almacenamiento de datos.

La generación de una estructura de datos en la que de manera lógica se resguarden los datos es una forma en que la tecnología se encuentra presente en nuestro día a día y la manera en que evoluciona constantemente, desde un comienzo hasta lo logrado a la fecha y todo lo que está por venir hace parte del futuro y la generación de datos y correcto procesamiento de la información será clave en un futuro más tecnificado la obtención de respuestas importantes que nos ayuden a comprender el funcionamiento de muchas cosas. (Villa Sánchez, 2018)

5.1.6 Hypertext Transfer Protocol (HTTP).

Es un protocolo de capa de aplicación para transmitir documentos hipermedia, como HTML. Fue diseñado para la comunicación entre navegadores web y servidores web, pero también se puede utilizar para otros fines. HTTP sigue un modelo clásico de cliente-servidor, con un cliente que abre una conexión para realizar una solicitud y luego espera hasta que recibe una respuesta. HTTP es un protocolo sin estado, lo que significa que el servidor no guarda ningún dato (estado) entre dos solicitudes. Aunque a menudo se basa en una capa TCP / IP, se puede usar en cualquier capa de transporte confiable, es decir, un protocolo que no pierde mensajes silenciosamente como lo hace UDP. RUDP, la actualización confiable de UDP, es una alternativa adecuada. (Mozilla, s.f.)

5.1.7 MQ Telemetry Transport (MQTT).

MQTT es un protocolo de mensajería estándar de OASIS para Internet de las cosas (IoT). Está diseñado como un transporte de mensajería de publicación / suscripción extremadamente liviana que es ideal para conectar dispositivos remotos con una huella de código pequeña y un ancho de banda de red mínimo. Hoy en día, MQTT se utiliza en una

amplia variedad de industrias, como la automotriz, la fabricación, las telecomunicaciones, el petróleo y el gas, etc. (Mqtt, s.f.)

5.1.8 Microcontrolador.

Recibe el nombre de controlador el dispositivo que se emplea para el gobierno de uno o varios procesos. Por ejemplo, el controlador que regula el funcionamiento de un horno dispone de un sensor que mide constantemente su temperatura interna y, cuando traspasa los límites prefijados, genera las señales adecuadas que accionan los actuadores que intentan llevar el valor de la temperatura dentro del rango estipulado. (Valverde Villarán, 2005)

5.1.9 Sensor de temperatura.

Un sensor es un objeto capaz de detectar magnitudes físicas o químicas llamadas variables de instrumentación, que pueden ser la intensidad lumínica, la temperatura, distancia, aceleración, etc. El sensor de temperatura empleado más habitualmente es el sensor termopar. Está formado por mecanismos muy sencillos constituidos por dos materiales metálicos de diferente naturaleza unidos mediante uniones llamadas unión fría y unión caliente. El grado de temperatura que detectan estas uniones genera una diferencia de potencial estrechamente dependiente de la naturaleza de los materiales. El sensor térmico transforma esta diferencia de potencial en una señal eléctrica. (siberzone, s.f.)

5.1.10 Sensor de humedad.

El sensor de humedad es un aparato de lectura utilizado en espacios interiores para controlar la humedad del aire y la temperatura. Las magnitudes medidas por el sensor de humedad se transforman en una señal eléctrica normalizada, cuya intensidad suele estar comprendida entre 4 y 20 mA. Un material semiconductor es el encargado de determinar con precisión los valores de humedad y temperatura que se corresponden con la señal emitida. Este

tipo de sensores son especialmente útiles en los sistemas de ventilación mecánica higrorregulables, ya que permiten regular el caudal de aire renovado en función de la humedad ambiental. Estos sistemas pueden emplearse tanto en viviendas individuales como colectivas. (Siberzone, s.f.)

5.1.11 Sensor de lluvia.

El Sensor Electrónico de lluvia es un dispositivo electrónico que nos permite medir las precipitaciones de lluvia, la fuerza del viento y la temperatura existente, con estas mediciones se evita el riego en caso de no necesidad por lluvias y anula la aplicación de este en casos de fuertes vientos y heladas automáticamente, evitando así que el sistema desperdicie agua y que pueda perjudicar al césped y las plantaciones. (Ecu Red, s.f.)

5.1.12 JavaScript Object Notation (JSON).

El acrónimo JSON deriva de la expresión de la lengua inglesa JavaScript Object Notation, que puede traducirse como Notación de Objeto de JavaScript. Se trata de un formato de texto ligero que permite intercambiar datos. (Definicion, s.f.)

6. Metodología

Para el planteamiento de este proyecto se ha seleccionado la metodología cuasi experimental, esto debido a que los grupos de datos a evaluar no pueden ser aleatorios y el conocer el comportamiento los datos es importante para construir un modelo de solución concisa que pueda replicarse y escalarse.

Para este propósito se utiliza un grupo segmentado con características particulares, que en este caso es el cultivo de aguacate, en donde se utiliza metodología descriptiva y en algunos elementos cuantitativos y cualitativos.

De esta manera el tipo de diseño cuasi experimental que aplica es del tipo estudio con controles históricos en donde se cuenta con una variable independiente y una o más variables dependientes con las que se genera la experimentación.

El poder seleccionar grupos establecidos para realizar las pruebas es una ventaja de este tipo de experimentación con el que se puede llegar a generar pruebas controladas y que hace conseguir los resultados tenga una guía que facilite el proceso de investigación.

Para este tipo de experimentación se cuenta con dos tipos de diseños, uno transversal y otro longitudinal, donde el primero se estudia con valores en un momento determinado, mientras que en la segunda se basa en la evolución de los datos seleccionados durante un periodo de tiempo.

Con base en lo anterior el desarrollo del proyecto se cimenta sobre las siguientes etapas que permiten llegar a la construcción del producto esperado, evaluando el avance en cada una de las mismas, segmentando en cada etapa los datos y buscando la manera fiel de obtener los datos que conduzcan al correcto desarrollo del proyecto.

Actividad / Mes	1	2	3	4	5	6	7	8	9	10	11	12
Selección de sensores	✓											
Adquisición de los sensores para el monitoreo de las variables climatológicas	✓											
Implementación e integración de los sensores adquiridos	✓	✓										
Implementación del protocolo de comunicación entre los dispositivos		✓										
Construcción del prototipo hardware		✓	✓	✓	✓	✓						
Medición de las variables monitoreadas y análisis estadístico de las variables monitoreadas						✓						
Consolidación del informe final						✓						

Figura 23 Diagrama Gantt de la metodología del proyecto. Fuente: Elaboración propia.

6.1 Selección de los sensores

Esta etapa corresponde a la búsqueda de los componentes que puedan suplir los requerimientos del proyecto tanto en los aspectos de funcionalidad, disponibilidad y costo, para esto se consultarán las referencias más utilizadas en el mercado con el fin de identificar componentes estables y estandarizados que puedan contar con soporte al momento de realizar el desarrollo.

6.2 Adquisición de los sensores para el monitoreo de las variables climatológicas

Luego de identificar los componentes requeridos se procederá a realizar la búsqueda de proveedores que puedan brindar los elementos requeridos en un tiempo corto y a un precio óptimo siempre manteniendo el enfoque de funcionalidad y costo.

6.3 Implementación e integración de los sensores adquiridos

Al contar con los componentes requeridos, se procederá a realizar las pruebas de conexión y validación de funcionamiento de cada uno de los elementos, esto con el fin de validar la manera en que se debe programar y/o controlar cada uno.

Posterior a esto se deberá realizar las pruebas de integración de cada prueba al igual que establecer los diagramas de conexión de los circuitos, esto con el objetivo de establecer una base sólida de la parte electrónica del proyecto.

6.4 Implementación del protocolo de comunicación entre los dispositivos

Para esta etapa se realizará la validación del protocolo de comunicación a implementar tanto a nivel electrónico como a nivel de software, partiendo de la necesidad de utilizar el protocolo de comunicación LoRA y MQTT, además de validar el resultado en una aplicación existente que facilite la visualización de los datos obtenidos.

6.5 Construcción del prototipo hardware

En esta etapa se realizará la construcción de un prototipo que integre todo lo alcanzado en las anteriores etapas, llegando a una versión del planteamiento realizado en este proyecto sólida que permitirá realizar las pruebas que determinarán el éxito del proyecto en cuanto a los objetivos propuestos.

6.6 Medición de las variables monitoreadas y análisis estadístico de las variables monitoreadas

Para esta etapa teniendo establecido un prototipo con los sensores descritos a lo largo de este proyecto se procederá a realizar las pruebas iniciales de comunicación y monitoreo de las variables que se obtendrán, de esa manera dichas variables podrán ser analizadas para determinar una viabilidad del proyecto en una implementación real del mismo y la efectividad del sistema tanto de IoT como de comunicación LoRA.

6.7 Consolidación del informe final

En esta última etapa se presentarán los resultados condensados del trabajo realizado con el fin de evidenciar la manera en que el proyecto evolucionará y permitirá la evidencia final del proyecto.

7. Resultados

Luego de seguir la metodología propuesta se ha generado el desarrollo de una solución en la cual se han dado los resultados esperados, encontrando un alto potencial en el protocolo de comunicación LoRa con el que las soluciones IoT para el sector de la agricultura se verán exponencialmente beneficiadas.

De esta manera y siguiendo el plan establecido en la metodología se presenta el compendio de resultados obtenidos para el presente proyecto el cual se sustenta en el siguiente diagrama:



Figura 24 Diagrama de la solución que se aborda con el proyecto. Fuente: Elaboración propia.

7.1 Selección de los sensores

En la realización de esta etapa se identificaron los componentes que cumplen con el objetivo del desarrollo del prototipo del proyecto, de acuerdo con el diagrama de solución presentado en la Figura 24 estos serán los componentes necesarios:

- Sensor de temperatura relativa.
- Sensor de humedad relativa.
- Sensor de temperatura en tierra.

- Sensor de lluvia.
- Módulo de comunicación LoRa.
- Sensor de intensidad lumínica
- Sensor de presión barométrica
- Panel solar.
- Batería
- Módulo de carga
- Controlador
- Módulo lector de microSD

De esta manera entendiendo los requerimientos de cada uno de los componentes anteriormente descritos y con la búsqueda de los sensores más utilizados para esos propósitos se seleccionan los siguientes elementos para cumplir a cabalidad el alcance del presente proyecto:

- Sensor de temperatura y humedad relativa de referencia DHT-11

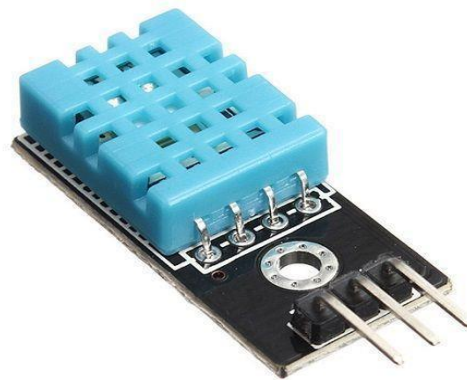


Figura 25 Sensor DHT-11. Fuente: Tomado de Compel (<https://www.compelectronica.com/product/modulo-de-sensor-temperatura-y-humedad-dht11/>)

Características:

- Opera con un rango de voltaje de 3V a 5V
 - La máxima corriente usada durante la conversión es 2.5mA
 - Mide temperatura entre 0°C hasta 50°C, con precisión de $\pm 2^\circ\text{C}$
 - Mide humedad entre 20% RH hasta 90% RH, con precisión de 5%
 - La frecuencia de muestreo es no mayor a 1Hz (una vez por segundo)
 - Es compatible con tarjetas Arduino
 - El pin de salida (DATA) es digital
 - Dimensiones de la tarjeta son 3.2cm x 1.4cm
-
- Sensor de humedad en tierra de referencia YI100



Figura 26 Sensor de humedad en tierra. Fuente: Tomado de Tdrobotica (<https://tienda.tdrobotica.co/retirados/514-sensor-de-humedad-del-suelo.html>)

Características:

- Voltaje de funcionamiento: 3.3v - 5v
 - Señal de salida: 0 - 4.2 v
 - Consumo de corriente: 35mA
- Sensor de lluvia YC69.



Figura 27 Sensor de lluvia YC69. Fuente: Tomado de Moviltronics (<https://n9.cl/q906p>)

Características:

- Voltaje de Operación: 3.3V-5V
- Tamaño de PCB: 3.2cm x 1.4cm
- Tamaño de celda :5cm x 4cm
- Chip Comparador: LM393
- Compatible con Arduino

- Módulo de comunicación LoRa RA02 433 MHz

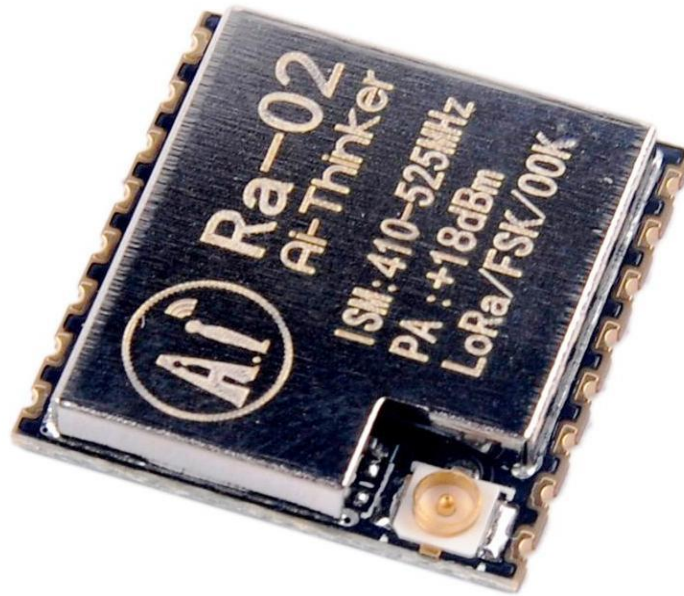


Figura 28 Módulo de comunicación LoRA 02. Nota 5 Tomado de Didácticas electrónicas (<https://n9.cl/0voe>)

Características:

- Voltaje de operación: 1.8 ~ 3.7V
- Corriente en la recepción: $\leq 10.8\text{mA}$ (LnaBoost off, band 1)
- Corriente en transmisión: $\leq 120\text{mA}$ (+ 20dBm)
- Corriente en Sleep mode: 0.2 μA
- Chip WiFi: ESP8266
- Rango de frecuencia: 420MHz ~ 450MHz
- Estándar wireless: 433MHz
- Interface: SPI GPIO Half-duplex
- Bit rates hasta de 300kbps
- Conector para antena externa
- Para ESP-32
- Temperatura de operación: - 40°C ~ 85°C

- Panel solar 6w 500 mA



Figura 29 Panel solar. Tomado de Mercado Libre (<https://n9.cl/btymz>)

Características:

- Voltaje: 6v
 - Corriente: 500 Ma
 - Potencia: 3 W
 - 16cm x Ancho 14.8cm
- Regulador de voltaje Xc6206



Figura 30 Regulador de voltaje Xc6206 Tomado de Iamsmutual (<https://n9.cl/9ihul>)

Características:

- Voltaje mínimo de salida de 3V
 - Voltaje máximo de salida 3V
 - Voltaje mínimo de entrada 4V
 - Voltaje máximo de entrada 6V
 - Corriente máxima de salida: 200 mA
 - Encapsulado: SOT-23
 - Voltaje dropout: 250 mV 100 mA
- Módulo de carga micro-USB para batería de litio

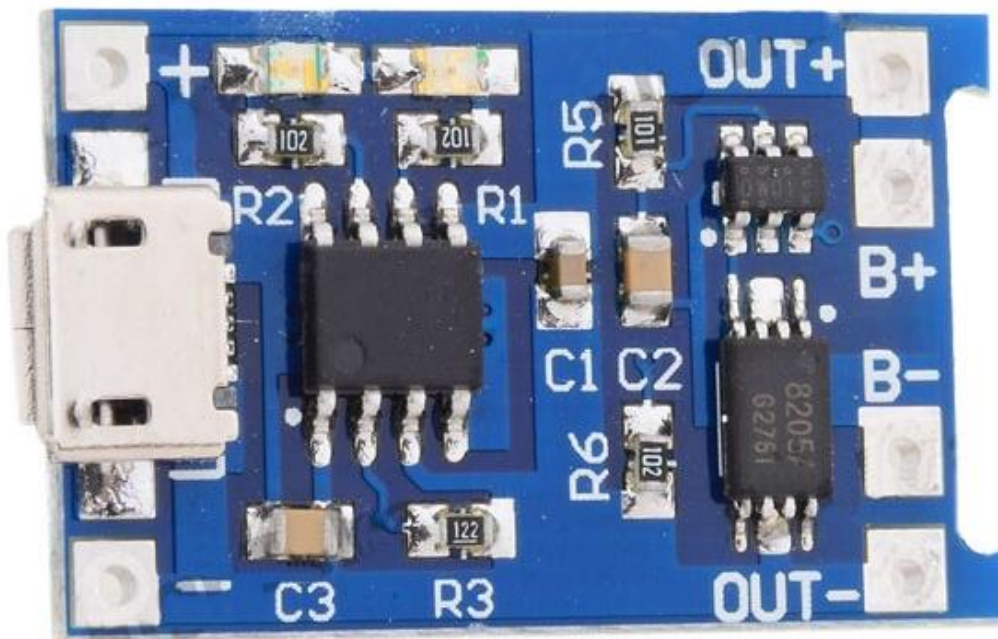


Figura 31 Módulo de carga micro-USB para batería de litio. Fuente: Tomado de Didácticas electrónicas (<https://n9.cl/2o0u2>)

Características:

- Voltaje de entrada: 5 V
- Voltaje de apagado: 4,2 V + / - 1 %
- Máxima corriente de carga: 1000 mA

- Voltaje de protección de descarga: 2,5 V
 - Protección de Sobre corriente de la batería: 3 A
 - Tamaño: 2.6 * 1.7 CM
-
- Controlador para los dispositivos nodos ESP8266 – ESP12F



Figura 32 Módulo ESP8266 - ESP12F. Fuente: Tomado de Didácticas Electrónicas (<https://n9.cl/icrx>)

Características:

- Pines compatibles con el ESP-12E
 - Soporta protocolo IEEE802.11 b/g/n.
 - Mejoras en la distancia de comunicación comparada con el ESP-12E
 - Antena integrada (PCB)
 - Voltaje de operación: 3.0 ~ 3.6V
 - Dimensiones: 24x16mm
-
- Sensor de presión barométrica BME280

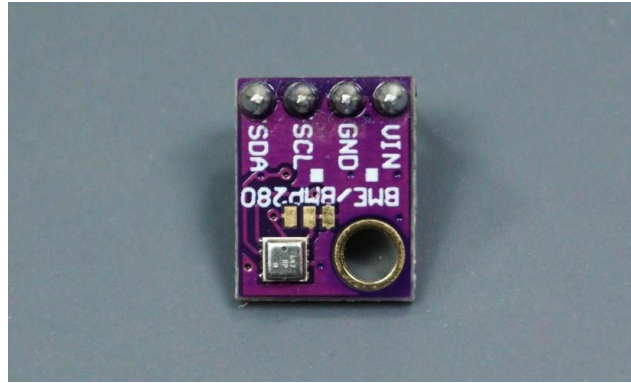


Figura 33 Sensor BMP280. Fuente: Tomado de randomnerdtutorials.com (<https://n9.cl/aew75>)

Características:

- Rango de medición de temperatura: - 40°C ~ 85°C
 - Sensor: Bosch BMP280
 - Voltaje de operación: 3V/5V
 - Interface de comunicación: I2C/SPI
 - Dimensiones: 19.2mm x 17.9mm x 2.9mm
- Sensor de intensidad lumínica ambiental DF Robots



Figura 34 Sensor de intensidad lumínica ambiental. Fuente: Tomado de dfrobot.com (https://wiki.dfrobot.com/DFRobot_Ambient_Light_Sensor_SKU_DFR0026)

Características:

- Alimentación de 3.3 V a 5V
 - Rango de iluminación de 1 lux a 6000 lux
 - Tiempo de respuesta de 15 us
 - Interfaz análoga
 - Tamaño 22 X 30 mm
-
- Controlador para los dispositivos ESP32



Figura 35 Módulo ESP32. Fuente: Tomado de Didácticas Electrónicas (<https://n9.cl/p1lls>)

Características:

- WiFi 802.11/b/g/n/e/I
 - Bluetooth 4.2 BR/EDR y BLE
 - Oscilador interno de 8MHz con calibración
 - Oscilador RC interno con calibración
 - Oscilador externo de cristal de 2MHz ~ 40MHz
 - Oscilador externo de cristal de 32KHz para calibración del RTC
 - Dos grupos de temporizadores, que incluyen dos temporizadores de 64 bits y un watchdog por grupo
 - Watchdog RTC
 - Dimensiones: 1.6cm x2.4cm x 0.3cm
-
- Módulo lector de microSD compatible con Arduino

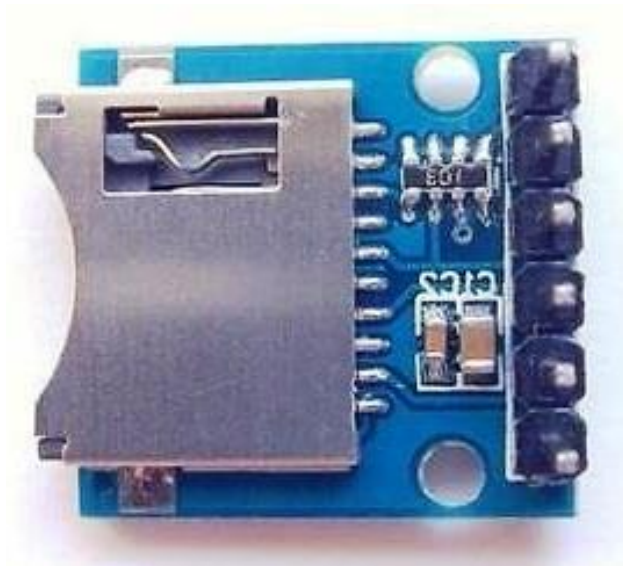


Figura 36 Módulo lector de microSD. Fuente: Tomado de Didácticas Electrónicas (<https://n9.cl/jwkm>)

Características:

- Dimensiones: 18.5mm x 17.5mm
- Compatible con Arduino

7.2 Adquisición de los sensores para el monitoreo de las variables climatológicas

Para la adquisición de los componentes se validaron varios proveedores de componentes electrónicos en los que por costos y disponibilidad se encuentran las referencias anteriormente mencionados:

- Didácticas Electrónicas - <https://didacticaselectronicas.com/>
- ElectroniLab - <http://electronilab.co/>
- TdRobotic - <http://tdrobotica.co/>
- Vistrónica - <https://www.vistronica.com/>
- Sigma Electrónica - <http://sigmaelectronica.net/>

7.3 Implementación e integración de los sensores adquiridos

En la etapa de implementación de los sensores luego de ser adquiridos en los correspondientes proveedores, se procedió a realizar las pruebas correspondientes a cada componente con mitas a tener las pruebas básicas de cada elemento para validar su funcionamiento y prever la manera en que se integrarán todos los sensores.

7.3.1 Sensor de temperatura y humedad relativa – DHT11.

Para el sensor de temperatura y humedad se presenta a continuación las pruebas realizadas partiendo del diagrama esquemático y el código de prueba:

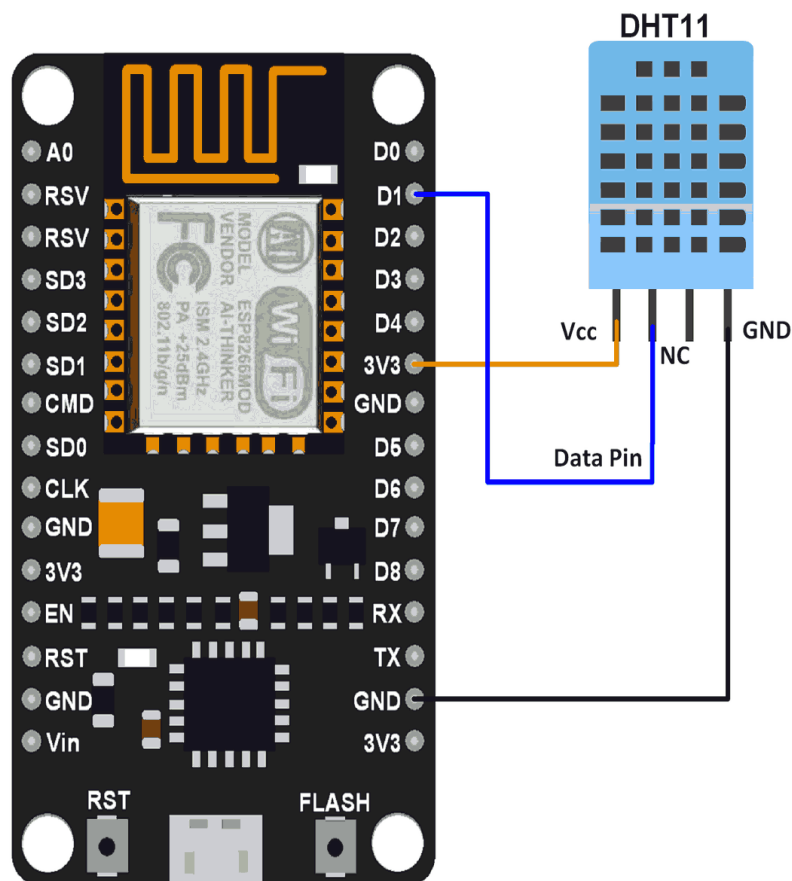
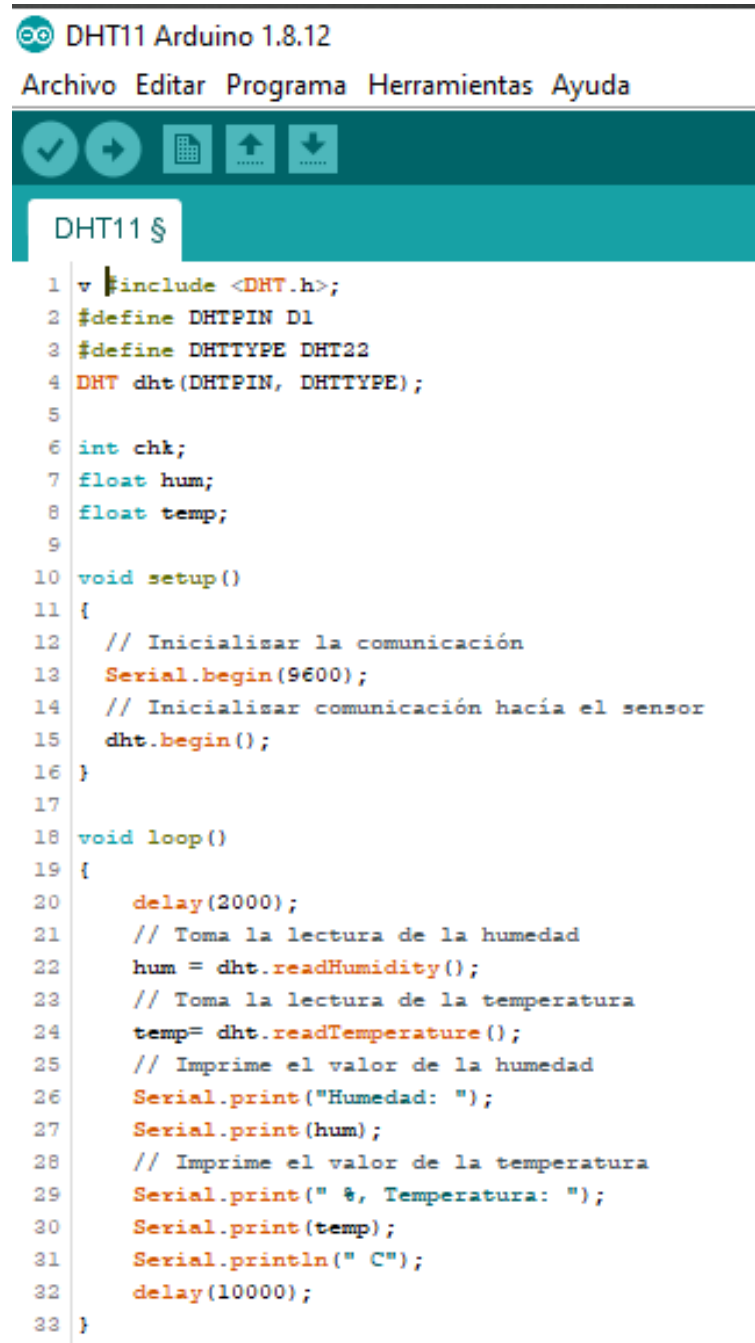


Figura 37 Diagrama de conexión del sensor DHT11. Nota 6 Tomado de ElectronicWings (<https://www.electronicwings.com/nodemcu/dht11-sensor-interfacing-with-nodemcu>)



```

DHT11 §
1  v #include <DHT.h>;
2  #define DHTPIN D1
3  #define DHTTYPE DHT22
4  DHT dht(DHTPIN, DHTTYPE);
5
6  int chk;
7  float hum;
8  float temp;
9
10 void setup()
11 {
12   // Inicializar la comunicación
13   Serial.begin(9600);
14   // Inicializar comunicación hacia el sensor
15   dht.begin();
16 }
17
18 void loop()
19 {
20   delay(2000);
21   // Toma la lectura de la humedad
22   hum = dht.readHumidity();
23   // Toma la lectura de la temperatura
24   temp= dht.readTemperature();
25   // Imprime el valor de la humedad
26   Serial.print("Humedad: ");
27   Serial.print(hum);
28   // Imprime el valor de la temperatura
29   Serial.print(" º, Temperatura: ");
30   Serial.print(temp);
31   Serial.println(" C");
32   delay(10000);
33 }

```

Figura 38 Código sensado de humedad y temperatura relativa utilizando el sensor DHT11. Fuente:
Elaboración propia.

7.3.2 Sensor de humedad en tierra – YI100.

EL sensor de humedad en tierra YI100 realiza su operación de manera análoga, generando los datos que se obtienen de la humedad en tierra a partir de un canal ADC todo como se muestra a continuación en el diagrama y código funcional.

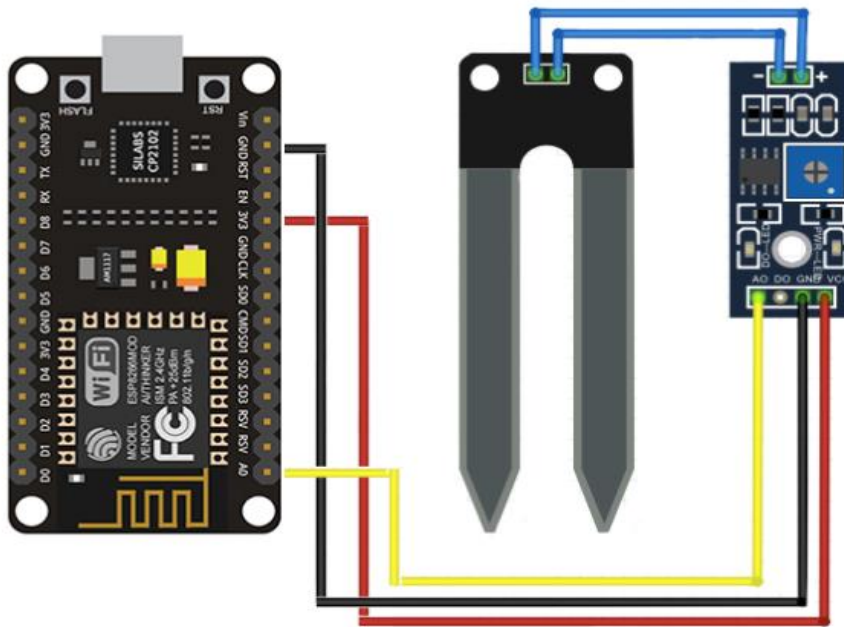


Figura 39 Diagrama de conexión del sensor YI100. Fuente: Tomado de Hackster.io (<https://www.hackster.io/amruthp/soil-moisture-sensor-arduino-tutorial-ffcb5b>)

```

YI100 Arduino 1.8.12
Archivo Editar Programa Herramientas Ayuda
YI100
1 // Variable que almacenará la lectura análoga
2 int output_value ;
3 void setup() {
4   // Inicializar su comunicación Serial
5   Serial.begin(115200);
6   Serial.println("Leyendo el sensor...");
7   // Espera dos segundos para inicial la lectura del sensor
8   delay(2000);
9 }
10 void loop() {
11   // Asigna la variable a la lectura análogo
12   output_value = analogRead(sensor_pin);
13   // Imprimir el valor de la humedad
14   Serial.print("Humedad : ");
15   Serial.print(output_value);
16   Serial.print("\n");
17   delay(1000);
18 }

```

Figura 40 Código de lectura del sensor YI100 Arduino. Fuente: Elaboración propia.

7.3.3 Sensor de lluvia – YC69.

El sensor de lluvia opera con un pin digital el cual en el momento en que detecta agua en su superficie indica un pulso bajo para realizar la detección o el control, a continuación, se describirá el proceso a realizar para el uso de dicho sensor y la codificación de este:

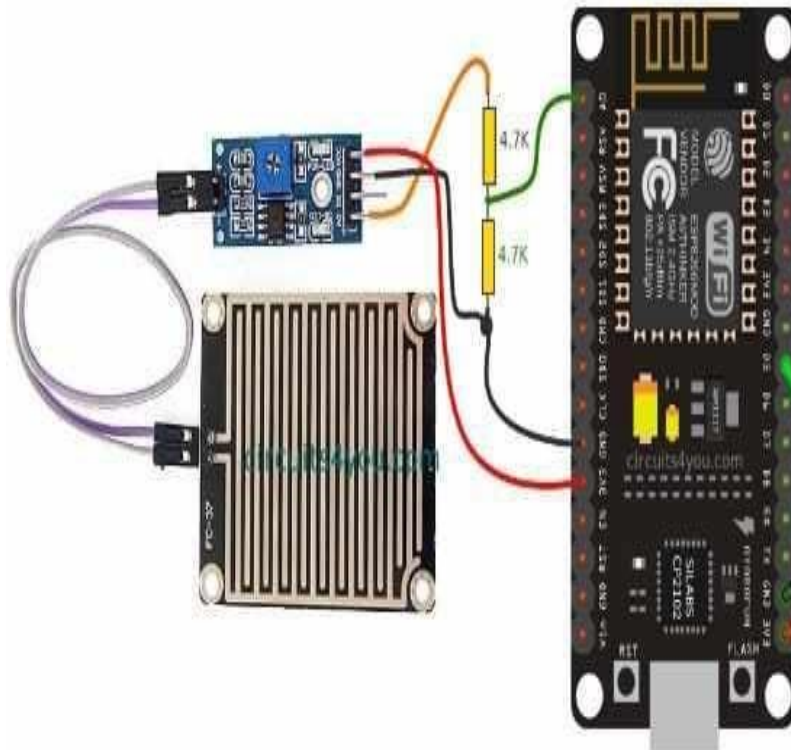
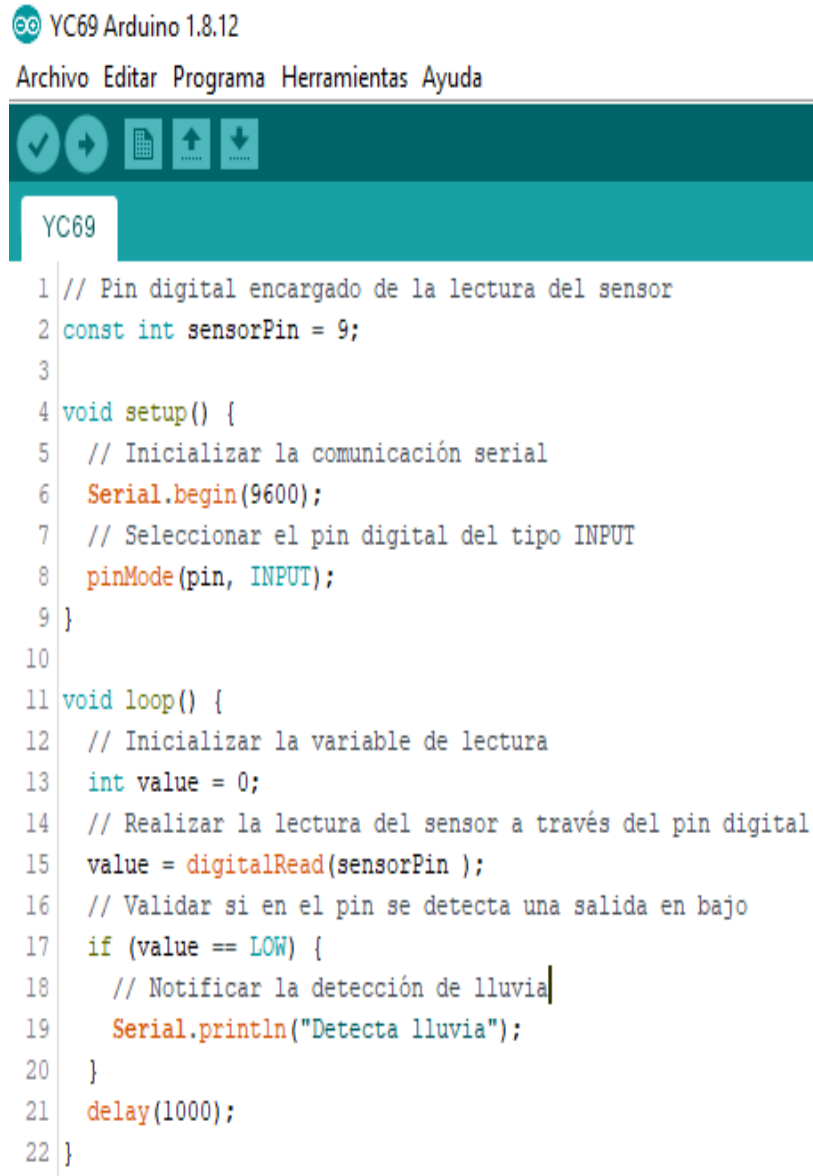


Figura 41 Diagrama de conexión del sensor YC69. Nota 7 Tomado de Descubre Arduino (<https://descubrearduino.com/construir-una-estacion-meteorologica-con-esp8266/>)



```

1 // Pin digital encargado de la lectura del sensor
2 const int sensorPin = 9;
3
4 void setup() {
5   // Inicializar la comunicación serial
6   Serial.begin(9600);
7   // Seleccionar el pin digital del tipo INPUT
8   pinMode(sensorPin, INPUT);
9 }
10
11 void loop() {
12   // Inicializar la variable de lectura
13   int value = 0;
14   // Realizar la lectura del sensor a través del pin digital
15   value = digitalRead(sensorPin );
16   // Validar si en el pin se detecta una salida en bajo
17   if (value == LOW) {
18     // Notificar la detección de lluvia
19     Serial.println("Detecta lluvia");
20   }
21   delay(1000);
22 }

```

Figura 42 Código para realizar la lectura del sensor de lluvia YC69. Fuente: Elaboración propia.

7.3.4 Sensor de intensidad lumínica ambiental.

El sensor de intensidad lumínica ambiental v2 cuenta con una interfaz de comunicación analógica fácil de implementar, además de contar con una alimentación desde los 3.3V hasta los 5V lo cual facilita la adopción de este sensor, este sensor hace parte del dispositivo Nodo.

A continuación, se presentan los esquemáticos de conexión del mismo, así como el código de prueba utilizado para validar el funcionamiento.

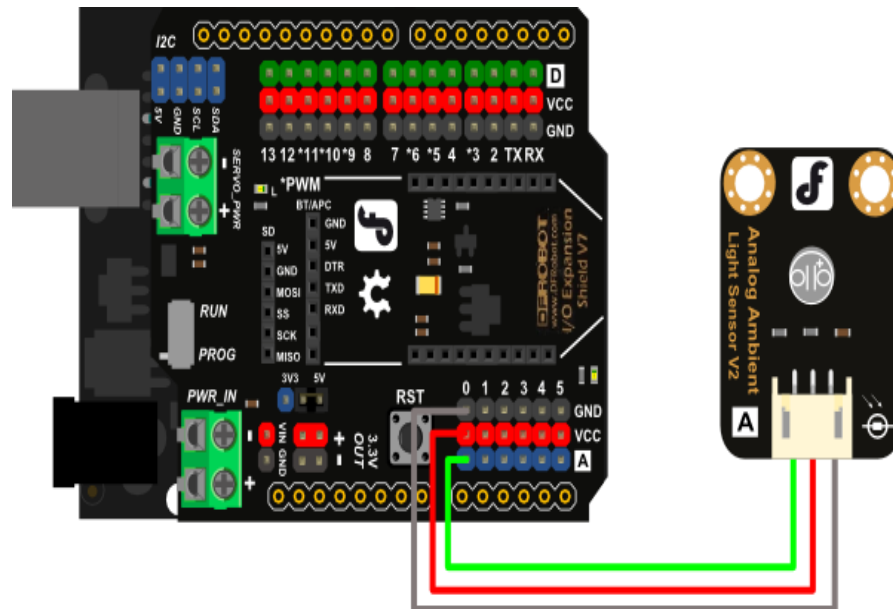


Figura 43 Esquemático de la conexión del sensor. Fuente: Tomado de RF Robot (https://wiki.dfrobot.com/DFRobot_Ambient_Light_Sensor_SKU_DFR0026)

```

Iluminacion-ambiental Arduino 1.8.12
Archivo Editar Programa Herramientas Ayuda
Iluminacion-ambiental
1 void setup()
2 {
3   // Configurar la comunicación serial
4   Serial.begin(115200);
5 }
6 void loop()
7 {
8   // Leer el valor del sensor a través del puerto analógico 0
9   int sensor = analogRead(0);
10  // Imprimir el valor alojado en la variable sensor.
11  Serial.println(sensor, DEC);
12  // Esperar 100 ms para tomar una nueva muestra
13  delay(100);
14 }

```

Figura 44 Código correspondiente a la prueba del sensor de intensidad lumínica ambiental. Fuente: Elaboración propia.

7.3.5 Sensor de presión barométrica - BMP280.

El sensor de presión barométrica cuenta con una interfaz de comunicación i2c con la cual mediante la dirección de los registros podemos obtener los valores que está midiendo el mismo para luego ser procesados y enviados al dispositivo Base.

En la prueba realizada de funcionamiento mínimo del sensor se realizó la conexión de la siguiente manera:

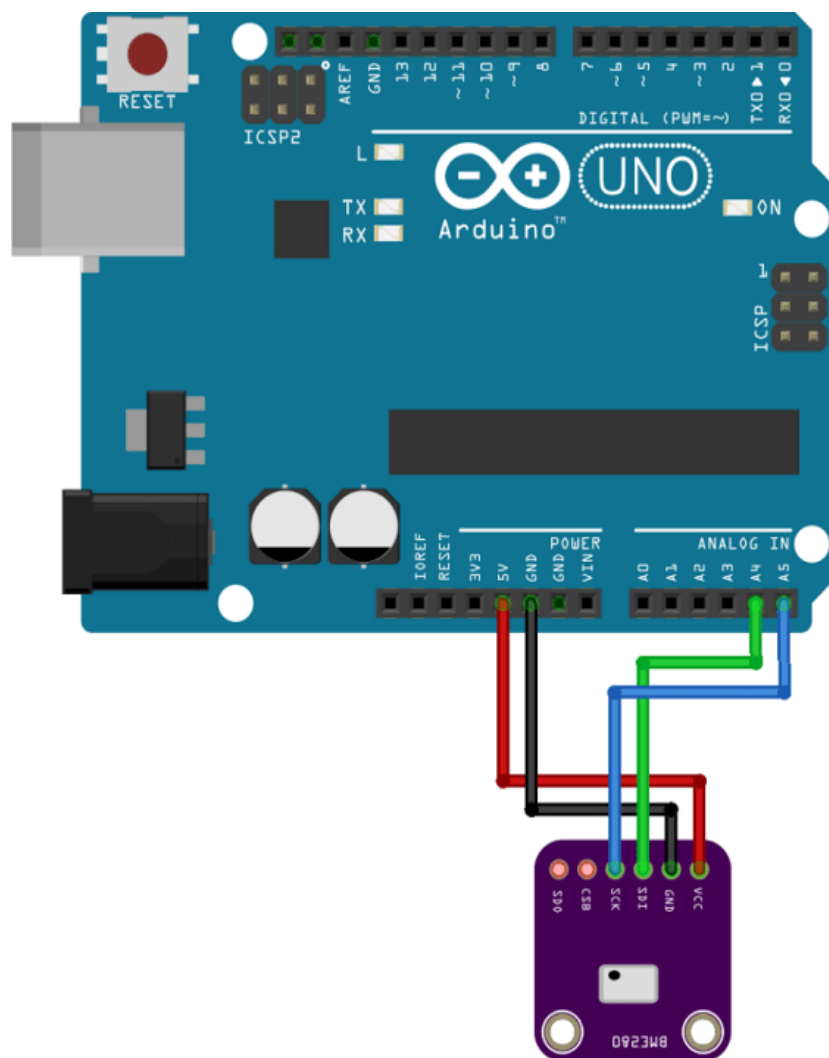
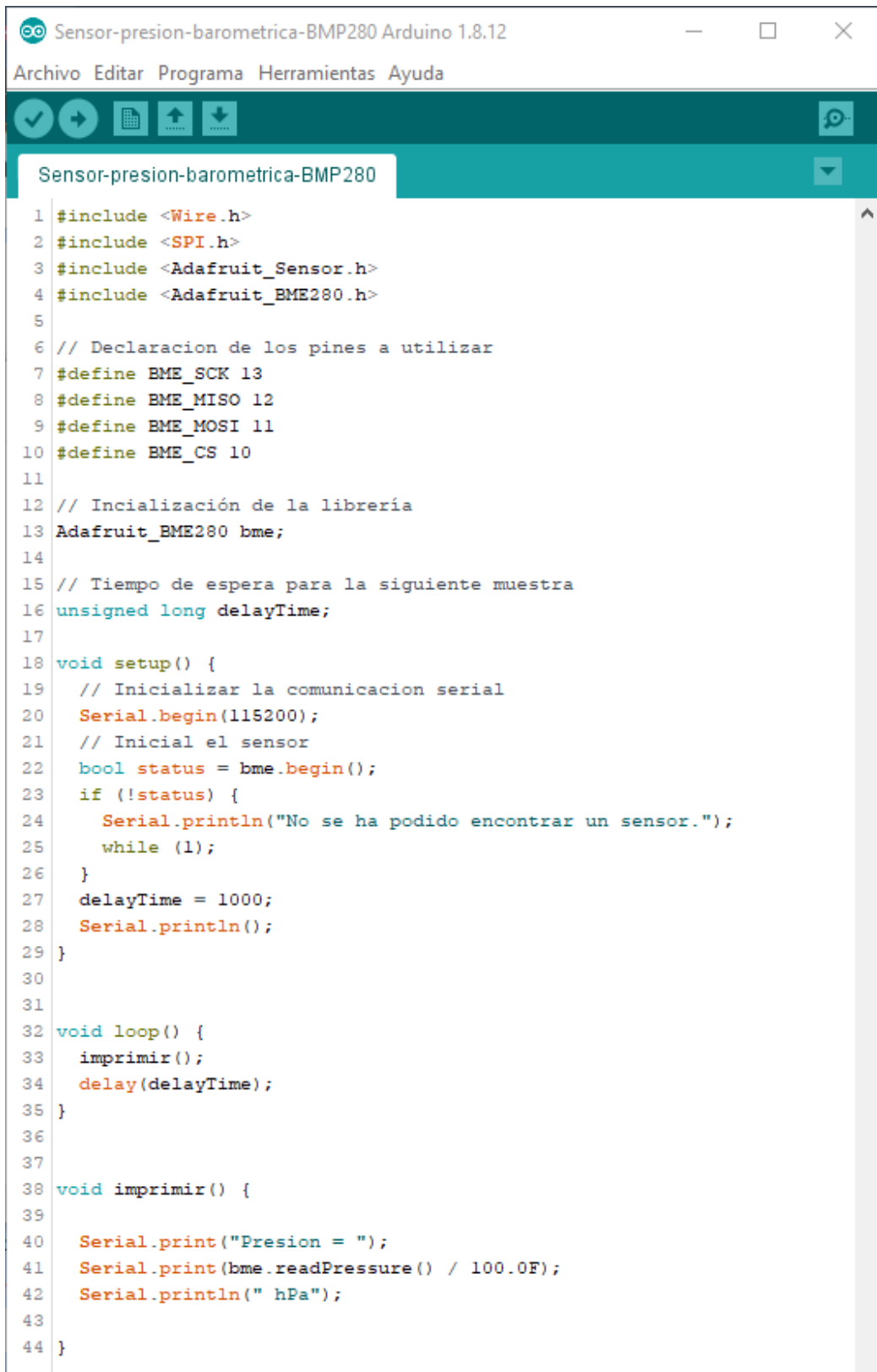


Figura 45 Esquemático de conexión del sensor BMP280 Nota 8 Tomado de Radomnerdtutorials.com (<https://randomnerdtutorials.com/bme280-sensor-arduino-pressure-temperature-humidity/>)



```
1 #include <Wire.h>
2 #include <SPI.h>
3 #include <Adafruit_Sensor.h>
4 #include <Adafruit_BME280.h>
5
6 // Declaracion de los pines a utilizar
7 #define BME_SCK 13
8 #define BME_MISO 12
9 #define BME_MOSI 11
10 #define BME_CS 10
11
12 // Inicialización de la librería
13 Adafruit_BME280 bme;
14
15 // Tiempo de espera para la siguiente muestra
16 unsigned long delayTime;
17
18 void setup() {
19   // Inicializar la comunicacion serial
20   Serial.begin(115200);
21   // Inicial el sensor
22   bool status = bme.begin();
23   if (!status) {
24     Serial.println("No se ha podido encontrar un sensor.");
25     while (1);
26   }
27   delayTime = 1000;
28   Serial.println();
29 }
30
31
32 void loop() {
33   imprimir();
34   delay(delayTime);
35 }
36
37
38 void imprimir() {
39
40   Serial.print("Presion = ");
41   Serial.print(bme.readPressure() / 100.0F);
42   Serial.println(" hPa");
43
44 }
```

Figura 46 Código de prueba del sensor BMP280. Fuente: Elaboración propia.

7.3.6 Módulo lector de microSD

El módulo de lectura de microSD se enfoca en dos ejercicios que funcionan sobre el mismo esquema de conexión, uno realizar la lectura de los registros en la memoria y otro en la escritura de la memoria.

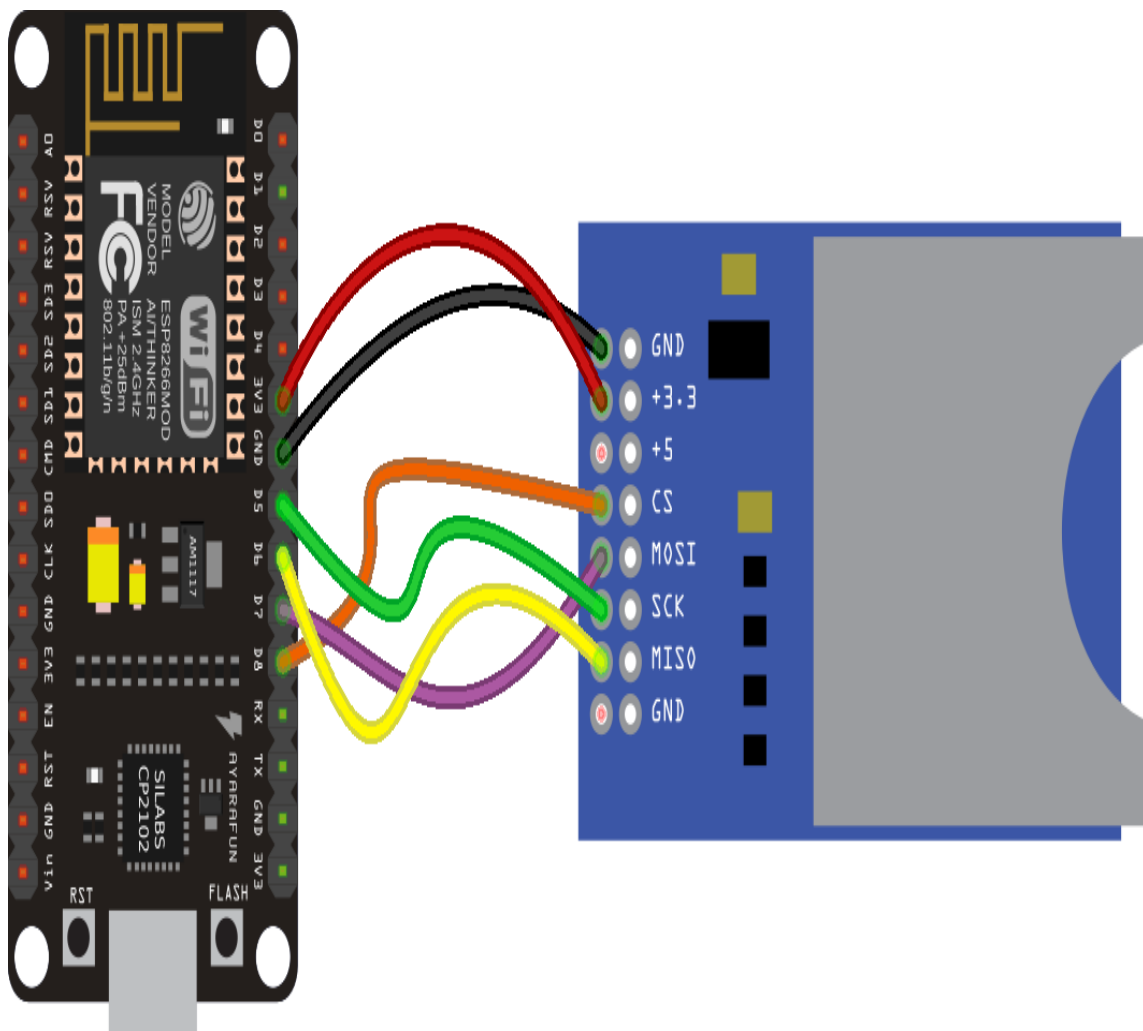



Figura 47 Diagrama de conexión del módulo de lectura de microSD Nota 9 Tomado de Github (<https://github.com/lvidarte/esp8266/blob/master/examples/sdcard/README.md>)


 Lector_SD_-_Lectura Arduino 1.8.12

Archivo Editar Programa Herramientas Ayuda



```
Lector_SD_-_Lectura
1 #include <SD.h>
2
3 File dataFile;
4
5 void setup()
6 {
7   Serial.begin(9600);
8   Serial.print(F("Iniciando SD ..."));
9   if (!SD.begin(9))
10  {
11    Serial.println(F("Error al iniciar"));
12    return;
13  }
14  Serial.println(F("Iniciado correctamente"));
15
16  // Abrir fichero y mostrar el resultado
17  dataFile = SD.open("datalog.txt");
18  if (dataFile)
19  {
20    string dataLine;
21    while (dataFile.available())
22    {
23      dataLine = dataFile.read();
24      Serial.write(dataLine);
25    }
26    dataFile.close();
27  }
28  else
29  {
30    Serial.println(F("Error al abrir el archivo"));
31  }
32 }
33
34 void loop()
35 {
36 }
```

Figura 48 Código correspondiente a la lectura de los registros en la microSD. Fuente: Elaboración propia.

 Lector_SD_-_Escritura Arduino 1.8.12
 Archivo Editar Programa Herramientas Ayuda



```

1 #include <SD.h>
2 File logFile;
3 void setup()
4 {
5   Serial.begin(9600);
6   Serial.print(F("Iniciando SD ..."));
7   if (!SD.begin(9))
8   {
9     Serial.println(F("Error al iniciar"));
10    return;
11  }
12  Serial.println(F("Iniciado correctamente"));
13 }
14
15
16 // Funcion que simula la lectura de un sensor
17 int readSensor()
18 {
19   return 0;
20 }
21
22 void loop()
23 {
24   // Abrir archivo y escribir valor
25   logFile = SD.open("datalog.txt", FILE_WRITE);
26
27   if (logFile) {
28     int value = readSensor;
29     logFile.print("Time(ms)=");
30     logFile.print(millis());
31     logFile.print(", value=");
32     logFile.println(value);
33     logFile.close();
34
35   }
36   else {
37     Serial.println("Error al abrir el archivo");
38   }
39   delay(500);
40 }
  
```

Figura 49 Código de escritura de registro en microSD. Fuente: Elaboración propia.

7.3.7 Panel solar.

En cuanto al módulo del panel solar, este genera la energía necesaria para poder funcionar de manera autónoma, en este proceso se vincula también la batería de litio y el módulo de carga de la batería de litio.

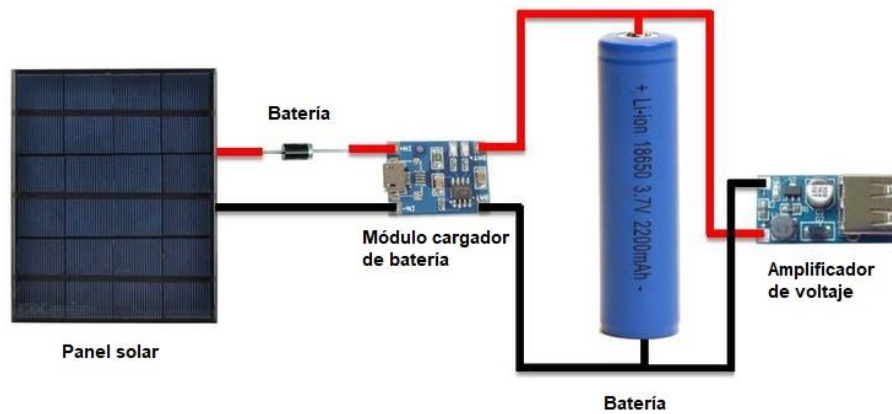


Figura 50 Diagrama de conexión del panel solar, la batería y el módulo de carga micro-USB Nota 10 Tomado de Arduino.cc (<https://create.arduino.cc/projecthub/igorF2/solar-charged-battery-powered-arduino-uno-645d89>)

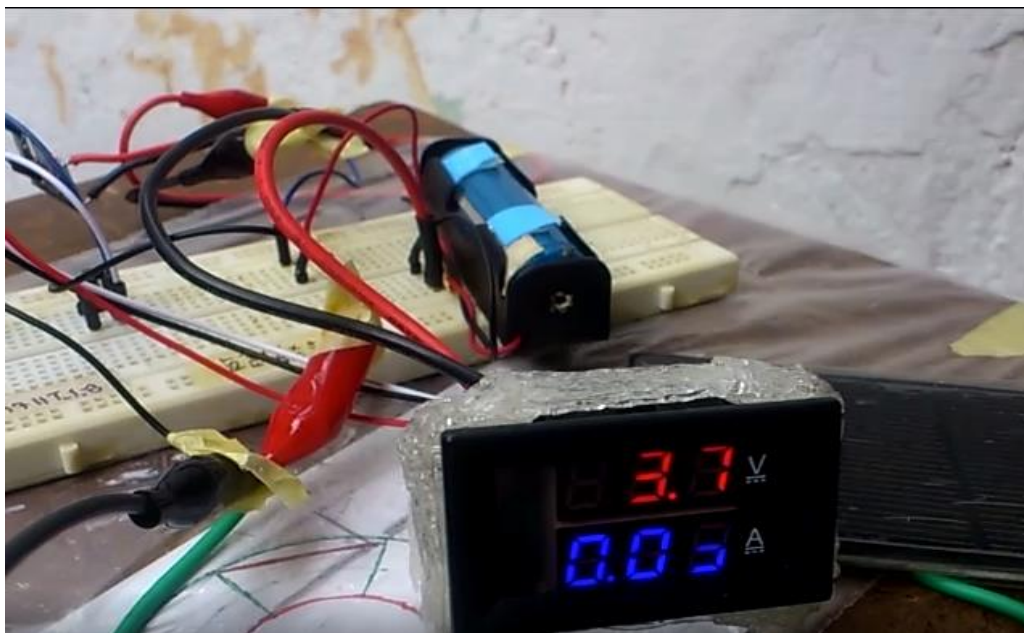


Figura 51 Prueba realizada cargando la batería de litio con el panel solar. Fuente: Elaboración propia.

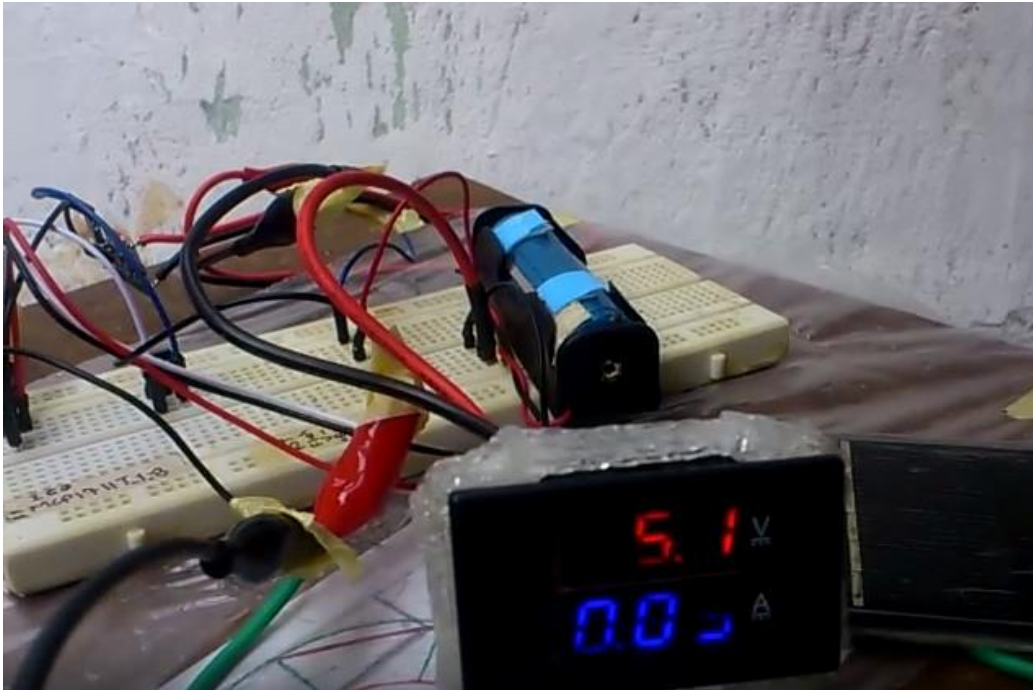


Figura 52 Prueba realizada cargando la batería de litio con el panel solar. Fuente: Elaboración propia.

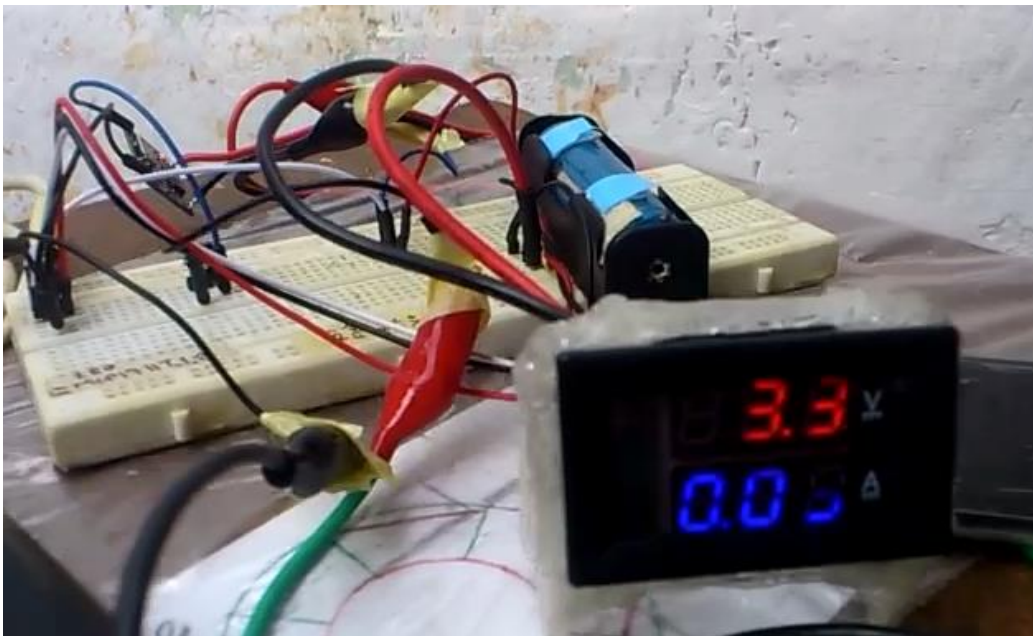


Figura 53 Prueba realizada cargando la batería de litio con el panel solar. Fuente: Elaboración propia.

7.3.8 Módulo de comunicación LoRA02

El módulo de comunicación LoRA02 destaca de su tamaño, bajo costo, su interfaz de comunicación SPI y el uso de la banda de 433MHz son parte de sus características, a continuación, se muestra el diagrama de conexión y el código para el transmisor y el código de recepción.

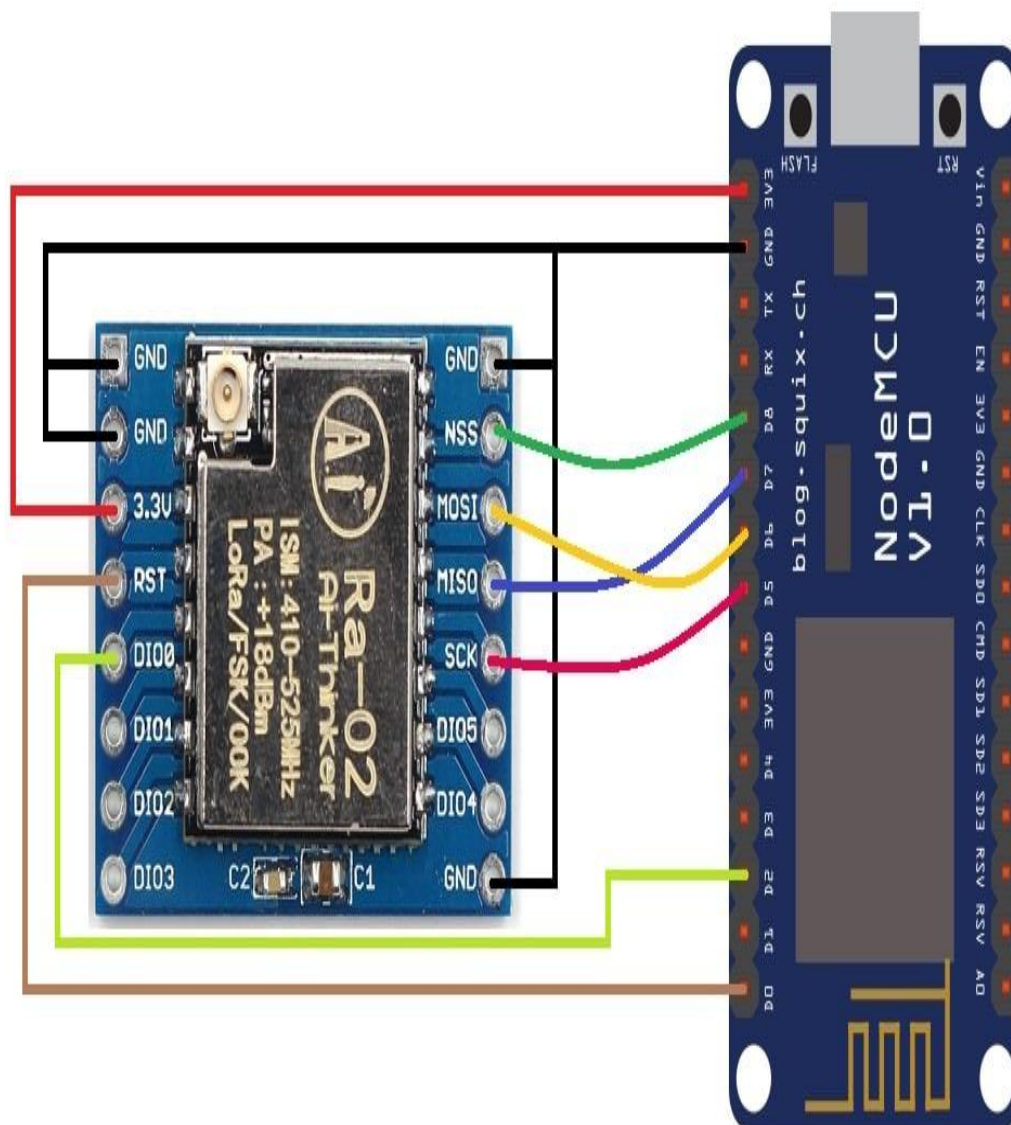
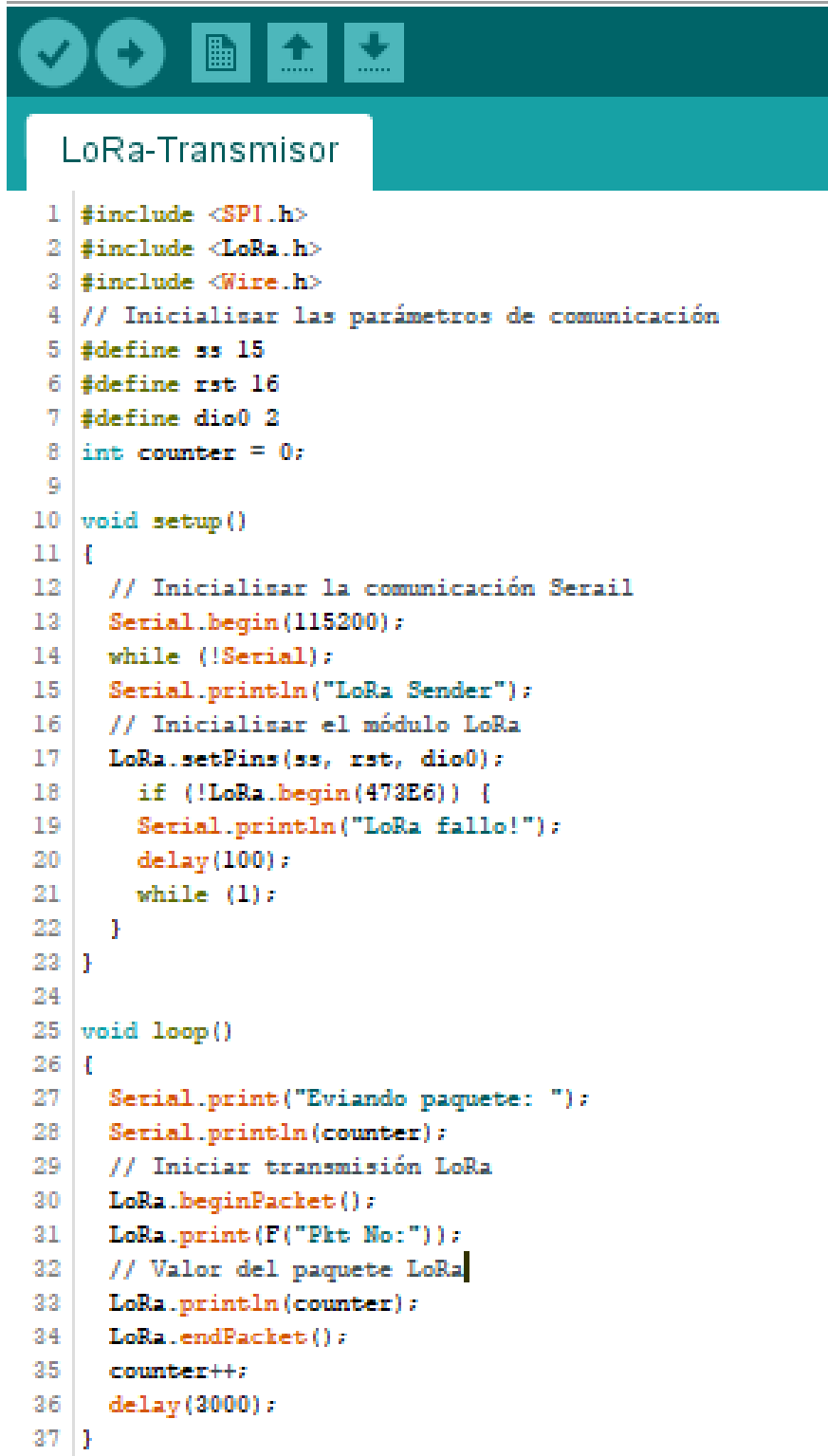


Figura 54 Diagrama de conexión LoRa 02. Fuente: Tomado de How2electronics (<https://how2electronics.com/lora-sx1278-esp8266-transmitter-receiver/>)

LoRa-Transmisor Arduino 1.8.12

Archivo Editar Programa Herramientas Ayuda



```

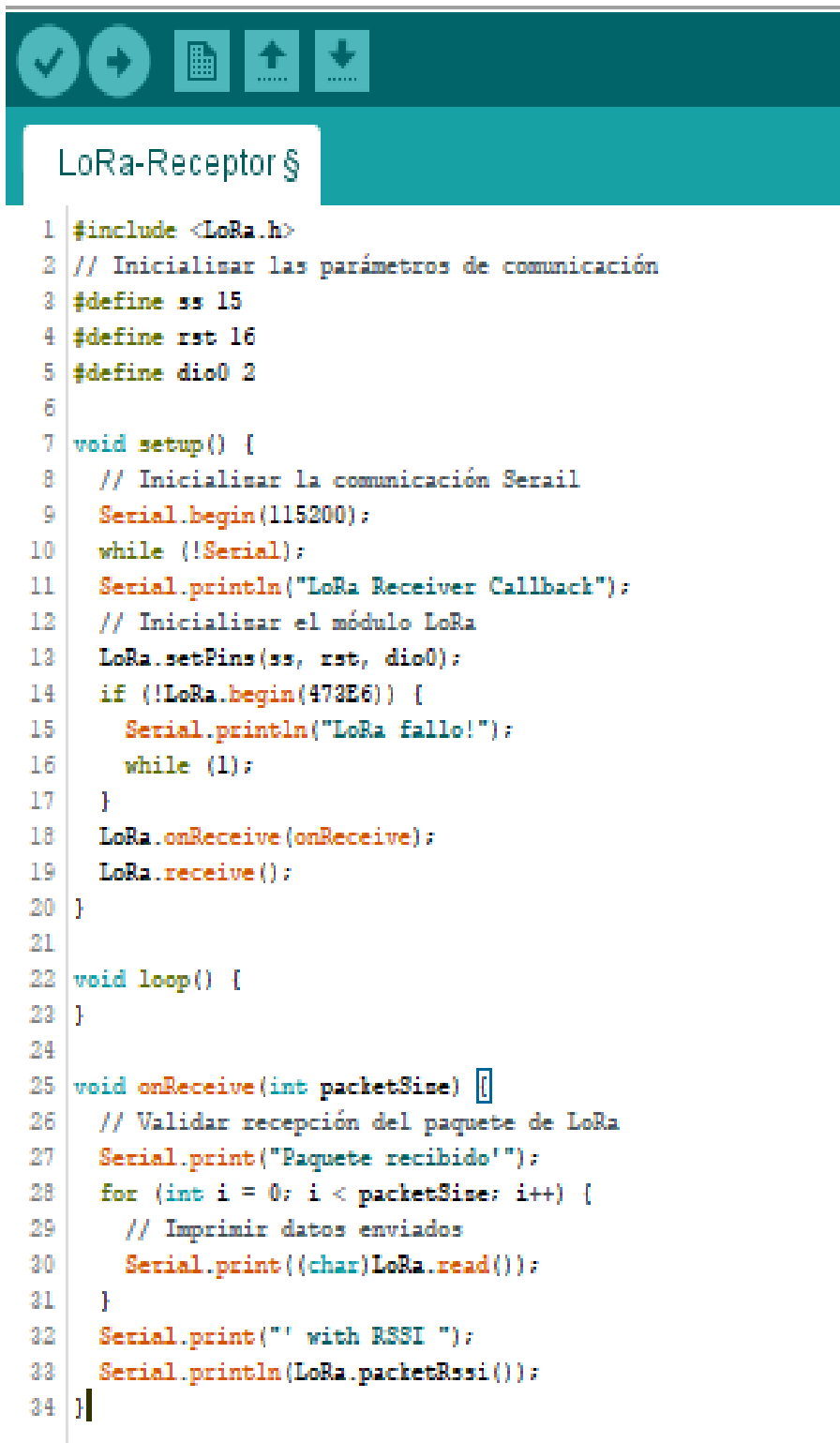
1  #include <SPI.h>
2  #include <LoRa.h>
3  #include <Wire.h>
4  // Inicializar las parámetros de comunicación
5  #define ss 15
6  #define rst 16
7  #define dio0 2
8  int counter = 0;
9
10 void setup()
11 {
12   // Inicializar la comunicación Serial
13   Serial.begin(115200);
14   while (!Serial);
15   Serial.println("LoRa Sender");
16   // Inicializar el módulo LoRa
17   LoRa.setPins(ss, rst, dio0);
18   if (!LoRa.begin(473E6)) {
19     Serial.println("LoRa fallo!");
20     delay(100);
21     while (1);
22   }
23 }
24
25 void loop()
26 {
27   Serial.print("Eviando paquete: ");
28   Serial.println(counter);
29   // Iniciar transmisión LoRa
30   LoRa.beginPacket();
31   LoRa.print(F("Pkt No:"));
32   // Valor del paquete LoRa
33   LoRa.println(counter);
34   LoRa.endPacket();
35   counter++;
36   delay(3000);
37 }

```

Figura 55 Código del transmisor LoRa02. Fuente: Elaboración propia.

LoRa-Receptor Arduino 1.8.12

Archivo Editar Programa Herramientas Ayuda



```

1 #include <LoRa.h>
2 // Inicializar los parámetros de comunicación
3 #define ss 15
4 #define rst 16
5 #define dio0 2
6
7 void setup() {
8   // Inicializar la comunicación Serial
9   Serial.begin(115200);
10  while (!Serial);
11  Serial.println("LoRa Receiver Callback");
12  // Inicializar el módulo LoRa
13  LoRa.setPins(ss, rst, dio0);
14  if (!LoRa.begin(473E6)) {
15    Serial.println("LoRa fallo!");
16    while (1);
17  }
18  LoRa.onReceive(onReceive);
19  LoRa.receive();
20 }
21
22 void loop() {
23 }
24
25 void onReceive(int packetSize) {
26   // Validar recepción del paquete de LoRa
27   Serial.print("Paquete recibido");
28   for (int i = 0; i < packetSize; i++) {
29     // Imprimir datos enviados
30     Serial.print((char)LoRa.read());
31   }
32   Serial.print("' with RSSI ");
33   Serial.println(LoRa.packetRssi());
34 }

```

Figura 56 Código del receptor LoRa02. Fuente: Elaboración propia.

7.4 Implementación del protocolo de comunicación entre los dispositivos

En la implementación del protocolo de comunicación, para este caso LoRa y MQTT se realizaron las siguientes pruebas correspondientes al alcance y funcionalidad obteniendo los siguientes resultados:

7.4.1 LoRa.

En la implementación del protocolo LoRa se realizaron una serie de pruebas para validar el alcance de los controladores en los cuales se logra un alcance en zona urbana de 350m. lo que es acorde a la distancia propuesta para las pruebas del proyecto, para la realización de esta prueba se utilizó de una parte el ESP8266 junto con el módulo LoRa02 y de otra parte se utilizó el módulo ESP32 junto con el módulo LoRa02, esto con el fin de establecer la comunicación en dicho protocolo, en cuánto al servidor MQTT se utilizó el acceso público brindado por <https://www.hivemq.com/public-mqtt-broker/> utilizando los parámetros de conexión :

- Broker: broker.hivemq.com
- TCP Port: 1883
- Websocket Port: 8000
- Topic: test

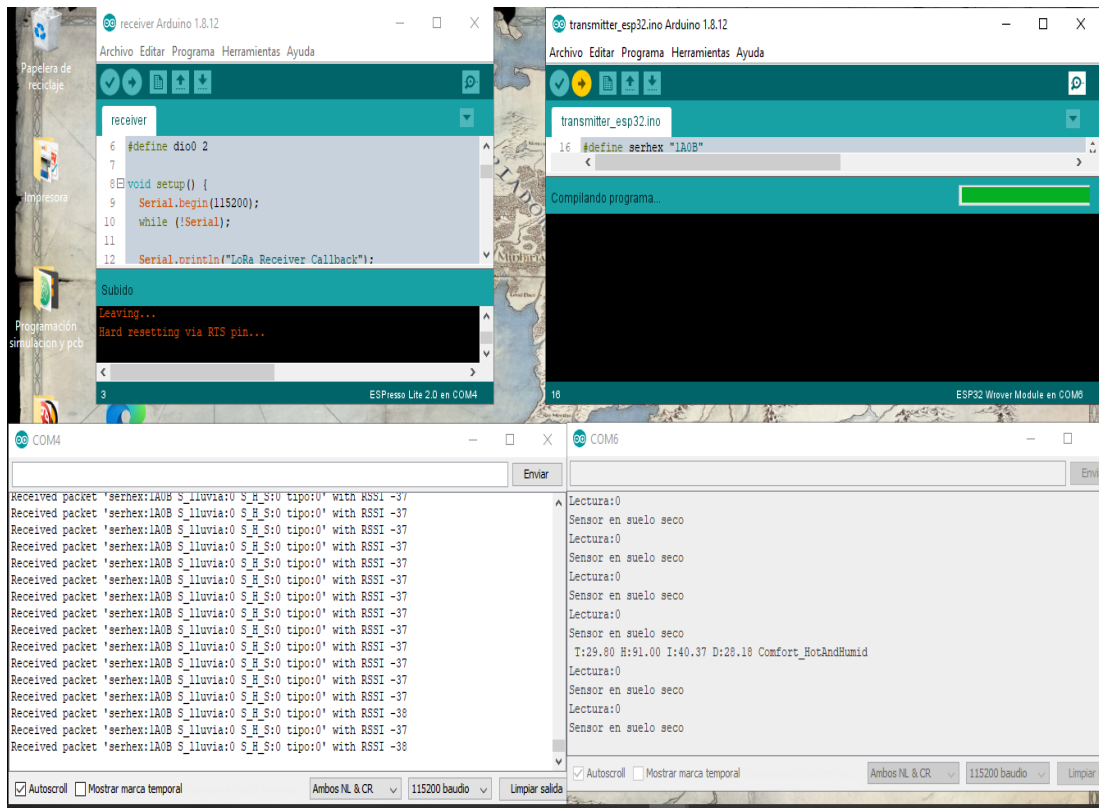


Figura 57 Prueba de comunicación entre módulos LoRa. Fuente: Elaboración propia.

```

{"id": "1A0B", "lluvia": "0", "H_suelo": "1456", "tipo": "2"}
{"id": "1A0B", "lluvia": "0", "H_suelo": "1462", "tipo": "2"}
{"id": "1A0B", "lluvia": "0", "H_suelo": "1456", "tipo": "2"}
{"id": "1A0B", "lluvia": "0", "H_suelo": "1460", "tipo": "2"}
{"id": "1A0B", "lluvia": "0", "H_suelo": "1458", "tipo": "2"}
{"id": "1A0B", "lluvia": "0", "H_suelo": "1456", "tipo": "2"}
{"id": "1A0B", "lluvia": "0", "H_suelo": "1459", "tipo": "2"}
{"id": "1A0B", "lluvia": "0", "H_suelo": "1460", "tipo": "2"}
{"id": "1A0B", "lluvia": "0", "H_suelo": "1459", "tipo": "2"}
{"id": "1A0B", "lluvia": "0", "H_suelo": "1460", "tipo": "2"}
{"id": "1A0B", "lluvia": "0", "H_suelo": "1460", "tipo": "2"}
{"id": "1A0B", "lluvia": "0", "H_suelo": "1456", "tipo": "2"}
{"id": "1A0B", "lluvia": "0", "H_suelo": "1461", "tipo": "2"}
{"id": "1A0B", "lluvia": "0", "H_suelo": "1456", "tipo": "2"}
{"id": "1A0B", "lluvia": "0", "H_suelo": "1457", "tipo": "2"}
{"id": "1A0B", "lluvia": "0", "H_suelo": "1456", "tipo": "2"}

```

Figura 58 Log de los registros de la comunicación en el dispositivo Base. Fuente: Elaboración propia.

7.4.2 MQTT

En la comunicación MQTT se realizó la prueba en la que el dispositivo nodo emite los valores al dispositivo base y este emite los datos vía MQTT, para la realización de esta prueba se utilizó los módulos ESP8266, ESP32 y dos módulos de comunicación LoRA02.

The image shows two windows from the Arduino IDE. The left window, titled 'COM6', displays a series of serial output messages. The messages include data points (dato1, dato2, dato3) and status updates like 'escribiendo:', 'hecho', and 'Paquet recu'. At the bottom, it shows 'Attempting MQTT connection...connected'. The right window, titled 'receiver2 Arduino 1.8.12', shows the C++ code for the receiver. The code includes a function 'void mqtt_json()' that checks for a client connection, reconnects if needed, and sends data via MQTT when a specific WiFi status is reached. Below the code, the 'Subido' (Upload) log shows the compilation and upload process, including memory usage and a successful upload of 348128 bytes.

```

COM6
dato0: 1A0B
dato1: 0
dato2: 1461
dato3: 2
escribiendo:
hecho
Paquet recu
dato0: 1A0B
dato1: 0
dato2: 1456
dato3: 2
escribiendo:
hecho
Paquet recu
dato0: 1A0B
dato1: 0
dato2: 1457
dato3: 2
escribiendo:
hecho
Paquet recu
dato0: 1A0B
dato1: 0
dato2: 1456
dato3: 2
escribiendo:
hecho
Paquet recu
dato0: 1A0B
dato1: 0
dato2: 1457
dato3: 2
Paquet recu
dato0: 1A0B
dato1: 0
dato2: 1456
dato3: 2
Attempting MQTT connection...connected
Paquet recu
dato0: 1A0B
dato1: 0
dato2: 1456
dato3: 2

receiver2
76 // }
77 // }
78
79 void mqtt_json()
80 {
81   if (!client.connected())
82     {reconnect();}
83   client.loop();
84   if (WiFi.status() == 3) {
85     long now = millis();
86     long corte = now - lastMsg;
87     if (corte > 10000)
88     {
89       lastMsg = now;
90       if (DEBUG == true)
91       {
92         control_sd_datos=0;msgl(true);contl=0;
93       }
94     }
95   }
Subido
Writing at 0x0002c000... (75 %)
Writing at 0x00030000... (81 %)
Writing at 0x00034000... (87 %)
Writing at 0x00038000... (93 %)
Writing at 0x0003c000... (100 %)
Wrote 348128 bytes (251772 compressed) at 0x00000000 in 22.3 seconds (effective 124
Hash of data verified.
Leaving...
Hard resetting via RTS pin...

```

Figura 59 Prueba de envío de datos del dispositivo base a servidor vía MQTT. Fuente: Elaboración propia.

7.4.3 Google Sheet

Adicionalmente a enviar la información vía MQTT se realizó la integración para tener el registro de los datos obtenidos por los dispositivos en una hoja de Google Sheet, a continuación, se detalla la prueba realizada en Arduino para emitir los datos a dicha hoja:



```

1 #include <ESP8266WiFi.h>
2 #include <WiFiManager.h>
3
4 #define SERHEX          "2A0B"
5 #define SERVER2        "api.pushingbox.com"
6 #define PORT2          80
7 WiFiClient Client2;
8
9 int cont1 =0;
10 String serhexT;
11 String sumar;
12 String clima_temperatura;
13 String clima_humedad;
14 String clima_presion;
15
16 //-----inicializacion-----
17 void setup() {
18   Serial.begin(115200);
19   //-----wifi-----
20   WiFi.mode(WIFI_STA);
21   WiFiManager wifiManager;
22   char combinedArray[20];
23   char array1[10] = "[test] ";
24   sprintf(combinedArray, "%s %s", array1, SERHEX); // With word space
25   wifiManager.autoConnect(combinedArray);
26   //-----Lora-----
27 }
28
29 void loop()
30 {
31
32   if (!Client2.connect(SERVER2, PORT2))
33   {
34     Serial.println("connection failed");
35     return;
36   }
37   serhexT = "01AB";
38   sumar = "10";
39   clima_temperatura = "30.2";
40   clima_humedad = "50.4";
41   clima_presion = "93.4";
42   // We now create a URI for the request
43   String url = "/pushingbox?";
44   url += "devid=";
45   url += "v4AFE9910E00D4AC";
46   url += "&id_0="+String(SERHEX);
47   url += "&id_1="+serhexT;
48   url += "&sensores_0="+sumar;
49   url += "&sensores_1="+clima_temperatura;
50   url += "&sensores_2="+clima_humedad;
51   url += "&sensores_3="+clima_presion;
52   Serial.print("Requesting URL: ");
53   Serial.println(url);
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Figura 60 Código de prueba envío de datos a Google Sheet - parte 1. Fuente: Elaboración propia.

```

55 // This will send the request to the server
56 Client2.print(String("GET ") + url + " HTTP/1.1\r\n" + "Host: " + SERVER2 + "\r\n" + "Connection: close\r\n\r\n");
57 unsigned long timeout = millis();
58 while (Client2.available() == 0) {
59   if (millis() - timeout > 4000) {
60     Serial.println(">>> Client2 Timeout !");
61     Client2.stop();
62     return;
63   }
64 }
65
66 // Read all the lines of the reply from server and print them to Serial
67 while(Client2.available()){
68   String line = Client2.readStringUntil('\r');
69   Serial.print(line);
70   Serial.println("Data Sent!");
71 }
72 Serial.println("closing connection");
73 delay(600000);
74 }

```

Figura 61 Código de prueba envío de datos a Google Sheet - parte 2. Fuente: Elaboración propia.

7.5 Construcción del prototipo hardware

Luego de tener las pruebas exitosas en la etapa previa, se procede a realizar las pruebas de cada componente se procedió a consolidar cada uno de los dispositivos que dan forma al objetivo del presente proyecto.

7.5.1 Dispositivo nodo

Este dispositivo es el encargado de realizar la toma de las mediciones de los sensores y posteriormente los envía al dispositivo base a través del protocolo LoRa, este dispositivo cuenta con las siguientes características:

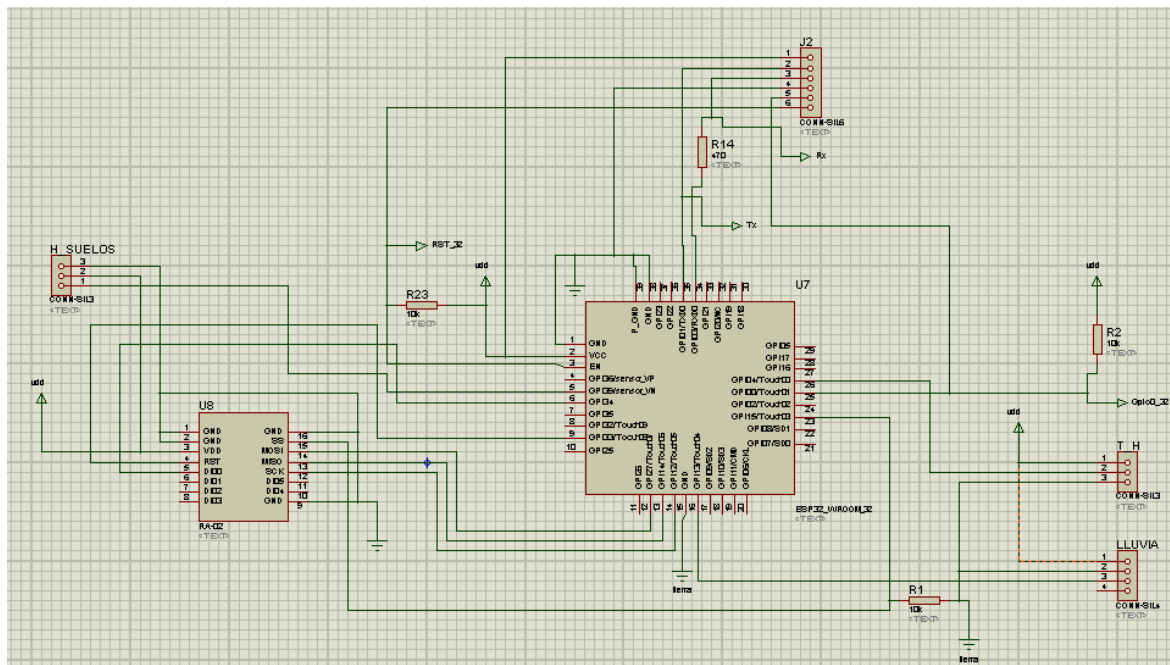


Figura 62 Esquemático del circuito correspondiente al dispositivo Nodo. Nota 11 Diseño realizado utilizando el software Proteus (R)

Los componentes descritos en el diagrama de Figura 62 se ubican de la siguiente manera:

- RA02: LoRa02
- H SUELOS: Sensor de humedad en tierra YI100
- ESP32_WROOM_32: Módulo ESP32
- COMM-SIL6: Pines de conexión serial
- T_H: Sensor de temperatura y humedad DHT-11
- LLUVIA: Sensor de lluvia YC69
- R1, R2, R23: Resistencia de 10K Ohm
- R14: Resistencia de 470 Ohm
- 1 panel solar
- 1 porta pilas para batería 18650
- 1 batería litio 3.7 v 2000 mA 18650

7.5.2 Dispositivo Base

Este dispositivo es el encargado de realizar la toma de los paquetes emitidos desde los nodos vía LoRa que contienen los datos de los sensores para posteriormente en caso de tener acceso a red enviar al servidor la información utilizando el protocolo MQTT o si por el contrario no se cuenta con acceso a red, los datos sean almacenados en la memoria microSD para aguardar al momento en que se tenga acceso a la red para enviar dichas tramas; este dispositivo cuenta con las siguientes características:

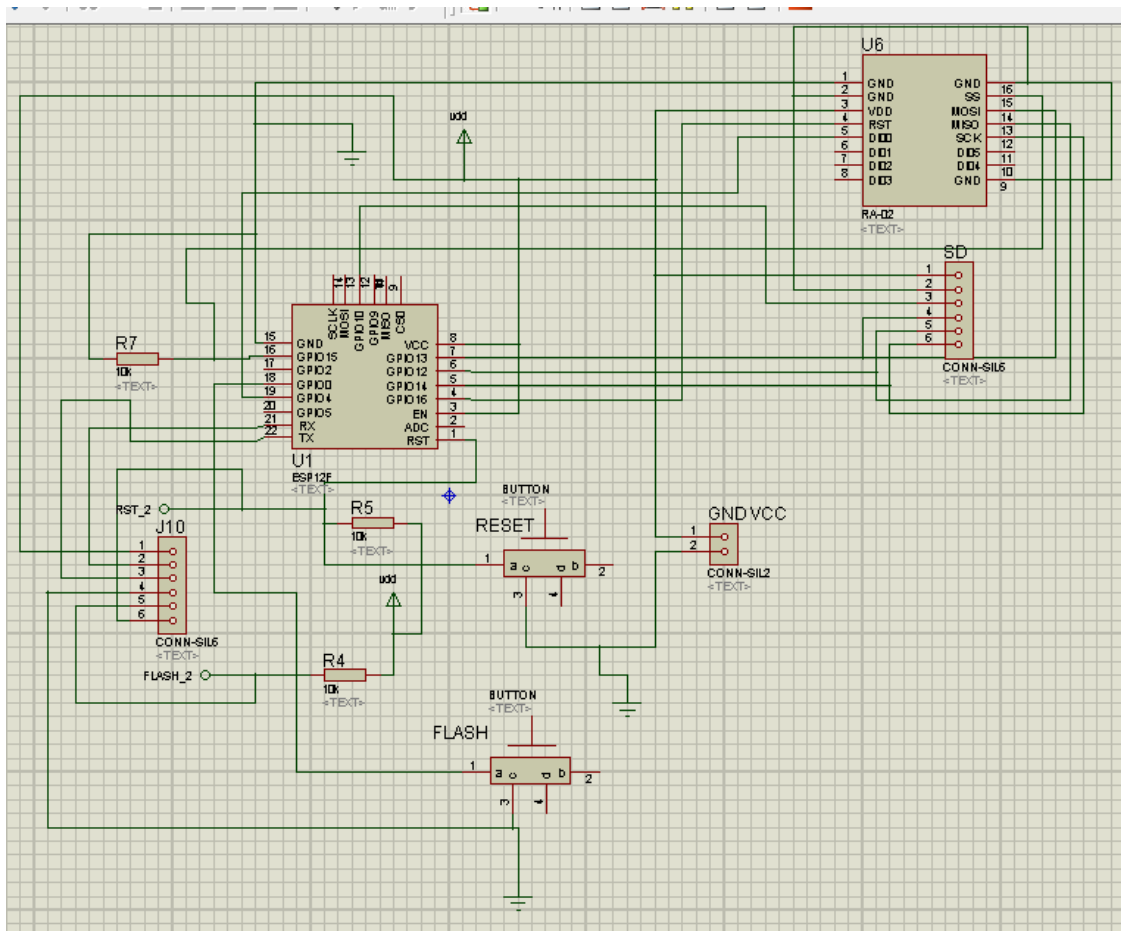


Figura 63 Esquemático del circuito correspondiente al dispositivo Base. Nota 12 Diseño realizado utilizando el software Proteus (R)

Los componentes descritos en el diagrama de Figura 63 se ubican de la siguiente manera:

- COMM-SIL6: Pines de conexión serial
- FLASH: Botón para realizar la programación del módulo ESP8266
- RESET: Botón para realizar reset físico
- ESP12F: Módulo ESP8266
- RA02: Módulo controlador LoRA02
- SD: Módulo lector de microSD
- GND-VCC: Ingreso de la alimentación
- R4, R5 y R7: Resistencia de 10K Ohm

7.5.3 Diseño PCB

El diseño de la PCB se realizó utilizando el software Proteus ®, dentro del proceso se le ha apuntó a generar una tarjeta de un tamaño pequeño llegando a obtener la PCB evidenciada en la Figura 62 *Esquemático del circuito correspondiente al dispositivo Nodo. Nota 11 Diseño realizado utilizando el software Proteus (R)* y Figura 63 *Esquemático del circuito correspondiente al dispositivo Base. Nota 12 Diseño realizado utilizando el software Proteus (R)* Figura 63, en esta, se realizó el diseño de los dos dispositivos pensando en hacer eficiente el espacio al momento de realizar la fabricación de estas.

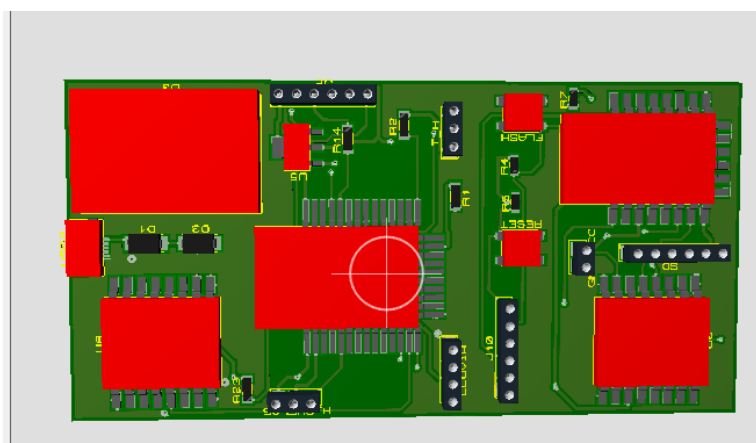


Figura 64 Diseño frontal de la PCB dispositivo Base y Dispositivo Nodo. Fuente: Elaboración propia

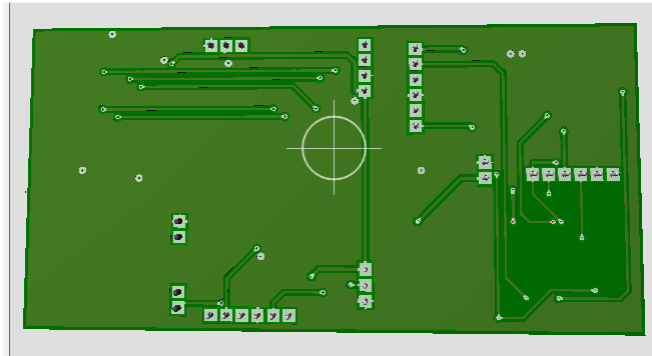


Figura 65 Diseño posterior de la PCB dispositivo Base y Dispositivo Nodo. Fuente: Elaboración propia

7.6 Medición de las variables monitoreadas

Luego de establecer los dos dispositivos se realizaron las pruebas de funcionamiento, comunicación y alcance de los dispositivos.

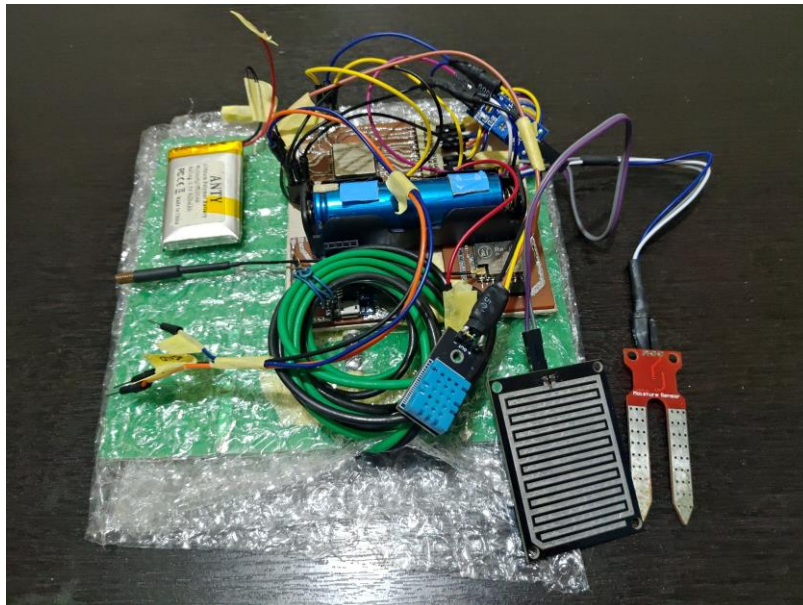


Figura 66 Dispositivo Nodo fuera del case. Fuente: Elaboración propia.

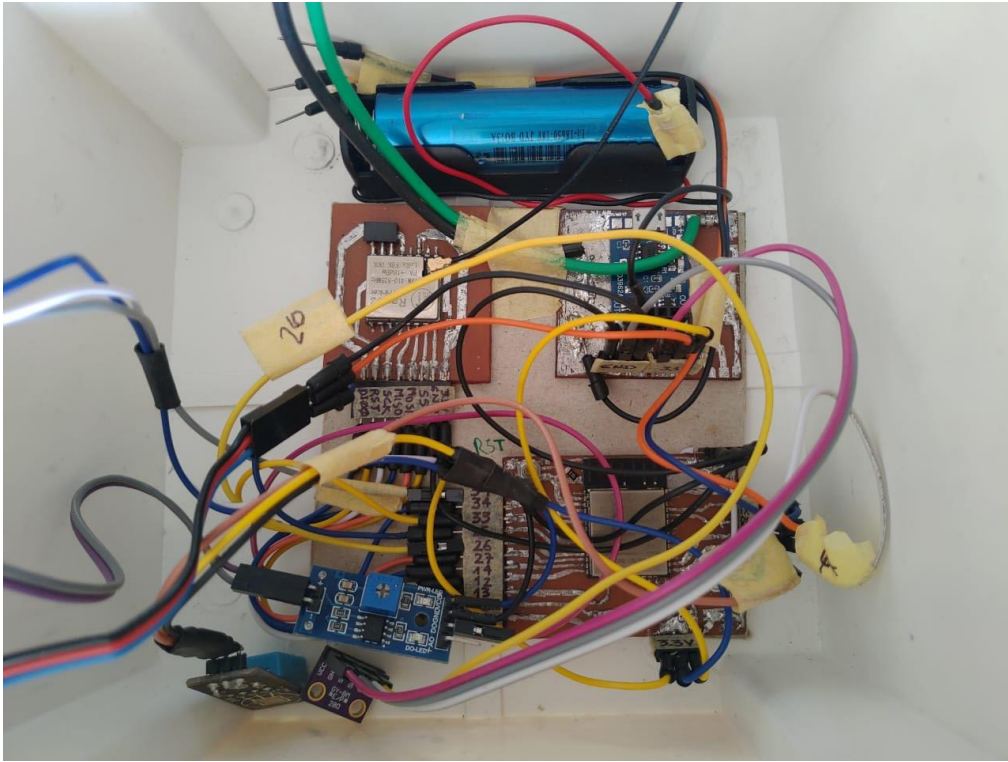


Figura 67 Dispositivo Nodo en case. Fuente: Elaboración propia.

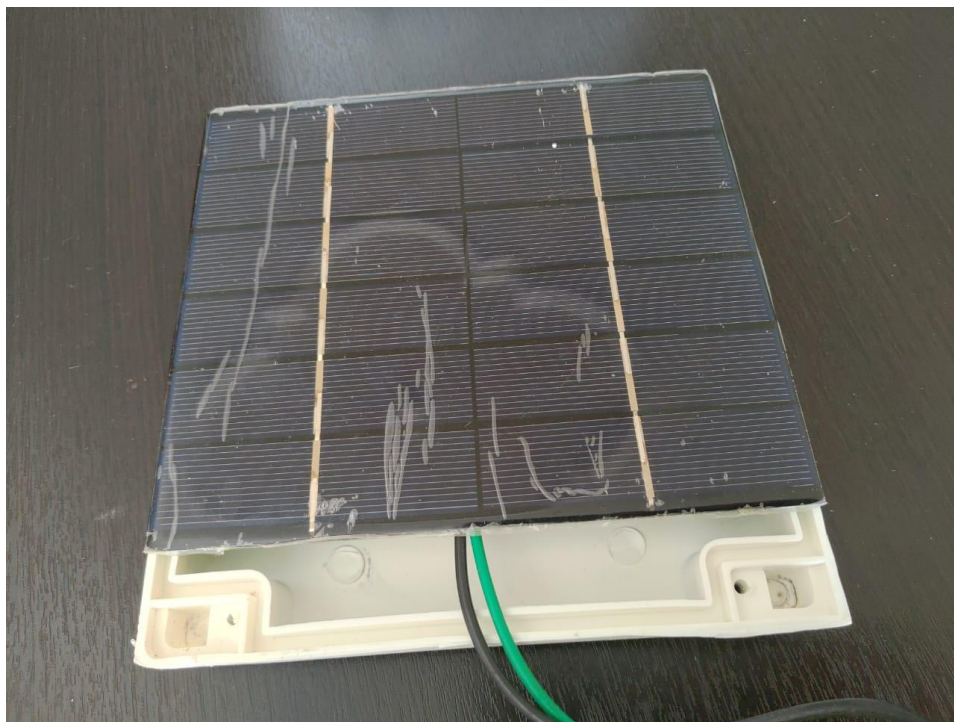


Figura 68 Panel solar del dispositivo Nodo. Fuente: Elaboración propia.

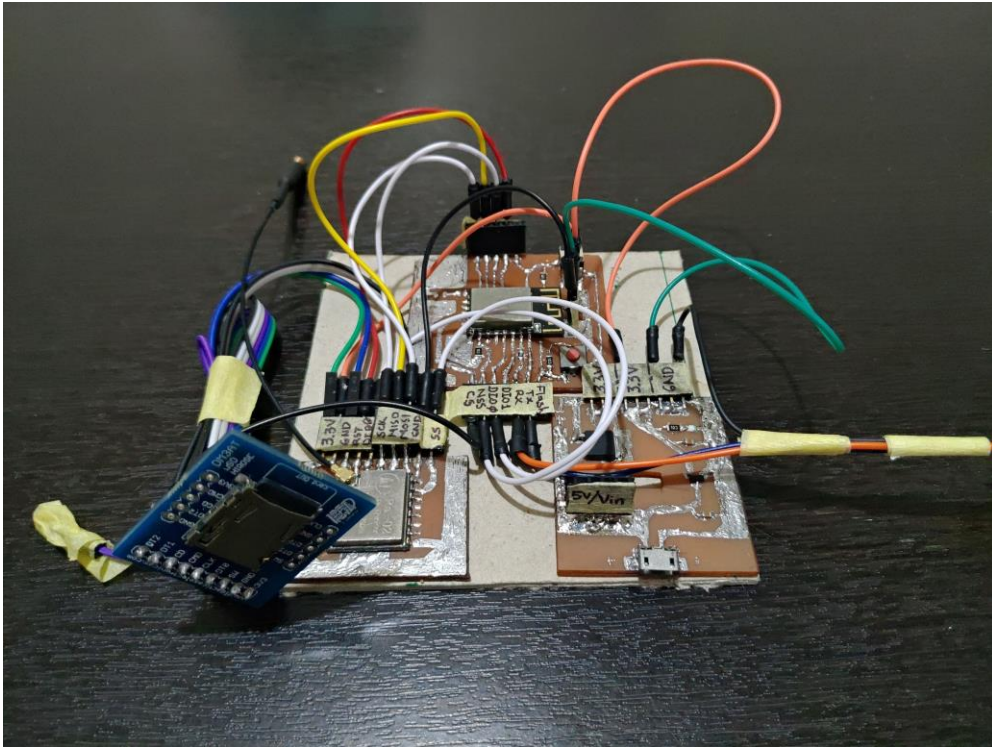


Figura 69 Dispositivo Base. Fuente: Elaboración propia.

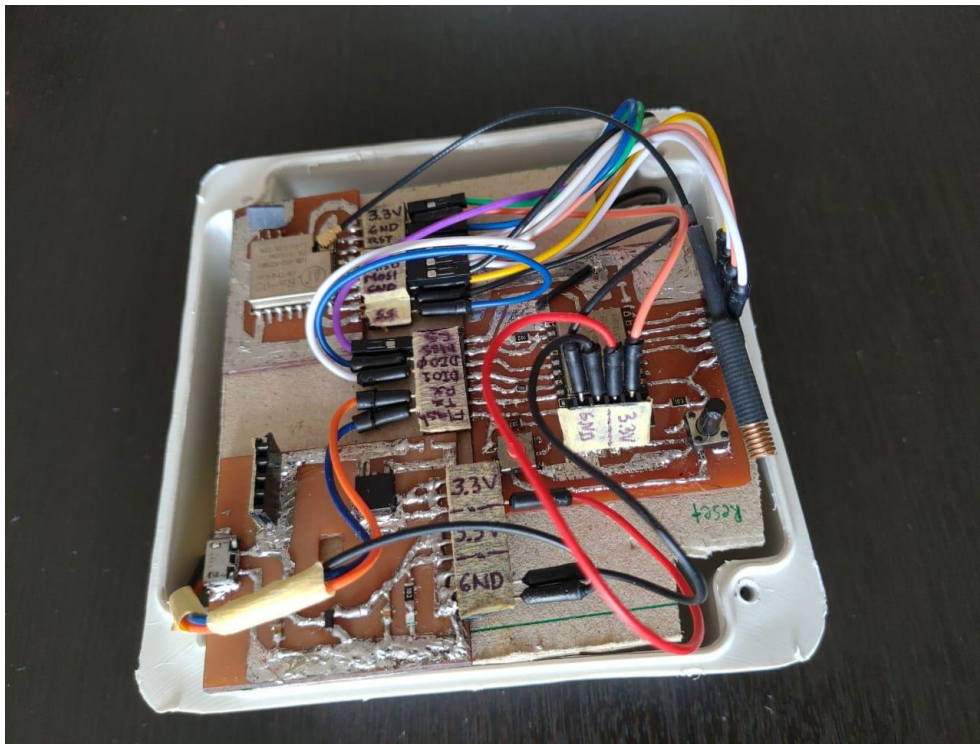


Figura 70 Dispositivo Base en case. Fuente: Elaboración propia.

	A	B	C	D	E	F
1	id_0	id_1	sensores_0	sensores_1	sensores_2	sensores_3
2	2A0B	01AB	10	30.2	50.4	93.4
3	2A0B	01AB	10	30.2	50.4	93.4
4	2A0B	01AB	10	30.2	50.4	93.4
5	2A0B	01AB	10	30.2	50.4	93.4
6	2A0B	01AB	10	30.2	50.4	93.4
7	2A0B	01AB	10	30.2	50.4	93.4
8	2A0B	01AB	10	30.2	50.4	93.4
9	2A0B	01AB	10	30.2	50.4	93.4
10	2A0B	01AB	10	30.2	50.4	93.4
11	2A0B	01AB	10	30.2	50.4	93.4
12	2A0B	01AB	10	30.2	50.4	93.4
13	2A0B	01AB	10	30.2	50.4	93.4
14	2A0B	01AB	10	30.2	50.4	93.4
15	2A0B	01AB	10	30.2	50.4	93.4
16	2A0B	01AB	10	30.2	50.4	93.4
17	2A0B	01AB	10	30.2	50.4	93.4
18	2A0B	01AB	10	30.2	50.4	93.4
19	2A0B	01AB	10	30.2	50.4	93.4
20	2A0B	01AB	10	30.2	50.4	93.4
21	2A0B	01AB	10	30.2	50.4	93.4
22	2A0B	01AB	10	30.2	50.4	93.4
23	2A0B	01AB	10	30.2	50.4	93.4
24	2A0B	01AB	10	30.2	50.4	93.4
25	2A0B	01AB	0	25.6	73	89.8
26	2A0B	01AB	0	25.6	73	89.8
27	2A0B	01AB	0	25.7	73	89.8
28	2A0B	01AB	0	25.6	73	89.8
29	2A0B	01AB	0	25.6	72	89.8
30						

Figura 71 Datos emitidos por el dispositivo base y almacenados en una hoja de Google Sheet.

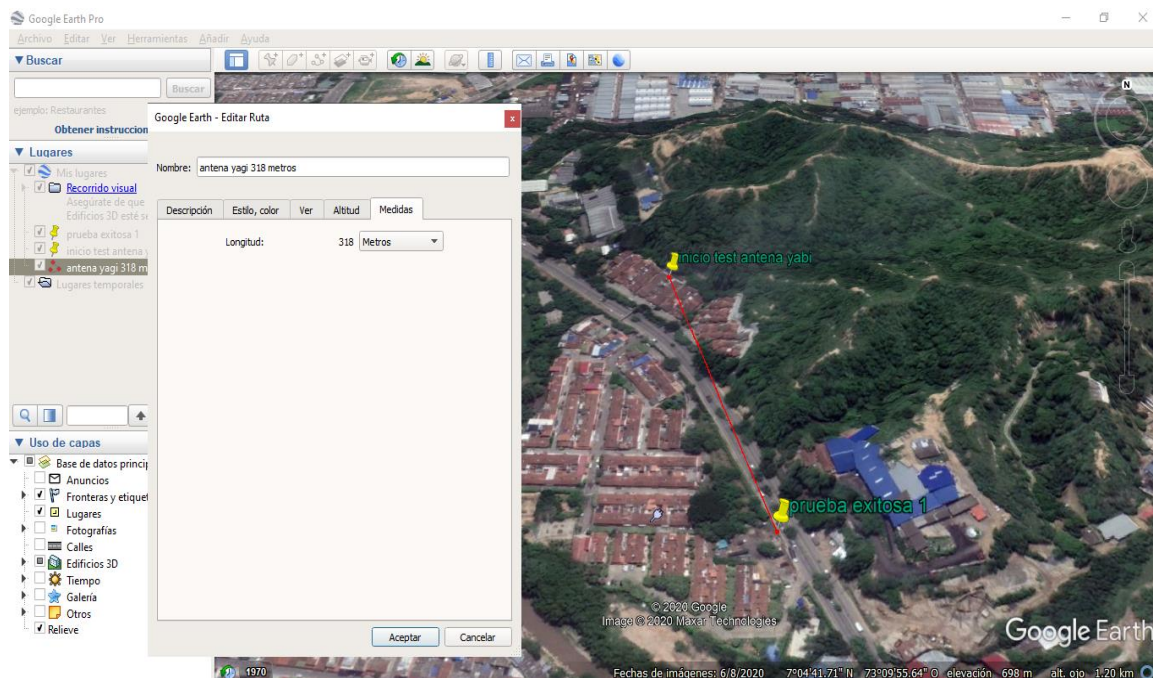


Figura 72 Prueba de alcance realizada, llegando a una distancia de 318 m. Fuente: Imagen generada utilizando la herramienta Google Earth.

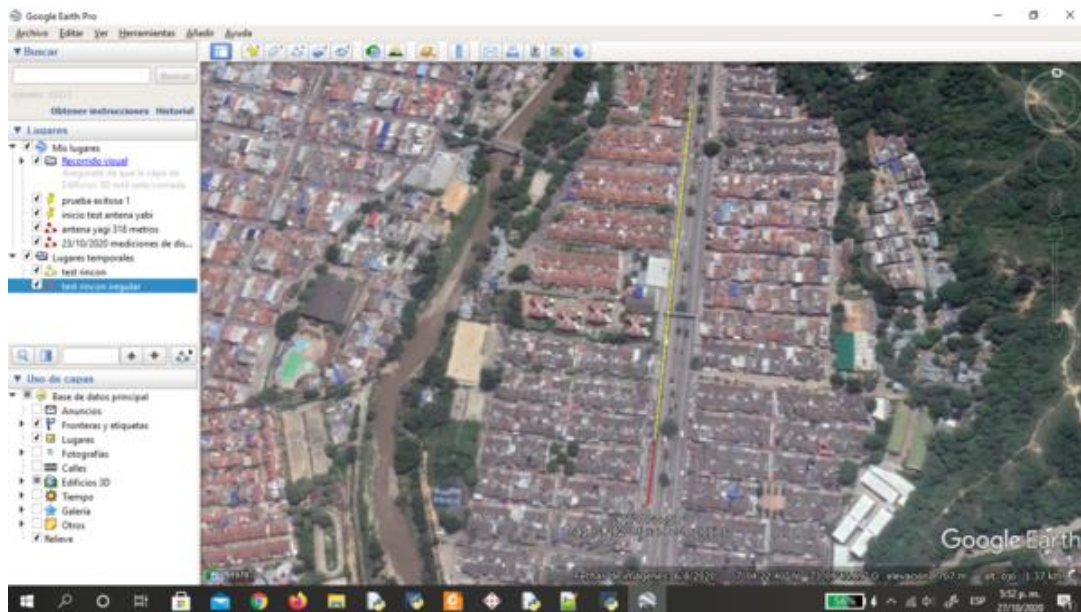


Figura 73 En una segunda prueba se tenía una expectativa de alcance de 776 m (línea amarilla), y el dispositivo alcanzó una distancia de 348m (línea roja). Fuente: Imagen generada utilizando la herramienta Google Earth.

```

Dashboard
30 Received Messages

{"identificadores":
["1A0B", "2A0B"], "lluvia": "0", "H_suelo": "0", "tipo
": "0", "RSSI": "-101"}
agroiot/datos

{"identificadores":
["1A0B", "2A0B"], "lluvia": "0", "H_suelo": "0", "tipo
": "0", "RSSI": "-100"}
agroiot/datos

{"identificadores":
["1A0B", "2A0B"], "lluvia": "0", "H_suelo": "0", "tipo
": "0", "RSSI": "-99"}
agroiot/datos

{"identificadores":
["1A0B", "2A0B"], "lluvia": "0", "H_suelo": "0", "tipo
": "0", "RSSI": "-99"}
agroiot/datos

{"identificadores":
["1A0B", "2A0B"], "lluvia": "0", "H_suelo": "0", "tipo
": "0", "RSSI": "-99"}
agroiot/datos

{"identificadores":
["1A0B", "2A0B"], "lluvia": "0", "H_suelo": "0", "tipo
": "0", "RSSI": "-98"}
agroiot/datos

{"identificadores":
["1A0B", "2A0B"], "lluvia": "0", "H_suelo": "0", "tipo
": "0", "RSSI": "-98"}
agroiot/datos

```

Figura 74 Tramas recibidas vía MQTT enviadas por el dispositivo Base luego de ser enviadas a este por el dispositivo Nodo vía LoRa.. Nota: La visualización de los registros se realiza gracias al software MyMQTT



```

Dispositivo_Nodo
1 #include <SPI.h>
2 #include <LoRa.h>
3
4 #include "DHTesp.h"
5 #include "DHT.h"
6 #include "Ticker.h"
7
8 #ifndef ESP32
9 #pragma message (THIS EXAMPLE IS FOR ESP32 ONLY!)
10 #error Select ESP32 board.
11 #endif
12
13 #define serhex      "1A0B"
14 #define SS          15      //33
15 #define RST         33      //12
16 #define DIO         34      //14
17 #define DHTPIN      4       //humedad y temperatura
18 #define DHTTYPE DHT11      // DHT 11
19 const int sensorPin = 13;  //lluvia
20 int H_Suelos = 39;        // suelos
21 DHT dht(DHTPIN, DHTTYPE); // Iniciar variable DHT
22 int data=0;              // Registros LoRa
23 int value = 0;          // Lectura del pin digital del sensor de lluvia
24 int humidity;          // Variable de lectura de humedad
25 byte tipo = 3;         // Tipo de sensor procesado
26 byte S_H_D = 2;        // Variable de control
27 byte sumar = 8;        // Guía de los paquetes
28 char* c_hum="";        // Valor de Humedad
29 char* c_tem="";        // Valor de temperatura
30

```

Figura 75 Código fuente correspondiente a dispositivo Nodo - parte 1. Fuente: Elaboración propia.

```

31 void setup()
32 {
33   pinMode(sensorPin, INPUT); // Definir pin como entrada
34   SPI.begin(12, 14, 27, SS); // Iniciar comunicación SPI
35   LoRa.setPins(SS, RST, DIO); // Configurar pines de comunicador LoRa
36   Serial.begin(115200);      // Iniciar Comunicación Serial
37   dht.begin();              // Iniciar sensor DHT
38   while (!Serial);
39   Serial.println("LoRa Receiver");
40
41   if (!LoRa.begin(433E6))    // Iniciar LoRa
42   {
43     Serial.println("Starting LoRa failed!");
44     while (1);
45   }
46   LoRa.setTxPower(20);       // modifica la potencia de la antena en dbm max 20
47   LoRa.setSpreadingFactor(9); // modifica el SF
48   LoRa.setSignalBandwidth(62500); // modifica la banda ancha
49   LoRa.setCodingRate4(8);    // modifica el CR (codificación de error)
50   LoRa.setSyncWord(0xF3);    // register the receive callback
51   Serial.println("LoRa Initializing OK!");
52 }

```

Figura 76 Código fuente correspondiente a dispositivo Nodo - parte 2. Fuente: Elaboración propia.

```

54 void loop()
55 {
56   delay(2000);
57   // Inicializar las variables
58   tipo = 3;
59   S_H_D = 2;
60   sumar=8;
61   //-----humedad y temperatura-----
62   float h = dht.readHumidity();
63   float t = dht.readTemperature();
64   String hum = String(h,0);
65   String tem = String(t,1);
66   //-----humedad lluvia-----
67   value = digitalRead(sensorPin );
68   if (value == LOW)
69   {Serial.println("Detectada lluvia"); S_H_D=1;}
70   else {S_H_D=0;}
71   //-----humedad suelos-----
72   humidity = analogRead(H_Suelos);
73   Serial.print("Lectura:");
74   Serial.println(humidity);
75   // Clasificar lectura de humedad
76   if (humidity >= 0 & humidity <= 300){Serial.println("Sensor en suelo seco");tipo=0;}
77   else if (humidity > 301 & humidity <= 700){Serial.println("Sensor en suelo húmedo");;tipo=1;}
78   else if (humidity >= 701){Serial.println("Sensor en agua");;tipo=2;}

```

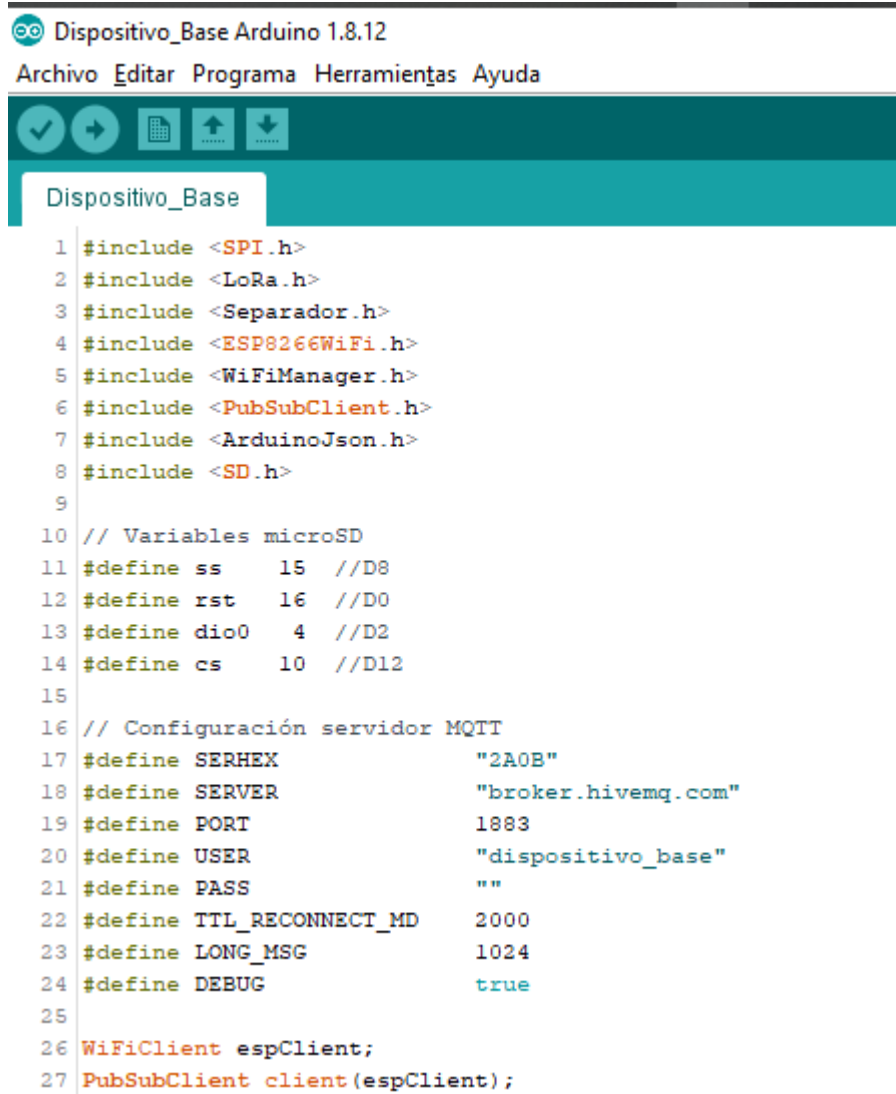
Figura 77 Código fuente correspondiente a dispositivo Nodo - parte 3. Fuente: Elaboración propia.

```

79 //----- agrupando lluvia y humedas suelos -----
80 if(S_H_D ==0)
81 {
82   if (tipo==0){sumar=0;}
83   else if(tipo==1){sumar=1;}
84   else if(tipo==2){sumar=2;}
85 }
86 else if(S_H_D ==1)
87 {
88   if (tipo==0){sumar=3;}
89   else if(tipo==1){sumar=4;}
90   else if(tipo==2){sumar=5;}
91 }
92 //-----envio de paquetes-----
93 LoRa.beginPacket();
94 LoRa.print(String(serhex) + "," + String(sumar) + "," + tem + "," + hum);
95
96 data = LoRa.readRegister(0x22);Serial.print("Registro 0x22: ");Serial.println(data);
97 LoRa.endPacket();
98 delay(42000);
99 }

```

Figura 78 Código fuente correspondiente a dispositivo Nodo - parte 4. Fuente: Elaboración propia.



The screenshot shows the Arduino IDE interface with the file 'Dispositivo_Base' open. The code is as follows:

```

1 #include <SPI.h>
2 #include <LoRa.h>
3 #include <Separador.h>
4 #include <ESP8266WiFi.h>
5 #include <WiFiManager.h>
6 #include <PubSubClient.h>
7 #include <ArduinoJson.h>
8 #include <SD.h>
9
10 // Variables microSD
11 #define ss    15 //D8
12 #define rst   16 //D0
13 #define dio0  4  //D2
14 #define cs    10 //D12
15
16 // Configuración servidor MQTT
17 #define SERHEX          "2A0B"
18 #define SERVER          "broker.hivemq.com"
19 #define PORT            1883
20 #define USER           "dispositivo_base"
21 #define PASS            ""
22 #define TTL_RECONNECT_MD 2000
23 #define LONG_MSG       1024
24 #define DEBUG          true
25
26 WiFiClient espClient;
27 PubSubClient client(espClient);

```

Figura 79 Código fuente correspondiente a dispositivo Base - parte 1. Fuente: Elaboración propia.

```

29 int cont1 =0; // Contador de intentos en la conexión MQTT
30 long lastMsg = 0; // Tiempo de actualización en ms
31 Separador Sepa; // Separador para los mensajes recibidos vía LoRa
32 String StemC;
33 String serhexT; // Datos obtenidos de la separación
34 String sumar; // Unir los datos
35 String clima_temperatura; // Capturar el valor de temperatura
36 String clima_humedad; // Capturar el valor de humedad
37 String RSSI_; // Intensidad de la señal LoRa
38 File myFile; // Variable del objeto de lectura de datos de la SD
39 File myFile2;
40 byte control_sd_datos = 2; // Variable para determinar si se guarda o no en la SD
41 byte guardar = 0;
42 int data = 0;

```

Figura 80 Código fuente correspondiente a dispositivo Base - parte 2. Fuente: Elaboración propia.


```

43 //----- wifi + mqtt + json -----//
44 void callback(char* topic, byte* payload, unsigned int length)
45 {
46     // Imprimir los mensajes recibidos de la comunicación MQTT y su Topic
47     Serial.print("Message arrived [");
48     Serial.print(topic);
49     Serial.print("] ");
50     String msg;
51     for (int i = 0; i < length; i++)
52     {
53         msg = msg + char(payload[i]);
54     }
55     Serial.print("length: ");Serial.println(length);
56     // Validar que los datos recibidos pertenecen al topic en que se trabaja "test/datos"
57     if (strcmp(topic,"test/datos")==0)
58     {
59         // Deserlizar los datos de la comunicación en formato JSON
60         const size_t capacity = LONG_MSG;
61         DynamicJsonDocument doc(capacity);
62         deserializeJson(doc, msg);
63     }
64     if (strcmp(topic,"test/wifi")==0)
65     {
66         // Deserializar los datos de la comunicación en formato JSON
67         const size_t capacity = LONG_MSG;
68         DynamicJsonDocument doc(capacity);
69         deserializeJson(doc, msg);
70     }
71 }

```

Figura 81 Código fuente correspondiente a dispositivo Base - parte 3. Fuente: Elaboración propia.

```

73 // Callback MQTT
74 void mqtt_json()
75 {
76     // Validar la conexión
77     if (!client.connected())
78     {reconnect();}
79     client.loop();
80     // Validar conexión a WIFI
81     if (WiFi.status() == 3)
82     {
83         // El dispositivo se encuentra conectado, enviar mensajes vía MQTT
84         control_sd_datos=0;msg1(true);cont1=0;control_sd_datos=2;
85     }
86     else{
87         // El dispositivo no se encuentra conectado, almacenar datos en la SD
88         control_sd_datos=1;msg1(true);control_sd_datos=2;
89     }
90 }

```

Figura 82 Código fuente correspondiente a dispositivo Base - parte 4. Fuente: Elaboración propia.

```

92 // Callback para reconectar al servidor MQTT
93 void reconnect()
94 {
95     if(cont1==0)
96     {
97         Serial.print("Attempting MQTT connection...");
98         if (client.connect("ESP8266Client", USER, PASS))
99         {
100             Serial.println("connected");
101             client.subscribe("test/datos");
102             client.subscribe("test/wifi");
103         }
104         else
105         {
106             Serial.print("failed, rc=");
107             Serial.print(client.state());
108             Serial.println(" try again");
109         }
110         cont1++;
111     }
112 }

```

Figura 83 Código fuente correspondiente a dispositivo Base - parte 5. Fuente: Elaboración propia.

```

114 void setup() {
115     Serial.begin(115200);
116     //-----wifi-----
117     // Configurar el dispositivo en modo estación
118     WiFi.mode(WIFI_STA);
119     WiFiManager wifiManager;
120     char combinedArray[20];
121     char array1[10] = "[Base] ";
122     // Fijar el nombre de la red
123     sprintf(combinedArray, "%s %s", array1, SERHEX); // With word space
124     wifiManager.autoConnect(combinedArray);
125     // Iniciar la conexión al servidor MQTT
126     client.setServer(SERVER, PORT);
127     client.setCallback(callback);
128     //-----Lora-----
129     while (!Serial);
130     Serial.println("LoRa Receiver Callback");
131     LoRa.setPins(ss, rst, dio0);
132     // Iniciar la comunicación LoRa indicando el ID del módulo LoRa de la Base
133     if (!LoRa.begin(433E6))
134     {
135         Serial.println("Starting LoRa failed!");
136         while (1);
137     }
138     LoRa.setSpreadingFactor(9);
139     LoRa.setSignalBandwidth(62500);
140     LoRa.setCodingRate4(8);
141     LoRa.setSyncWord(0xF3);
142     LoRa.receive();
143     //-----sd-----
144     Serial.println("Initializing SD card...");
145     // Inicializar la memoria SD

```

Figura 84 Código fuente correspondiente a dispositivo Base - parte 6. Fuente: Elaboración propia.

```

146  if (!SD.begin(cs))
147  {
148      Serial.println("initialization failed!");
149      return;
150  }
151  else
152  { Serial.println("initialization done.");}
153  // Abrir el archivo de respaldo en la SD
154  myFile = SD.open("datos1.txt", FILE_WRITE); //born to file
155  myFile.close();
156
157  if (myFile)
158  {
159      Serial.print("Writing to datos1.txt...");
160      myFile.close();
161      Serial.println("done.");
162  }
163  else
164  {
165      Serial.println("error opening datos1.txt");
166  }
167
168  myFile = SD.open("datos1.txt");
169  if (myFile)
170  {
171      Serial.println("datos1.txt:");
172
173      while (myFile.available()) {
174          Serial.write(myFile.read());
175      }

```

Figura 85 Código fuente correspondiente a dispositivo Base - parte 7. Fuente: Elaboración propia.

```

177  myFile.close();// close the file:
178  }
179  else
180  {
181      Serial.println("error opening datos_json.txt");
182  }
183  }
184
185  void loop()
186  {
187      // Esperar por el paquete LoRa
188      int packetSize = LoRa.parsePacket();
189      if (packetSize)
190      {
191          int longitud = sizeof(packetSize);
192          Serial.print("longitud: ");Serial.println(longitud);
193          guardar=0;
194          Serial.println("Paquet recu ");
195          String LoRaData = "";
196          while (LoRa.available())
197          {
198              // Imprimir mensaje recibido por LoRa
199              LoRaData = LoRaData+(char)LoRa.read();
200          }
201          // Imprimir los datos recibidos vía LoRa
202          String datosrecibidos = LoRaData;
203          serhexT = Sepa.separa(datosrecibidos, ',', 0);
204          sumar = Sepa.separa(datosrecibidos, ',', 1);
205          clima_temperatura = Sepa.separa(datosrecibidos, ',', 2);
206          clima_humedad = Sepa.separa(datosrecibidos, ',', 3);
207          RSSI = String(LoRa.packetRssi());

```

Figura 86 Código fuente correspondiente a dispositivo Base - parte 8. Fuente: Elaboración propia.

```

208     Serial.println("dato0: " + serhexT);
209     Serial.println("dato1: " + sumar);
210     Serial.println("dato2: " + clima_temperatura);
211     Serial.println("dato3: " + clima_humedad);
212     // Función encargada de hacer las validaciones y envío a MQTT
213     mqtt_json();
214 }
215 long now = millis();
216 long corte = now - lastMsg;
217 // Frecuencia de envío de los mensajes
218 if (corte > 20000)
219 {
220     lastMsg = now;
221     msg2(true);
222 }
223 }
224
225 void msg1(bool enviar)
226 {
227     // Validar la posibilidad de enviar los datos
228     if(enviar==true && control_sd_datos==0)
229     {
230         const size_t capacity = LONG_MSG; //envia ler mensaje
231         DynamicJsonDocument doc(capacity);
232         JSONArray identificadores = doc.createNestedArray("identificadores");
233         // Identificador, Nodo - Base
234         identificadores.add("1A0B");
235         identificadores.add("2A0B");
236         doc["sumar"] = sumar;
237         doc["clima_temperatura"] = clima_temperatura;
238         doc["clima_humedad"] = clima_humedad;

```

Figura 87 Código fuente correspondiente a dispositivo Base - parte 9. Fuente: Elaboración propia.

```

239     doc["RSSI"] = RSSI_;
240     char buffer[LONG_MSG];
241     serializeJson(doc, buffer);
242     client.publish("agroiot/datos", buffer); //guarda en la sd
243 }
244 // Si los datos no pueden ser enviados almacenar en la memoria SD
245 else if(enviar==true && control_sd_datos==1)
246 {
247     const size_t capacity = LONG_MSG;
248     DynamicJsonDocument doc(capacity);
249     JSONArray identificadores = doc.createNestedArray("identificadores");
250     identificadores.add("1A0B");
251     identificadores.add("2A0B");
252     doc["sumar"] = sumar;
253     doc["clima_temperatura"] = clima_temperatura;
254     doc["clima_humedad"] = clima_humedad;
255     doc["RSSI"] = RSSI_;
256     char buffer[LONG_MSG];
257     serializeJson(doc, buffer);
258     datos_sd(buffer);
259 }
260 }

```

Figura 88 Código fuente correspondiente a dispositivo Base - parte 10. Fuente: Elaboración propia.

```

262 // Realizar el control de otras variables a través de MQTT
263 void msg2(bool enviar)
264 {
265     if(enviar==true)
266     {
267         const size_t capacity = JSON_OBJECT_SIZE(1);
268         DynamicJsonDocument doc(capacity);
269         doc["control"] = "on";
270         char buffer[LONG_MSG];
271         serializeJson(doc, buffer);
272         client.publish("test/wifi", buffer);
273     }
274 }
275 // Lectura de los datos de la SD
276 void datos_sd(char mensaje [1024])
277 {
278     myFile = SD.open("datos1.txt", FILE_WRITE);
279     if (myFile) // if the file opened okay, write to it:
280     {
281         Serial.println("escribiendo: ");
282         myFile.println(mensaje);
283         myFile.close(); // close the file:
284         Serial.println("hecho");
285     }
286     else
287     {
288         Serial.println("error opening datos1.txt");
289     }
290 }

```

Figura 89 Código fuente correspondiente a dispositivo Base - parte 11. Fuente: Elaboración propia.

7.7 Consolidación del informe final

Durante el proceso de desarrollo del proyecto se ha podido identificar una serie de oportunidades presentes en el sector del agro y específicamente en el cultivo del aguacate, donde gracias a la demanda de exportación que viene creciendo hace crecer la demanda de soluciones que contribuyan en el seguimiento y mejora continua del producto con miras a llegar a los estándares exigidos para la exportación y teniendo la posibilidad de disputar los principales lugares junto con México como principales productores de Aguacate a nivel mundial.

De acuerdo con metodología planteada en la búsqueda de sensores se encuentra variedad limitada pero estable confiable para trabajar en la búsqueda del primer alcanzable

definido en el propósito de este proyecto, dejando entre ver que en un mayor margen de precisión y en una escala del tipo industrial la necesidad de sensores de mayor robustez es inminente, pero el propósito es procurar llegar a brindar una alternativa de calidad a un precio asequible para un mayor segmento del mercado.

Dentro de los valores diferenciales del presente proyecto se puede denotar principalmente el uso de tecnología IoT y el uso de tecnología LoRa para llevar la comunicación a distancias importantes y tratar de optimizar los recursos necesarios para poder contar con una solución de este tipo, esto entendiendo que la conectividad en la zona rural es bastante compleja y limitada, pero gracias a este protocolo se pueden desarrollar soluciones muy compactas de bajo costo y de gran funcionalidad y potencial.

Por otra parte, el uso de tecnología IoT abre la puerta a la llegada de otras tecnologías emergentes como Big Data, Inteligencia Artificial, Machine Learning, Deep Learning entre otras que pueden ser una manera de acelerar el desarrollo del sector rural en el mundo.

Como parte de este informe se presentan a continuación los costos relacionados con los dispositivos de los cuales fue objeto de desarrollo el presente proyecto:

Tabla 3

Presupuesto general

CANTIDAD	DESCRIPCIÓN	VALOR	TOTAL
1	Sensor DHT11	8.000	8.000
1	Sensor YI100	5.000	5.000
1	Sensor YC69	8.100	8.100
1	Módulo microSD	5.000	5.000
1	Panel Solar	37.300	37.300

2	LoRa 02	20.230	40.460
1	ESP32	19.040	19.040
1	ESP8266	13.090	13.090
7	Resistencias 0805	60	420
4	Suiches	200	800
2	Botones	300	300
1	Batería 18650	11.900	11.900
1	Porta pila	881	881
1	Módulo de carga USB	3.808	3.808
2	Conectores de 6 pines	19.9	119.4
2	Cajas de distribución eléctrica	19.700	39.400
		TOTAL	194.693

Fuente: Elaboración propia.

8. Conclusiones

- El uso de aplicaciones IoT han venido tomando fuerza durante los últimos años y esto se verá reflejado en la vida cotidiana y en los procesos tanto industriales como agrícolas en un corto plazo.
- El sector de la agricultura y en cultivos con amplia proyección de exportación como el aguacate requieren herramientas que estén al alcance de la mayoría de los productores para de esa manera alcanzar los estándares de calidad.
- La implementación de tecnología LoRa contribuirá en la potencialización del sector agrícola y permitirá la expansión de soluciones IoT que hoy en día han venido siendo limitadas por la infraestructura de conectividad.

9. Recomendaciones

- Disponer estratégicamente los dispositivos nodo y que estén en un rango de distancia en el cual los dispositivos base puedan captar la información de manera adecuada.
- Realizar un monitoreo y registro de los datos de manera continúa teniendo en cuenta la frecuencia de envío de información de estos dispositivos, el cual se estima en 10 minutos.
- Hacer una revisión trimestral de los sensores de humedad en tierra para detectar posible deterioro de este por uso u oxidación.
- Definir un plan de acción ante las diversas mediciones que puede ofrecer estos dispositivos.

10. Referencias Bibliográficas

ANACAFE (2019). *Programa de diversificación de ingresos en la empresa cafetera.*

RedMujeres. <http://redmujeres.org/>

Bancolombia (2018). *Guía completa como cultivar aguacate.* Grupo Bancolombia

<https://www.grupobancolombia.com/>

Cámara de comercio de Bucaramanga. (2018). *Actividad agrícola provincias de Santander.*

Cámara de comercio de Bucaramanga. <https://www.camaradirecta.com>

Canna (2020). *Plagas y enfermedades.* Canna. <https://www.canna.es>

Citri-Caldas. (2017). *Barrenador de las ramas del aguacate.* Citri Caldas.

<https://www.citricaldas.com.co>

Colegio Americano del sur del valle. (2015). *Humedad ambiental.* Google Sites.

<https://www.sites.google.com>

Comité estatal de sanidad vegetal del estado de México. (2008). *Campaña manejo*

fitosanitario. Gobierno de México. <https://gob.mx>

Comité estatal de sanidad vegetal del estado de México. (2008). *Campaña manejo*

fitosanitario. Gobierno de México. <https://gob.mx>

Definición (2020). *Definición de JSON.* Definición. <https://definicion.de/>

Destino negocio. (2020). *IoT en el agro nuevos sistemas para aumentar la productividad.*

Destino negocio. <https://destinonegocio.com/>

Ecu Red. (2020). *Sensor electrónico de lluvia.* Ecu Red. <https://www.ecured.cu/>

El Espectador. (2019). *Se concreta primera exportación de aguacate colombiano a China*.

El Espectador. <https://www.elespectador.com/>

Fluencecorp. (2017). *Qué son las precipitaciones pluviales*. Fluence News Team.

<https://www.fluencecorp.com/>

Hoy. (2020). *El cultivo del aguacate*. Canales Hoy. <https://canales.hoy.es/>

Hurwitz, J., & Kirsch, D. (2020). *Cloud Computing*. Google Books.

<https://books.google.com/>

Ideam. (2020). *Radiación solar ultravioleta*. Ideam. <https://ideam.gov.co/>

Info Agro. (2020). *El cultivo del aguacate*. Info Agro. <https://infoagro.com/>

Inia. (2020). *Temperatura del aire atmosférico*. Inia. <https://inia.org.uy/>

Intagri. (2020). *Control de la pudrición de raíz en aguacate*. Intagri. <https://intagri.com/>

Meteorología en red. (2020). *¿Qué es la meteorología?* Meteorología en red.

<https://www.meteorologiaenred.com/>

Mozilla. (2020). *HTTP*. Mozilla. <https://developer.mozilla.org/>

MQTT. (2020). *MQTT*. MQTT. <https://mqtt.org/>

Navarra. (2020). *Metereología y climatología de Navarra*. Navarra. <https://navarra.es/>

Parra Ramírez, d., & serrano scarpetta, s. (2017). *Guía de exportación de aguacate Hass para pequeños y medianos productores en Colombia*. Icesi. <https://icesi.edu.co/>

PCE-Ibérica. (2020). *Sensórica*. PCE Ibérica. <https://pce-iberica.es/>

Pérez. F. A. (2017). *Internet de las cosas*. Uigv. <http://uigv.edu.pe/>

Siberzone. (2020). *¿Qué es un sensor de temperatura y para qué se utiliza?* Siberzone.

<http://siberzone.es/>

Universidad de Sevilla. (2004). *Introducción a las aplicaciones web*. Isi. <http://isi.us.es/>

Valverde villarán. (2005). *Capítulo 2. Microcontroladores: Características y aplicaciones*

generales. Bibing. <http://bibing.us.es/>

Villa Sánchez. (2018). *Modelo de evaluación de plataformas tecnológicas para el*

almacenamiento y procesamiento de grandes datos caso en salud asistencial. Eafit.

<http://eafit.edu.co/>