



**UNIVERSIDAD
DE ANTIOQUIA**

**ADQUISICIÓN DE DATOS PARA EL MONITOREO
REMOTO DE VARIABLES DE UN CULTIVO VERTICAL
A TRAVÉS DE UNA PLATAFORMA IOT**

Autor

María Katherine Plazas Olaya

Universidad de Antioquia

**Facultad de Ingeniería, Departamento de Ingeniería
Electrónica y Telecomunicaciones**

Medellín, Colombia

2020



ADQUISICIÓN DE DATOS PARA EL MONITOREO REMOTO DE VARIABLES DE
UN CULTIVO VERTICAL A TRAVÉS DE UNA PLATAFORMA IOT

María Katherine Plazas Olaya

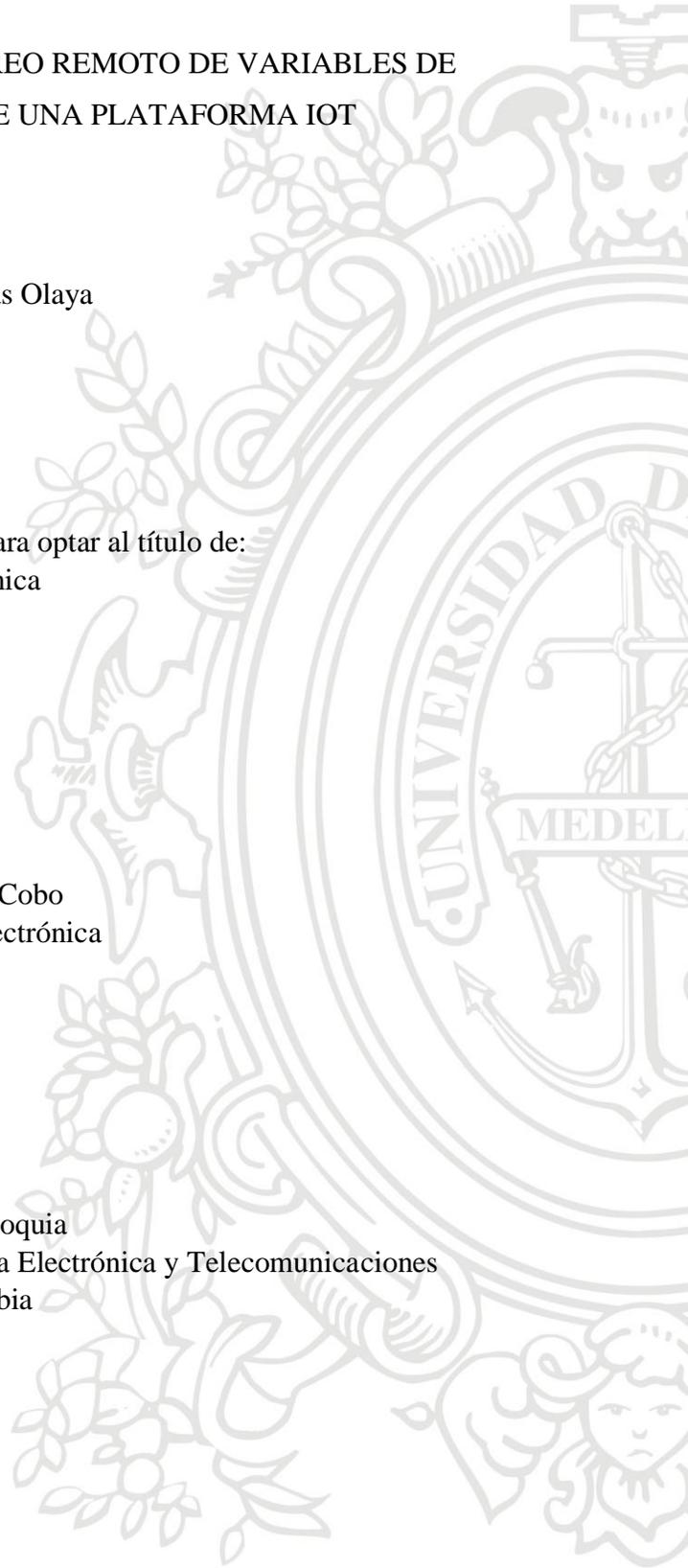
Trabajo de grado como requisito para optar al título de:
Ingeniera Electrónica

Asesor:

José Edinson Aedo Cobo
Ph. D en Ingeniería Electrónica

Universidad de Antioquia
Facultad de Ingeniería, Departamento de Ingeniería Electrónica y Telecomunicaciones
Medellín, Colombia

2020



Contenido

1. Resumen.....	4
2. Introducción	4
3. Objetivos	6
3.1. Objetivo general	6
3.2. Objetivos específicos	6
4. Marco Teórico	6
4.1. Cultivos verticales.....	6
4.2. Plataforma IoT Sentilo para el monitoreo de cultivos verticales	9
4.3. Red inalámbrica de sensores	11
4.4. Protocolo de comunicación LoRa.....	13
4.5. Protocolo MQTT.....	14
4.6. Variables para el monitoreo de cultivos verticales.....	14
5. Metodología.....	15
5.1. Diseño del hardware para los nodos sensores.....	15
5.2. Software para los nodos sensores.....	25
5.3. Implementación de la red de sensores basado en protocolo de comunicación LoRa.....	41
5.4. Implementación de la plataforma IoT basada Sentilo.	43
6. Resultados y análisis	47
6.1. Nodo sensor.....	47
6.2. Plataforma IoT basada en Sentilo.....	48
6.3. Prueba de funcionamiento del sistema.....	49
6.4. Prototipo del cultivo vertical.....	50
7. Conclusiones	51
8. Referencias Bibliográficas	51

1. Resumen

La población mundial viene creciendo a un ritmo acelerado, lo cual implica un crecimiento continuo en la demanda de alimentos, especialmente en las grandes ciudades donde la tasa de densificación es superior. Este fenómeno torna a la seguridad alimentaria como un factor muy importante en el desarrollo de las ciudades. De esta manera, las ciudades deben preocuparse por tener acceso a alimentos nutricionalmente adecuados y mantener una capacidad de producir alimentos, ya sea nivel urbano o periurbano. En este contexto, se viene investigando y desarrollando tecnologías alternativas que permitan el cultivo de alimentos dentro de las ciudades y en territorios cercanos, con una mayor productividad. Dentro de estas tecnologías se pueden destacar la hidroponía y los cultivos verticales, los cuales podrían ser una alternativa en la solución a dicho problema.

Este trabajo se concentra en el desarrollo de un sistema IOT basado en el protocolo de comunicación LoRa, para la gestión de cultivos verticales hidropónicos. El sistema permite el monitoreo remoto de parámetros como PH, conductividad eléctrica (CE), temperatura, humedad relativa, intensidad lumínica e índice UV, a través de la plataforma Sentilo. Como resultado se tiene un registro de los parámetros del cultivo, tomados en un intervalo de tiempo, permitiendo al cultivador tomar decisiones acertadas en cuanto al suministro de nutrientes y control de riego, entre otros. En el proyecto, también se construye un prototipo de cultivo hidropónico a escala, con el propósito de realizar pruebas del sistema desarrollado.

2. Introducción

El crecimiento desproporcionado de las ciudades, la contaminación, la desertización y el cambio climático han provocado una creciente disminución de las zonas agrícolas, en donde se torna muy importante el concepto de seguridad alimentaria, por lo cual se viene investigando y desarrollando tecnologías alternativas que permitan el cultivo de alimentos dentro de las ciudades y territorios cercanos. Dentro de estas tecnologías se pueden destacar la hidroponía y los cultivos verticales, dando una alternativa de solución a dicho problema [1].

Por otro lado, las ciudades inteligentes son conocidas no solo desde un punto de vista tecnológico, sino también por su sostenibilidad económica, social y ambiental [2]. Los cultivos verticales pueden instalarse en casi todas partes y tienen el potencial de urbanizar la producción de alimentos, permitiendo el suministro de verduras desde áreas próximas o muy próximas, aunque no convencionales (patios,

terrazas, paredes, etc) a los grandes centros de consumo (hogares, mercados) [1]. Por lo tanto, esto podría traer importantes beneficios socioambientales, conduciendo a un sistema de vida diferente, renaturalizando las ciudades inteligentes y además se pueden resolver varios problemas que actualmente afectan a estas ciudades inteligentes, en el marco de que deben ser sostenibles.

Los cultivos verticales pueden construirse en edificios de varios pisos o rascacielos que utilizan tecnologías de control climático y sistemas agrícolas avanzados para cultivar, permitiendo la ampliación de la agricultura urbana, mejorar el rendimiento de la producción y mantener la disponibilidad de verduras frescas de alta calidad durante todo el año [3]. La técnica de hidroponía en general es utilizada en estos edificios, el cual permite el cultivo de plantas sin la necesidad de tierra. Con esta técnica se pueden producir plantas principalmente de tipo herbáceo como lechuga, tomate, fresas entre otras. En este sistema las plantas reciben los elementos minerales esenciales por medio de una solución nutritiva que circula a través de las raíces.

Por otra parte, la internet de las cosas también conocido como IoT (Internet of Things), es un nuevo tipo de sistema informático en el que se utilizan pequeños dispositivos electrónicos equipados con sensores para detectar el entorno operativo de un sistema y facilitar su control [4]. Esta tecnología se viene usando en diversas aplicaciones. Recientemente, la agricultura inteligente es una aplicación de este tipo de tecnología y puede entenderse como la gestión de cultivos de manera innovadora, en donde se utilizan sistemas de monitoreo por medio de sensores y dispositivos que permiten obtener una información más detallada del cultivo, permitiendo una gestión más eficaz.

Este trabajo se desarrolla dentro de este contexto. Para lo cual, se diseña un sistema IoT, que facilita la gestión cultivos verticales, considerando los requerimientos para el control del cultivo, el manejo de la productividad y su gestión inteligente. Específicamente, se desarrolla dos nodos sensores inteligentes para el monitoreo de los cultivos y una plataforma para la interconexión inalámbrica de los nodos usando el protocolo de comunicación LoRa, permitiendo la disposición de información del cultivo mediante una aplicación WEB.

3. Objetivos

3.1. Objetivo general

Desarrollar un nodo sensor inteligente que permita el monitoreo de variables de un cultivo vertical, usando una plataforma IoT, para la gestión de cultivos en zonas urbanas.

3.2. Objetivos específicos

- Implementar un sistema de acondicionamiento de señales para los sensores de PH, conductividad eléctrica del agua, temperatura y humedad del ambiente, intensidad lumínica, temperatura del agua y radiación UV, con el propósito de obtener datos confiables de las variables medidas.
- Desarrollar un nodo sensor inteligente, a nivel de hardware, seleccionando componentes de bajo consumo energético y bajo costo.
- Desarrollar los componentes del software para el nodo sensor inteligente, para las funciones de muestreo de las variables procesamiento de información y transmisión de datos hacia una plataforma IoT ya existente.
- Construir un prototipo de un cultivo vertical para realizar las pruebas del sistema.

4. Marco Teórico

4.1. Cultivos verticales

La práctica de cultivar en capas apiladas verticalmente, utilizando tecnologías de control climático y sistemas agrícolas avanzados, se conoce como cultivos verticales [3]. La hidroponía es una de las técnicas utilizadas para el cultivo de plantas en este tipo de práctica.

Los cultivos hidropónicos se caracterizan por el cultivo de plantas en soluciones nutritivas, el suelo no es necesario para el crecimiento de las plantas, pero debe tenerse en cuenta que los nutrientes y minerales que contiene el suelo son definitivamente necesarios para su normal desarrollo [3]. El suelo es el lugar en donde tradicionalmente viven las raíces de las plantas y es básicamente el

substrato de los nutrientes necesarios para el desarrollo de estas. Al no utilizar el suelo, se puede tener un control preciso en cuanto a los requerimientos nutricionales de las plantas, y de esta manera administrar las proporciones correctas para que las plantas puedan desarrollarse sin estrés, madurar más rápido y en la cosecha tener una mejor calidad. Como también eliminar enfermedades y malezas que son generalmente transmitidas por el suelo hacia la planta [1].

La hidroponía es un método ampliamente aceptado para producir ciertos cultivos especializados, como tomates, lechugas, pepinos, pimientos, hierbas, plantas de follaje y flores. La agricultura en ambientes controlados y la hidroponía, parecen ser la respuesta a muchas de las dificultades asociadas con la producción de cultivos en suelo, debido a la degradación continua de este, la pérdida de fertilidad, el uso indiscriminado de insumos químicos y, sobre todo, el continuo agotamiento de los recursos hídricos [5].

Existen diferentes técnicas utilizadas en los cultivos hidropónicos, algunos de estos son:

- Sistema de mecha o pabilo: es una de las técnicas más simples, ya que no requiere de bombas para transportar la solución nutritiva desde el depósito hasta las bandejas de crecimiento, este funciona por medio de la acción capilar, la cual absorbe agua y nutrientes mediante una mecha de tela especial. Cuenta con una maceta fabricada con diferentes materiales (vidrio, barro, plástico y aluminio) un recipiente donde se almacena el agua, nutrientes y una mecha absorbente [6]. Este sistema funciona con plantas individuales y no a gran escala. Además, se recomienda utilizar plantas que requieran poca agua.
- Sistema de flujo y refluo: esta técnica consiste en bombear agua con solución nutritiva al recipiente en donde se encuentran las plantas, luego es devuelta al recipiente de depósito. Esto se repite en determinados intervalos de tiempo para alimentar las raíces de las plantas [6].
- Sistema por goteo: funciona mediante dos cubetas sobrepuestas. En la cubeta inferior se almacena el agua con nutrientes, y en cubeta superior contiene algún tipo de sustrato. También posee un multi gotero circular. En el proceso una mini bomba es accionada a intervalos mediante un temporizador, con el fin de programar los riegos necesarios para cada planta y para cada etapa [6].

- Sistema de película nutritiva (NFT): este sistema tiene el beneficio de adaptarse a espacios reducidos, ya que éste puede construirse en forma piramidal; aprovechando los espacios verticales que tienen la capacidad de cultivar un gran número de plantas.

Funciona por medio de tramos de PVC, donde una ligera capa de agua es impulsada desde un recipiente hacia el interior de los tubos; mediante la acción de una bomba de agua en constante funcionamiento o por periodos de tiempo. Esta ligera película de agua circula a través de los tubos para llegar nuevamente al mismo recipiente, y así, estar recirculando constantemente. Estos tubos son perforados para crear las cavidades donde las plantas crecerán. Las raíces son colocadas dentro de estos espacios para que se alimenten del agua que corre por los tubos [6].

- Sistema de raíz flotante: Funciona por medio de camas de unicel (poliestireno expandido) o algún otro material ligero; los cuales contienen múltiples cavidades donde se insertan las raíces de las plantas. Estas camas se colocan sobre un recipiente lleno de agua con nutrientes, donde las raíces quedan suspendidas sobre la solución nutritiva; la cual debe de ser oxigenada continuamente [7].

- Sistema aeropónico: Funciona, en general, por medio de tramos de PVC, donde un rocío de agua y nutrientes es creado mediante micro aspersores instalados dentro de estos tubos. Una bomba de agua se acciona a intervalos para rociar la raíz de la planta que se encuentran instaladas dentro de las cavidades de los tubos. Una vez que la bomba se desactive, el agua con nutrientes regresa al recipiente donde es almacenada para la próxima microaspersión. La planta crece velozmente, debido a que se encuentra suspendida en el aire recibiendo mayor cantidad de oxígeno [6].

En la figura 1 se observan cada uno de los sistemas de cultivos hidropónicos mencionados anteriormente.

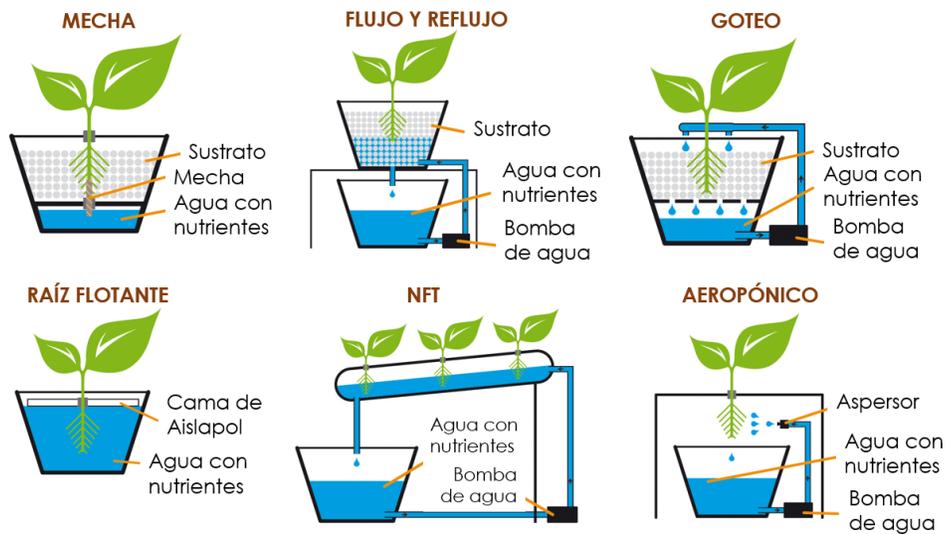


Figura 1. Sistemas para cultivos hidropónicos. Tomado de [6]

4.2. Plataforma IoT Sentilo para el monitoreo de cultivos verticales

Sentilo es una plataforma de manejo de datos, que proporciona una arquitectura simple para el uso y el aislamiento de las diferentes herramientas, aplicaciones y sensores, que proporcionan y requieren datos generados por la ciudad en la que estas se implementa, con el fin de administrar y suministrar estos datos adquiridos en una estructura de mucha mayor complejidad para la creación de ciudades inteligentes.

Desarrollada por el Instituto Municipal de Informática de Barcelona (IMI) en el 2012, con el fin de proporcionar y definir una estrategia en el desarrollo de ciudades inteligentes, Sentilo es una plataforma de actuadores y sensores de código abierto apoyada siempre en software libre, que tiene como fin el encajar y garantizar una fácil operabilidad en cualquier arquitectura de cualquier ciudad que busque dar un paso en el tema de ciudades inteligentes [8].

Esta plataforma tiene como elementos a destacar:

- Una interfaz para el procesamiento de mensajes, con una simple API REST.

- Una consola de administración para configurar el sistema y administrar el catálogo de aplicaciones.
- Una base de datos, orientada a lograr altas tasas de rendimiento.
- Una base de datos noSQL, con el fin de obtener un sistema más flexible y escalable.
- Un módulo básico de estadísticas que registra y muestra indicadores básicos de rendimiento de la plataforma.
- Una arquitectura de componentes extensible, garantizando la ampliación de la plataforma y su funcionalidad sin modificar el núcleo del sistema. Sentilo comienza con un conjunto inicial de agentes: uno para exportar datos a bases de datos relacionales y otro para procesar alarmas internas basadas en reglas básicas.

Sentilo utiliza agentes individuales para su comunicación, los agentes son módulos internos orientados a ampliar la funcionalidad de la plataforma, sin tener que alterar su núcleo y procesar casos específicos para mayor velocidad de reacción, escalabilidad e independencia. Sentilo ofrece varios agentes como:

- El servidor de Publicación y suscripción, el cual más que nada es el pilar fundamental y una API REST, la encargada de gestionar las comunicaciones relacionadas con el servidor.
- El agente de alarmas encargado de gestionar las alarmas dirigidas a la plataforma.
- El agente relacional que básicamente es el encargado de exportar a una base de datos relacional, los históricos.
- El agente actualizador de posición encargado de gestionar la ubicación global de los componentes y actualizar sus posibles cambios.
- El agente Histórico que se encarga de gestionar los históricos haciendo uso de OpenTSDB, una Base de Datos de Series temporales.

Además, Sentilo está compuesto por un catálogo web, el cual es una aplicación web que permite administrar, gobernar y monitorear los recursos y la actividad de la plataforma. El catálogo permite gestionar los siguientes recursos:

- Aplicaciones y módulos clientes: Puede ser vista como un catálogo, es la entidad que consume, administra, procesa, muestra los datos generados por la plataforma.
- Proveedores: Son aquellos que envían datos, es decir, aquellos que publican los datos (a diferencia de las aplicaciones que consumen los datos).

- Componentes: Corresponden a un elemento de hardware o software, con ubicación geoespacial (fija o móvil) que puede estar compuesto por uno o más sensores. Se usan los componentes en el catálogo para agrupar sensores que comparten un conjunto de propiedades (como ubicación, proveedor, potencia, conectividad, etc.). Se podría pensar en ellos como dispositivos físicos con un conjunto de sensores, como un microcontrolador con múltiples sensores conectados.
- Sensores: Dispositivos encargados de medir las variables de interés y generar los datos.

La seguridad en las comunicaciones de la plataforma se asegura mediante un token de autenticación. Este token debe enviarse en cada solicitud como un parámetro de encabezado de la solicitud HTTP denominada IDENTITY_KEY. Este token es único para cada aplicación de proveedor o cliente y es administrado por la aplicación de catálogo.

4.3. Red inalámbrica de sensores

Una red inalámbrica de sensores (Wireless Sensor Network, WSN), es una red de dispositivos distribuidos, los cuales utilizan sensores para monitorear condiciones físicas o ambientales. Cada dispositivo recibe el nombre de nodo y tiene como tarea principal recolectar datos de forma autónoma. Debido a que los nodos pueden ser ubicados en zonas remotas o de difícil acceso, cada nodo necesita consumir la menor cantidad de energía posible para asegurar la máxima autonomía [9]. Un nodo generalmente es un sistema basado en un microcontrolador que incorpora un módulo de radio frecuencia (RF), un sensor y un sistema de alimentación eléctrica [10], [11]. En la figura 2 se observa el diagrama de bloques de un nodo.

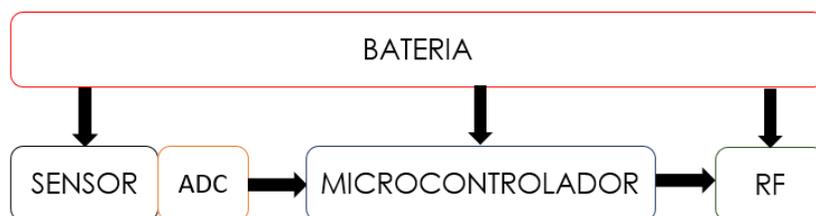


Figura 2. Componentes de un nodo.

La red inalámbrica de sensores en general incorpora un Gateway, como se observa en la figura 3. El Gateway permite conectar los nodos a alguna red de mayor jerarquía como internet. El protocolo inalámbrico usado en las redes de

sensores depende de los requerimientos de la aplicación. Algunos de los estándares disponibles incluyen radios de 2.4 GHz, basados en los estándares IEEE 802.15.4 o IEEE 802.11 (WiFi), redes de telefonía celular, Lora, entre otros [11].

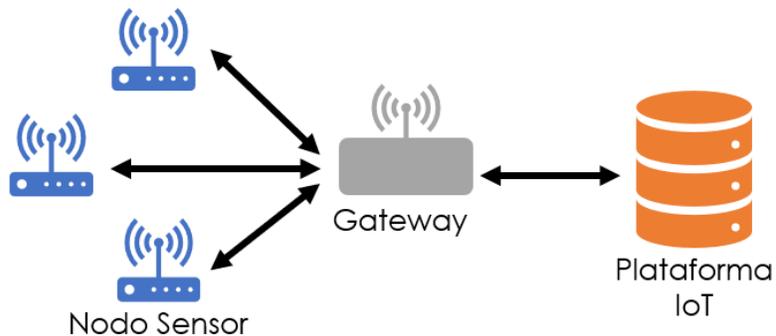


Figura 3. Componentes de una red inalámbrica de sensores.

Existen diferentes tipos de topologías para las redes inalámbricas de sensores, las cuales dependen de cómo se comunican los nodos con el Gateway, las topologías más comunes se muestran en la figura 4.

- Topología de estrella: todos los nodos se conectan directamente al Gateway.
- Topología de árbol: los nodos se conectan a un nodo de mayor jerarquía, transmitiendo a través de este, la información al Gateway.
- Topología tipo malla: es la topología que ofrece mayor confiabilidad, los nodos se pueden conectar a múltiples nodos del sistema y transmitir los datos por la ruta disponible de mayor confiabilidad [11], [12].

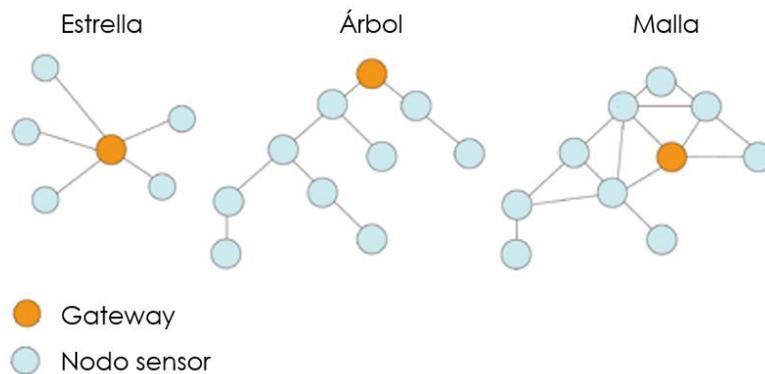


Figura 4. Topologías en una red inalámbrica de sensores. Tomado de [11]

Las redes inalámbricas de sensores tienen diferentes aplicaciones, por ejemplo en [13] se presenta un sistema de monitoreo de la calidad del aire de bajo costo y se basa en las redes inalámbricas. También pueden ser usadas en el monitoreo de volcanes [14], la agricultura de precisión [15] y la domótica [16].

4.4. Protocolo de comunicación LoRa

LoRa (Large Range) es un protocolo de comunicación a larga distancia a partir de radiofrecuencia, entre sus principales características se encuentran:

- Modulación de espectro expandido
- Bajo consumo
- Baja tasa con un máximo de 255 Bps
- Enlace punto a punto
- Comunicaciones a largas distancias hasta 20 kilómetros
- Opera en las bandas de frecuencia: 433[Mhz], 868[Mhz] o 915[MHz] y es libre dependiendo en la región en la que sea desplegado.

LoRa opera en la capa física del modelo OSI y para ejecutar una comunicación se requieren dos elementos: Gateway y nodo, como se muestra en la figura 5. El dispositivo sensor que recoge la información y la transmite es el nodo. El gateway es el mediador; recibe la información de los diferentes sensores y la retransmite, generalmente a través de Internet, a la nube para que ésta unifique la información de todos los sensores y actúe en consecuencia.

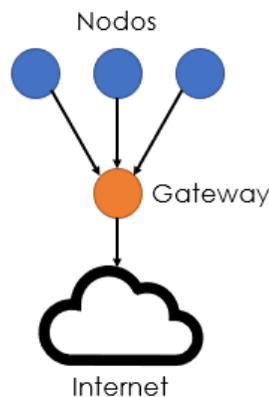


Figura 5. Elementos en una comunicación LoRa.

Para la utilización de la red con sistemas IoT existen dos formas: una es creando una propia red, en donde es necesario comprar los chips LoRa y hacer el diseño de los nodos, además configurar el Gateway, una de las ventajas que el usuario podrá hacer la red según sus necesidades; la otra forma es utilizar un operador de red, es decir, varias operadoras ofrecen las redes LoRaWAN en ciertas áreas, se pueden contratar planes de conectividad pero su principal desventaja es que el operador puede dejar de ofrecer el servicio [17].

4.5. Protocolo MQTT

MQTT (Message Queue Telemetry Transport) es un protocolo de transporte de mensajes Cliente/Servidor basado en publicaciones y suscripciones. El principio del modelo de comunicación de publicación suscripción consiste en que los componentes que están interesados en consumir cierta información registran su interés. Este proceso de registrar un interés se llama suscripción, por lo tanto, la parte interesada se llama suscriptor. Los componentes que desean transmitir cierta información lo hacen publicándola. Se les llama así publicadores. La entidad que garantiza que los datos lleguen de los publicadores a los suscriptores es el Broker. El Broker coordina las suscripciones y los suscriptores generalmente tienen que contactar al Broker explícitamente para suscribirse [18].

MQTT está diseñado para simplificar la implementación en el cliente concentrando toda la complejidad en el Broker. Publicadores y suscriptores están aislados, ellos no necesitan conocer la existencia del otro.

4.6. Variables para el monitoreo de cultivos verticales

Los cultivos verticales basados en sistemas hidropónicos, como se menciona previamente, no requieren de suelo para cultivar plantas. Estos se basan principalmente en los parámetros relacionados con el aire, el agua y los alrededores del sistema, por lo tanto, el nivel de pH, la conductividad eléctrica y los niveles de temperatura son importantes en los cultivos hidropónicos. Además, la intensidad de luz es un factor que influye en la tasa de eficiencia del crecimiento de la planta [19]. En la tabla 1 se presentan los parámetros esenciales para estos tipos de cultivos y su relevancia.

Tabla 1. Importancia de los parámetros en un cultivo hidropónico.

Parámetro	Importancia en el cultivo
PH	Es importante medir el pH en el agua en un cultivo hidropónico porque a determinados rangos de pH, las raíces pueden absorber nutrientes más fácilmente. [20]
CE	Al conocer la conductividad eléctrica del agua, se tiene una idea de la concentración general de los nutrientes requeridos para un correcto desarrollo de las plantas. Una baja conductividad indica que hay poca cantidad de nutrientes en el agua, lo que advierte al usuario la necesidad de suministrar nutrientes a la solución. [4]

Temperatura	La temperatura afecta los procesos biológicos, químicos y fisiológicos de las plantas como la germinación, división, expansión de las células, fotosíntesis, respiración, etc. Diferentes plantas requieren distintas temperaturas para poder crecer. [4]
Humedad relativa	La humedad relativa afecta el crecimiento de las plantas, especialmente la tasa de deshidratación resultante de la humedad relativa en el aire. Si la humedad relativa es baja, la planta tendrá una alta tasa de deshidratación. [4]
Índice UV e Intensidad Lumínica	La luz es un factor importante en el crecimiento y desarrollo de las plantas ya que el proceso de fotosíntesis por el cual convierte los nutrientes a energía requiere principalmente luz. [4]

En este trabajo, se diseñaron dos nodos sensores considerando los parámetros especificados en la tabla 1. Uno de los sensores se encarga de la medida de los parámetros ambientales (temperatura, humedad relativa, índice UV e intensidad Lumínica) y el otro está encargado de la medida de las condiciones en el agua (PH, temperatura y conductividad eléctrica).

5. Metodología

Para el desarrollo del trabajo, se llevaron a cabo un conjunto de actividades orientadas al diseño del hardware y del software del sistema. Inicialmente se describen las especificaciones y el desarrollo de los módulos de hardware y luego se detalla el diseño del software para los nodos sensores. También se realiza una descripción de la red de sensores y la implementación en la plataforma IoT basada en sentilo.

5.1. Diseño del hardware para los nodos sensores.

Como se mencionó anteriormente, el sistema de monitoreo para el cultivo vertical está formado por dos nodos sensores, uno encargado de las variables ambientales y el otro de las variables del agua. Los nodos constan de 4 bloques principales: microcontrolador, sensores, módulo Lora y módulos relés, el nodo sensor en el agua tiene adicionalmente un conversor análogo digital. El diagrama de bloques para nodo 1 (nodo ambiente) y nodo 2 (nodo agua) es mostrado respectivamente en la figura 6 y 7. El microcontrolador es el encargado de recibir las instrucciones desde el Gateway LoRa, procesarlas y enviar respuesta, si la instrucción lo requiere y controlar los demás bloques del circuito. Los sensores se encargan de medir magnitudes físicas y transformarlas

en señales eléctricas. El módulo LoRa permite transmitir datos de las variables sensadas. Como componentes adicionales se integraron dos relés en cada nodo sensor para dotar al sistema con una capacidad de realizar acciones de control mediante el encendido o pagado de dispositivos externos (por ejemplo, para el control de la bomba de riego). El conversor análogo digital convierte las señales de interés a una trama de bits, para que puedan ser procesadas por el microcontrolador.

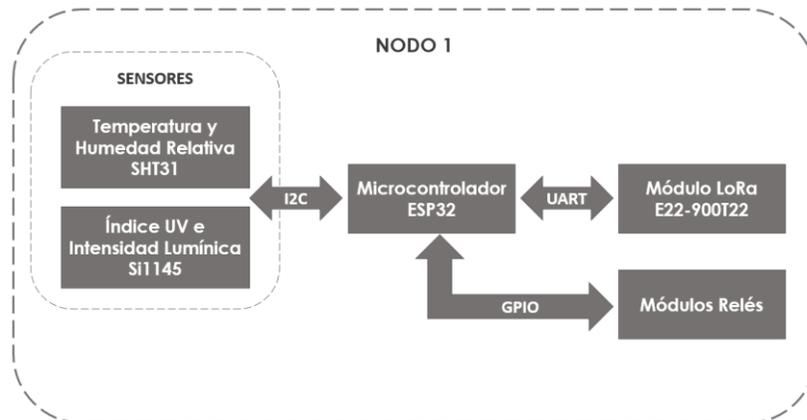


Figura 6. Diagrama de bloques para el nodo 1 (Nodo ambiente).

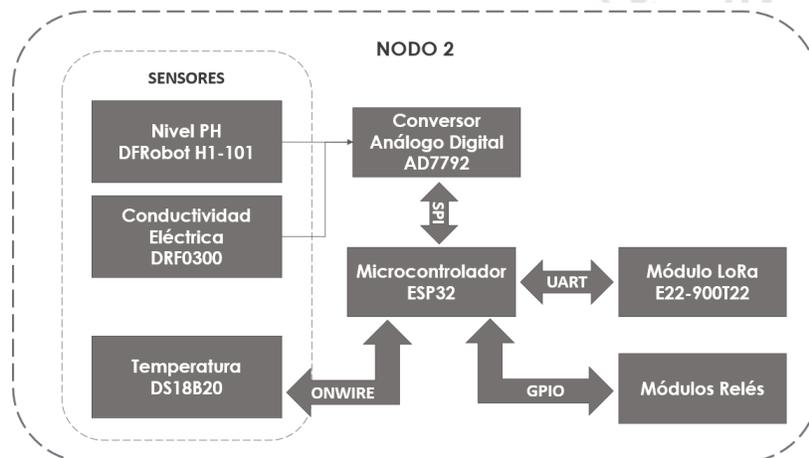


Figura 7. Diagrama de bloques para el nodo 2 (Nodo agua).

A continuación, se describen cada uno de los bloques de los nodos sensores.

Microcontrolador

El microcontrolador Wemos Mini D1 ESP32, fabricado por la empresa Espressif Systems [21], es el encargado del control de los dispositivos del nodo (sensores, ADC, módulo LoRa y módulos relés), la recolección y organización de los datos digitales y su posterior transmisión al Gateway LoRa. En la figura 8 se muestra el esquema de conexión del microcontrolador ESP32.

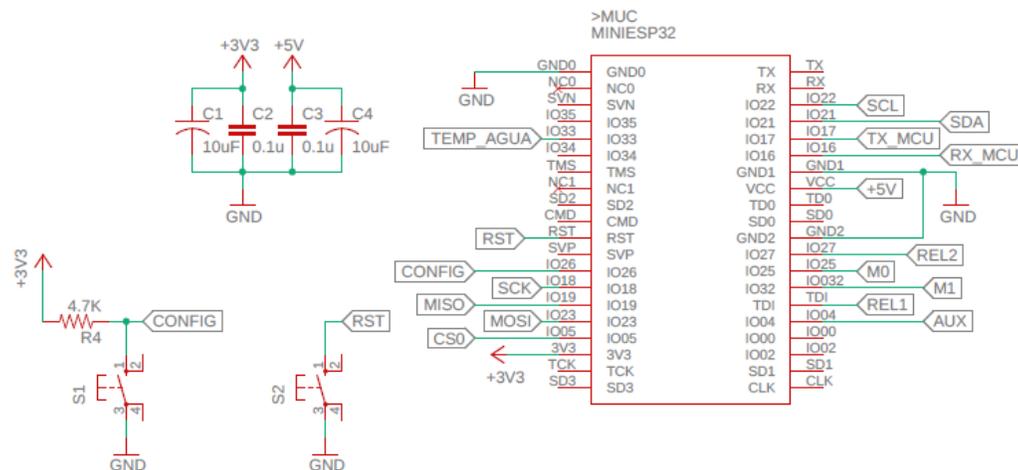


Figura 8. Esquema de conexión del Microcontrolador Wemos Mini D1 ESP32, realizado en los nodos.

El microcontrolador tiene una frecuencia de operación de 240MHz. Además, posee diversos periféricos necesarios para la comunicación con los demás bloques del circuito.

Uno de sus tres periféricos UART (Universal Asynchronous Receiver Transmitter) del microcontrolador, se empleó para realizar la comunicación serial con el módulo LoRa, a través del cual se realizó la transmisión de datos al Gateway LoRa.

Un periférico SPI (Serial Peripheral Interface), se empleó para la comunicación con el convertor análogo digital, ADC, es decir para la gestión de los sensores del nivel de PH y la conductividad eléctrica en el agua.

Un periférico I2C (Inter-Integrated Circuit), se empleó para la comunicación con los sensores de temperatura y humedad relativa en el ambiente.

Finalmente, algunos terminales de propósito general (GPIO) fueron utilizados para la comunicación con el sensor de temperatura en el agua, el control de

los módulos relés, botón de configuración, reset y configuración del módulo LoRa. La distribución de los terminales del microcontrolador se detalla en la tabla 2.

Tabla 2. Distribución de pines del microcontrolador.

Pin #	Nombre	Función
5	IO33	Comunicación ONWIRE para el sensor de temperatura en el agua.
11	RST	Control botón reset
13	IO26	Control botón configuración
14	IO18/SCK	Comunicación SPI para el ADC
15	IO19/MISO	
16	IO23/MOSI	
17	IO05/CS0	
25	IO04	Modo de operación para el módulo LoRa
27	IO32	
28	IO25	
26	TDI	Control relé 1
29	IO27	Control relé 2
35	IO16/RX	Comunicación UART para el módulo LoRa
36	IO17/TX	
37	IO21/SDA	Comunicación I2C para los sensores de temperatura y humedad relativa en el ambiente
38	IO22/SCL	

Sensor Temperatura y humedad relativa en el ambiente.

El dispositivo seleccionado para medir temperatura y humedad relativa es el sensor SHT31, fabricado por la empresa Sensirion [22], este tiene salida de señal digital de alta precisión, totalmente calibrada, linealizada y con compensación de temperatura. Sus características de operación se muestran en la tabla 3.

Tabla 3. Características de operación SHT31

Parámetro	Valor
Voltaje de alimentación	2.4 – 5.5 V DC
Interfaz	I2C, dirección: 0x44
Rango de operación temperatura	-40 – 125 °C
Rango de operación humedad	0 – 100 %HR
Precisión temperatura	±0.3 °C

Precisión humedad	±3 %HR
Repetibilidad baja	Temperatura: 0.24 °C Humedad: 0.25 %HR
Repetibilidad media	Temperatura: 0.12 °C Humedad: 0.15 %HR
Repetibilidad alta	Temperatura: 0.06 °C Humedad: 0.10 %HR
Resolución	Temperatura: 0.015 °C Humedad: 0.01 %HR

El protocolo de comunicación entre el MCU y SHT31 es I2C, el sensor admite frecuencias hasta 1000KHz. Todos los comandos y datos del SHT31 se asignan a un espacio de direcciones de 16 bits. Además, los datos y comandos están protegidos con una suma de verificación CRC. Esto aumenta la fiabilidad de la comunicación.

En el SHT31 los datos de medición siempre se transfieren como valores de 16 bits (entero sin signo). Estos valores ya están linealizados y compensados por los efectos de temperatura y voltaje de alimentación. La conversión de esos valores brutos en una escala física se puede lograr usando las siguientes ecuaciones.

Estimación para la conversión de humedad relativa (resultado en % HR):

$$HR = 100 * \frac{S_{HR}}{2^{16} - 1}$$

Estimación para la conversión de la temperatura (resultado en °C):

$$T = -45 + 175 * \frac{S_T}{2^{16} - 1}$$

S_{RH} y S_T denotan la salida bruta del sensor para humedad y temperatura, respectivamente.

Sensor índice UV e intensidad lumínica.

Para conocer las condiciones lumínicas del cultivo se utiliza el sensor Si1145, el cual es un detector de proximidad de reflectancia óptica, puede medir índice ultravioleta (UV) y es un sensor de luz ambiental cuyo estado operativo se controla a través de registros accesibles a través de la interfaz I2C. El circuito

integrado incluye un convertidor análogo digital, fotodiodos integrados de alta sensibilidad capaces de medir luz visible como la infrarroja, procesador de señal digital y uno, dos o tres controladores LED infrarrojos. El Si1145 ofrece un excelente rendimiento en un amplio rango dinámico y una variedad de fuentes de luz, incluida la luz solar directa [23]. El dispositivo Si1145 es capaz de operar desde 1.71 a 3.6 V sobre un rango de temperatura de -40 a +85 °C. En la tabla 4 se muestra sus características de operación.

Tabla 4. Características de operación Si1145.

Parámetro	Valor
Voltaje de alimentación	1.71 – 3.6 V DC
Interfaz	I2C, dirección: 0x60
Rango de operación índice UV	1 – 11
Rango de operación intensidad lumínica	1 – 128 Klx
Resolución	100 mlx

Los datos de medición de índice UV, luz visible e infrarroja son valores de 16 bits (entero sin signo). Para obtener el índice UV es dividir por 100 el valor entregado por el sensor. Y para calcular la intensidad lumínica se realiza el siguiente proceso:

- Tomar la medida de luz visible e infrarroja colocando el sensor en oscuridad absoluta (*darkVIS* y *darkIR*).
- Leer los parámetros *gainVIS* y *gainIR* que corresponden al valor de la ganancia para la luz visible e infrarroja, respectivamente.
- Determinar el rango visible (*rangeVIS*) e infrarrojo (*rangeIR*) de operación del sensor, el cual puede operar en un rango normal de la señal o en una alta sensibilidad. Para ello se usan las siguientes expresiones:

$$\begin{aligned} \text{rangeVIS} &= 1 + 13.5 * \text{miscVIS} \\ \text{rangeIR} &= 1 + 13.5 * \text{miscIR} \end{aligned}$$

Donde *miscVIS* y *miscIR* representan la selección del modo de operación para la luz visible e infrarroja, respectivamente, un valor de cero indica un rango normal y un valor de uno determina un alto rango de la señal, este último modo es útil en operaciones bajo la luz solar directa.

- Finalmente, se aplicará la siguiente ecuación para calcular los luxes:

$$lux = \frac{5.41 * rangeVIS * (VIS - darkVIS)}{2^{gainVIS}} + \frac{-0.08 * rangeIR * (IR - darkIR)}{2^{gainIR}}$$

En donde *VIS* e *IR* son las medidas entregadas por el sensor para la luz visible e infrarroja, respectivamente.

La conexión de los sensores SHT31 y Si1145 se realiza por el terminal 38 del microcontrolador y el terminal 37.

Sensor de temperatura en el agua

En este trabajo se seleccionó, el sensor DS18B20 en su forma de sonda, ya que es ideal para su uso en el agua [24]. Cuenta con tres terminales: VCC, GND y el pin DATA. El sensor utiliza la comunicación OnWire, es decir solo requiere una sola línea de datos (y tierra) para la comunicación con el microprocesador. Además, cada sensor tiene un código único de 64 bits que le permite compartir un mismo bus OneWire con otros dispositivos. Los datos de temperatura se almacenan como un número en complemento a dos de 16 bits, con signo extendido en el registro de temperatura. Sus características de operación se muestran en la tabla 5.

Tabla 5. Características de operación DS18B20

Parámetro	Valor
Voltaje de alimentación	3 – 5.5 V DC
Interfaz	OnWire
Rango de operación	-55 – 125 °C
Precisión	± 0.5°C En condiciones de: -10 a 85°C
Resolución seleccionable	9 – 12 bits

Sensor del nivel de PH en el agua

El sensor de PH seleccionado fue el DFRobot [25] modelo H1-101, que suministra una señal analógica directamente proporcional a la medición de PH a través de un conector BNC y un circuito acondicionador que transforma la medida a un valor en el rango de 0 – 5 V DC. Las principales especificaciones del sensor se muestran en la tabla 6.

Tabla 6. Características de operación sensor pH H-101.

Parámetro	Valor
Voltaje de alimentación	5 V DC
Señal de salida	Análoga
Rango de operación	0 – 14 pH En condiciones de 0 – 60 °C
Precisión	± 0.01 pH
Tiempo de respuesta	< 1 min
Potenciómetro	Potenciómetro de ajuste de ganancia
Conector del sensor	Conector BNC
Interfaz	pH2.0 conector de 3 pines

Sensor de conductividad eléctrica en el agua

Para la medida de la conductividad eléctrica se utilizó el sensor DRF0300 [26], que requiere también de un sensor de temperatura ya que es necesario de una compensación de la temperatura para realizar correctamente la medida de conductividad. Las principales especificaciones del sensor de conductividad eléctrica se muestran en la tabla 7.

Tabla 7. Características de operación sensor conductividad eléctrica DRF0300.

Parámetro	Valor
Voltaje de alimentación	5 V DC
Señal de salida	Análoga
Rango de operación	1 – 20 ms/cm En condiciones de 5 – 40 °C
Precisión	± 10% Depende de la solución de calibración
Tiempo de respuesta	< 1 min
Conector del sensor	Conector BNC
Interfaz	pH2.0 conector de 3 pines

Conversor análogo digital (ADC)

Este circuito tiene como objetivo digitalizar las señales analógicas de pH y conductividad eléctrica (CE). El circuito está basado en el integrado AD7792, el cual es un convertidor tipo Sigma Delta con resolución de hasta 24 bits [27]. En la figura 9 se muestra el circuito para ADC. El convertidor tiene tres canales diferenciales y su salida se transmite puede ser leída a través de una interfaz SPI. La referencia de tensión y el reloj, necesarios para la operación del convertidor, están integrados. Además, dispone de un sistema interno de multiplexado y

amplificación de bajo ruido, haciéndolo ideal para aplicaciones de alta precisión.

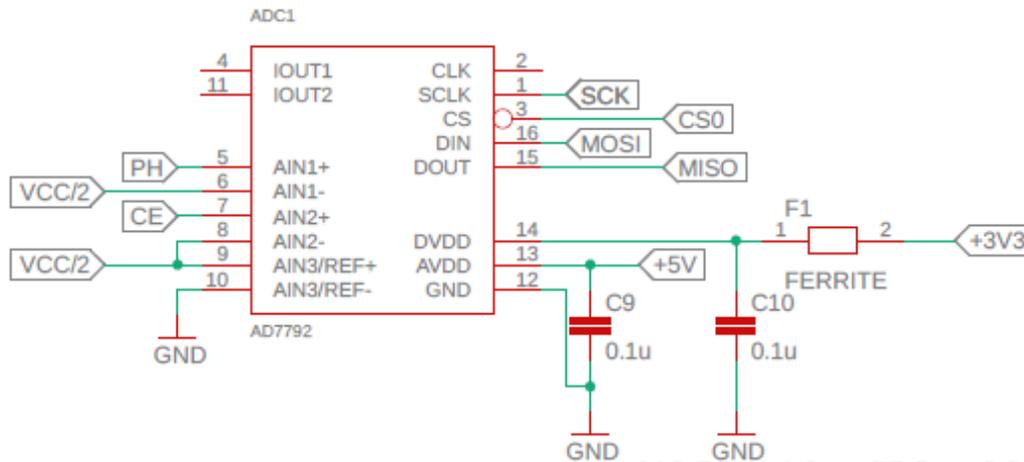


Figura 9. Conversor análogo digital.

El convertidor analógico digital trabaja con una referencia, V_{REF} , de $V_{CC}/2$ (2.5V) y el rango de medida para la señal analógica en su entrada positiva es de 0 – 5V. Debido a la topología Sigma Delta del convertidor, se dispone de una serie de filtros digitales preestablecidos. Se ha empleado un filtro digital que permite tener un rechazo simultaneo de señales a 50Hz y 60Hz (como las provenientes de redes eléctricas, principal componente de ruido en muchos sistemas sensoriales), con 65dB de atenuación. Esto limita el tiempo de conversión a 120ms por muestra.

El código binario de salida para las entradas de voltaje análogo puede ser representado como:

$$Codigo = 2^{N-1} \left(\frac{AIN * GAIN}{V_{REF}} + 1 \right)$$

Donde AIN es la entrada de voltaje análogo, $GAIN$ es la ganancia interna del ADC y N es el número de bits ($N = 16$).

Debido a que el nodo sensor es alimentado con una fuente de +5V, para el proceso de conversión de las señales analógicas a digitales, es necesario que el voltaje de referencia sea de 2.5V. Para obtener este voltaje, se utiliza el circuito integrado TLE2426CD, el cual entrega siempre a su salida una tensión igual a la mitad de la tensión de entrada. En la figura 10 se muestra el circuito del voltaje de referencia de 2.5V.

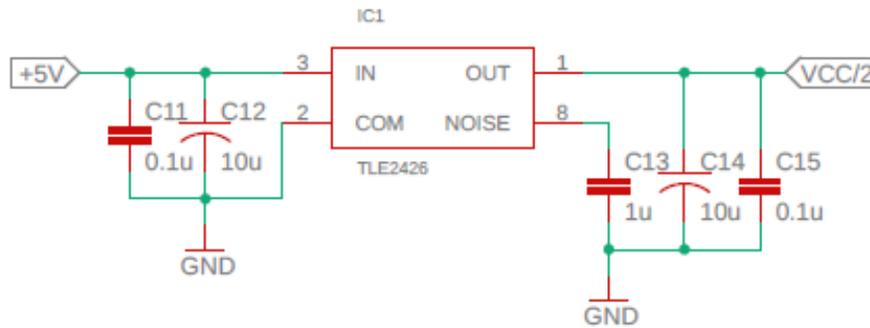


Figura 10. Circuito voltaje 2.5V.

Módulo LoRa

El módulo inalámbrico para la comunicación LoRa utilizado es el E22-900T22, este chip está basado en el transceptor de radio SX1262. Su método de modulación LoRaTM se basa en tecnología de espectro ensanchado, aumentando la distancia de comunicación. El módulo tiene un algoritmo de corrección de errores hacia adelante, FEC, el cual tiene una alta eficiencia de codificación y una fuerte capacidad de corrección de errores. Además, tiene encriptación de datos y permite la configuración de la longitud del paquete de datos [28]. Las principales características de este módulo se presentan en la tabla 8 y en la figura 11 se muestra el esquema de conexión.

Tabla 8. Características de operación del módulo LoRa E22-900T22.

Parámetro	Valor
Frecuencia de trabajo	850.125 – 980.125 MHz
Potencia de transmisión	22 dBm
Velocidad de datos de aire	0.3kbps a 62.5kbps
Distancia de prueba	7000 m (En aire abierto y despejado, con potencia máxima, ganancia de antena de 5dBi, altura de 2m.
Voltaje de alimentación	3.3 – 5.5 V
Nivel de comunicación	3.3 V
Interfaz de comunicación	UART

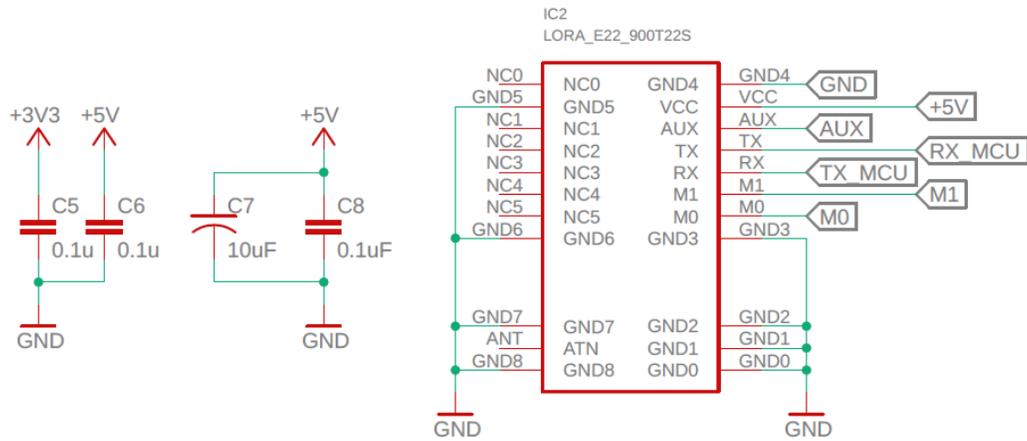


Figura 11. Esquema de conexión del módulo LoRa E22-900T2S.

Módulos Relés

El nodo sensor cuenta con dos módulos relés, cada uno tiene un transistor para su activación, poseen 3 terminales +5V, GND, y la entrada de señal de estado lógico, la cual puede ser accionada por el microcontrolador ESP32, pueden manejar cargas con una corriente máxima de 10A y hasta 250VAC. En la figura 12 se muestra el esquema de conexión para los relés.

Estos módulos relés permite controlar el encendido y apagado de aparatos, que se conecten a una fuente de alimentación eléctrica externa. Uno de los módulos relés es usado para accionar sobre una bomba de agua.

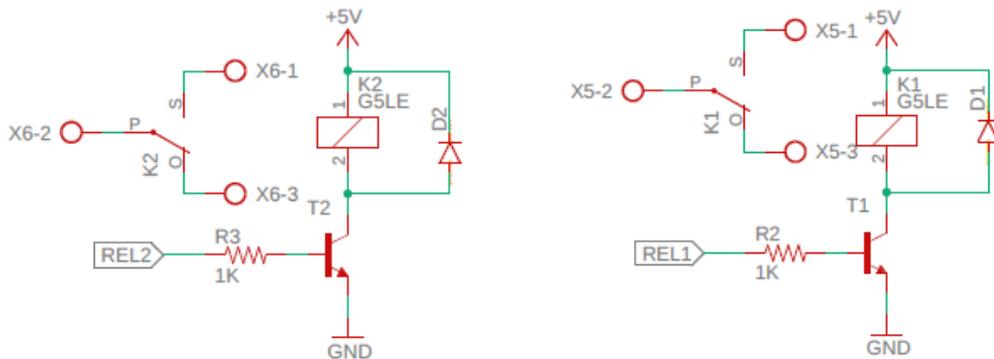


Figura 12. Esquema de conexión módulos relés.

5.2. Software para los nodos sensores.

Para el desarrollo del software del microcontrolador ESP32, se utilizó el entorno de desarrollo (IDE) de Arduino, el cual es una aplicación multiplataforma. El

código fuente para el IDE es abierto y permite los lenguajes de programación C y C++ [29].

En el proyecto, se usa la librería que creó la empresa Espressif Systems (fabricante del ESP32) para programar el microcontrolador usando IDE de Arduino, el cual permite desarrollar software para placas de otros proveedores [30].

El programa diseñado para cada nodo hace uso de un conjunto de componentes implementados, cada uno para desarrollar una tarea específica. En la figura 13 y 14 se muestra la estructura en capas de todos los componentes del sistema desarrollado a nivel de software. En la franja de color gris, se encuentran las librerías que permiten el uso de los periféricos del microcontrolador. Luego, en la franja azul están las librerías que permiten el uso de dispositivos externos, como el ADC, sensores, relés y módulo LoRa. En las franjas superiores, se encuentran las aplicaciones que permiten la configuración del nodo sensor, la transmisión de datos de los sensores, el control de los actuadores y la gestión de ahorro de energía.



Figura 13. Estructura del software para el nodo ambiente.



Figura 14. Estructura del software para el nodo agua

A continuación, se describen los componentes del software:

I2C

I2C (Inter-Integrated Circuit) es un protocolo de comunicación en serie y síncrono, destinado a permitir que múltiples circuitos integrados digitales, llamados esclavos, se comuniquen con uno o más circuitos integrados, llamados maestros. El bus I2C solamente necesita dos líneas, la de datos (SDA) y la de reloj (SCL), permite el intercambio de información a una velocidad de 100 Kbit/s en modo estándar o 400 Kbit/s en modo rápido. El protocolo I2C utiliza transferencias de 8 bits, cada dispositivo esclavo tiene una dirección de 7 bit y debe ser única.

La comunicación es iniciada por el dispositivo maestro, este envía una secuencia de inicio (START), cambiando la línea SDA de un nivel de voltaje alto a bajo antes de cambiar la línea SCL de alto a bajo, luego se transmite la dirección del dispositivo esclavo con el que desea comunicarse, junto con el bit que indica si es una operación de lectura o escritura. El dispositivo esclavo que coincide con la dirección envía un bit de ACK, colocando la línea SDA en bajo durante un bit. Posteriormente el maestro envía o recibe la trama de datos, después de la transmisión de datos el dispositivo receptor confirma la correcta operación enviando un bit de ACK. Finalmente, para detener la transmisión de datos, el dispositivo maestro envía una secuencia de parada (STOP), cambiando la línea SCL a alto antes de cambiar la línea SDA a alto. En la figura 15 se muestra el funcionamiento para la transmisión de datos en el bus I2C.

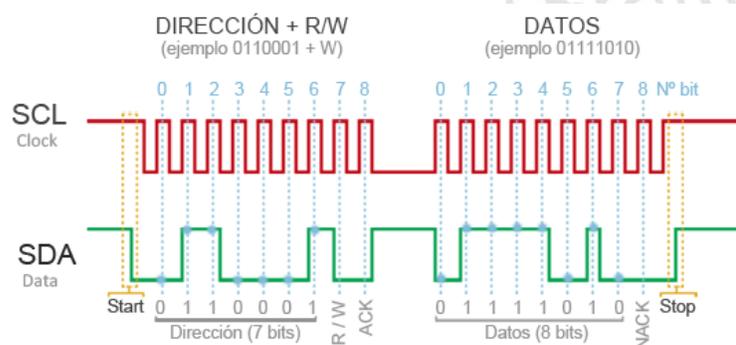


Figura 15. Trama de datos por el bus I2C. Tomado de [31]

En este sistema de monitoreo el bus I2C está formado por un dispositivo maestro, el microcontrolador ESP32 y dos esclavos, los sensores SHT31 y Si1145. En la figura 16 se ilustra el esquema de conexión para el bus I2C.

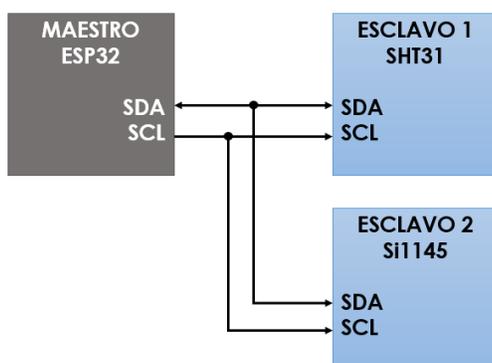


Figura 16. Esquema de conexión para el bus I2C.

La librería I2C es suministrada por el fabricante del microcontrolador ESP32. En la tabla 9 se describen las funciones utilizadas para la configuración y el funcionamiento del periférico I2C.

Tabla 9. Funciones usadas para el manejo del periférico I2C.

Función	Descripción
<code>bool Wire.begin(int sda=-1, int scl=-1, uint32_t frequency=0)</code>	Inicializa el periférico I2C como dispositivo maestro a una determinada frecuencia. Retorna verdadero, si la inicialización del bus I2C es exitosa.
<code>void Wire.beginTransmission(uint8_t address)</code>	Comienza una transmisión con un dispositivo esclavo cuya dirección es recibida como parámetro a través de la variable <code>address</code> .
<code>size_t Wire.write(uint8_t data)</code>	Escribe datos al dispositivo esclavo y retorna 1 si no se presentan errores.
<code>uint8_t Wire.endTransmission(void)</code>	Finaliza la transmisión con un dispositivo esclavo y transmite los datos que están en la cola del buffer.
<code>uint8_t Wire.requestFrom(uint8_t address, uint8_t size)</code>	Lee datos de un dispositivo esclavo retornando al número de bytes leídos.
<code>int Wire.available(void)</code>	Retorna el número de bytes disponibles para ser leídos.

UART

Se utiliza el periférico UART (Universal Asynchronous Receiver-Transmitter) para comunicación con el módulo LoRa. La librería UART es suministrada por el fabricante y en la tabla 10 se describen las funciones utilizadas.

Tabla 10. Funciones usadas para el manejo del periférico UART.

Función	Descripción
<code>void Serial.begin(unsigned long baud, uint32_t config=SERIAL_8N1, int8_t rxPin=-1, int8_t txPin=-1, bool invert=false, unsigned long timeout_ms = 20000UL)</code>	Inicializa el periférico UART definiendo la velocidad de transmisión, el formato de trama, el tiempo de espeta y los pines para RX y TX.
<code>int Serial.available(void)</code>	Retorna el número de bytes disponibles para leer desde el puerto serie.
<code>size_t Serial.write(uint8_t data)</code>	Escribe datos de 8 bits en el puerto serie.
<code>int Serial.availableForWrite(void)</code>	Retorna el número de bytes disponibles para escribir en el puerto serie.
<code>void Serial.flush(void)</code>	Limpia el buffer de entrada.

WIFI

El periférico WIFI es un módulo que permite la conexión del dispositivo ESP 32 a una red inalámbrica de 2,4 GHz para la transmisión y recepción de datos. La librería WIFI es proporcionada por el fabricante y en la tabla 11 se muestran las funciones que fueron utilizadas.

Tabla 11. Funciones usadas para el manejo del periférico WIFI.

Función	Descripción
<code>static bool WiFi.mode(wifi_mode_t)</code>	Selecciona el modo de operación del módulo WIFI, posee 4 modos: WIFI_OFF para apagar el módulo, WIFI_STA para trabajar en modo estación, es decir, como un dispositivo que se conecta a un punto de acceso, WIFI_AP el módulo trabaja como un punto de acceso,

	es decir, genera su propia red WIFI al que se conectan el resto de dispositivos y WIFI_AP_STA el módulo puede trabajar como estación y como punto de acceso.
<pre>bool WiFi.softAP(const char* ssid, const char* passphrase = NULL, int channel = 1, int ssid_hidden = 0, int max_connection = 4)</pre>	<p>Genera la red WIFI retornando a verdadero si se ha creado correctamente. Recibe como parámetros: <code>ssid</code>, es el nombre de la red que se va a crear (máximo 63 caracteres), <code>passphrase</code> es la contraseña de la red (mínimo 8 y máximo 32 caracteres), <code>channel</code> es el canal de la red (del 1 al 13), <code>ssid_hidden</code> este parámetro permite hacer visible la red, si toma un valor de 0, pero si su valor es 1 mantiene la red oculta y <code>max_connection</code> define el número máximo de conexiones (máximo 8). Los últimos 4 parámetros son opcionales.</p>

EEPROM

Esta librería es dada por el fabricante del microcontrolador y permite la escritura y lectura de la memoria EEPROM del microcontrolador. Esta memoria es utilizada para almacenar variables de configuración del nodo sensor, como también parámetros de calibración de los sensores. En la tabla 12 se describen las funciones que fueron utilizadas.

Tabla 12. Funciones de la librería de la memoria EEPROM.

Función	Descripción
<pre>bool EEPROM.begin(size_t size)</pre>	<p>Inicializa la memoria EEPROM y debe llamarse cada vez que se realice una operación de escritura o lectura. El parámetro es el número de bytes que se desean usar. Retorna verdadero si no presenta errores.</p>

<code>size_t EEPROM.writeByte(int address, uint8_t value)</code>	Inicializa una operación de escritura de un dato de 1 byte en una dirección específica de la memoria EEPROM.
<code>size_t EEPROM.writeUShort(int address, uint16_t value)</code>	Inicializa una operación de escritura de un dato tipo entero de 16 bits sin signo comenzando en una dirección específica de la memoria EEPROM.
<code>size_t EEPROM.writeString(int address, String value)</code>	Inicializa una operación de escritura de una cadena de caracteres comenzando en una dirección específica de la memoria EEPROM.
<code>size_t EEPROM.writeFloat(int address, float_t value)</code>	Inicializa una operación de escritura de un dato tipo float comenzando en una dirección específica de la memoria EEPROM.
<code>bool EEPROM.commit()</code>	Guarda definitivamente los datos en la memoria EEPROM, pero no la cierra, es decir es posible seguir realizando operaciones en la memoria.
<code>void EEPROM.end()</code>	Guarda los datos y cierra la memoria EEPROM.
<code>uint8_t EEPROM.readByte(int address)</code>	Retorna el dato de 1 byte almacenado en una dirección específica de la memoria.
<code>String EEPROM.readString(int address)</code>	Retorna una cadena de caracteres almacenadas en la memoria. Recibe como parámetro la dirección de almacenamiento.
<code>uint16_t EEPROM.readUShort(int address)</code>	Retorna un entero sin signo de 16 bits almacenado en la memoria. Recibe como parámetro la dirección de almacenamiento.
<code>float_t EEPROM.readFloat(int address)</code>	Retorna el dato tipo float almacenado en la memoria. Recibe como parámetro la dirección de almacenamiento.

Librería SHT31

Esta librería fue desarrollada para configurar y leer los valores de temperatura y humedad relativa del ambiente del sensor SHT31. En la tabla 13 se muestran las funciones que contiene la librería.

Tabla 13. Funciones de la librería SHT31.

Función	Descripción
<code>void begin_SHT31(TwoWire *port)</code>	Inicializa el sensor SHT31 enviado por el bus I2C un comando de restablecimiento parcial, esto activa el sensor para restablecer su controlador del sistema y recargar los datos de calibración a la memoria. Recibe como parámetro un apuntador al objeto de la clase TwoWire.
<code>data_SHT31 readTempHumRel_SHT31(uint16_t mode)</code>	<p>Lee los valores de temperatura y humedad relativa, para ello se transmite un comando de disparo único especificando el modo de medición. EL SHT31 posee tres modos de medición SHT31_HIGH_REPEATABLITY, SHT31_MEDIUM_REPEATABLITY y SHT31_LOW_REPEATABLITY, la diferencia entre ellos es la duración de la medida, nivel de ruido y consumo de la energía. A mayor repetibilidad aumenta el tiempo de medida y la potencia consumida, pero disminuye el nivel de ruido.</p> <p>Después de enviar el comando de disparo único se procede a recibir los datos transmitidos por el sensor, se realiza una verificación del CRC, si no hay errores en la transmisión se hacen las conversiones para obtener el valor de la temperatura en grados centígrados y el valor de la humedad relativa. Finalmente, los valores son almacenadas en una estructura del tipo data_SHT31.</p>

Librería Si1145

Para configurar el sensor Si1145 y leer el índice UV y la intensidad lumínica se creó la librería Si1145, la cual esta compuesta por funciones que son mostradas en la tabla 14.

Tabla 14. Funciones de la librería Si1145.

Función	Descripción
<code>void begin_SI1145(TwoWire *port)</code>	Inicializa el sensor Si1145, desactivando el modo automático, configurando parámetros de calibración del sensor y habilitando la medición de luz visible, luz infrarroja y el índice UV. Recibe como parámetro un apuntador al objeto de la clase TwoWire.
<code>void setDarkValue_SI1145(uint16_t darkIRLight, uint16_t darkVLight)</code>	Permite ajustar valores para calibrar el sensor.
<code>data_SI1145 readUVLux_SI1145()</code>	Envía un comando al sensor para tomar una sola medición de la luz visible, luz infrarroja y el índice UV. Luego se calcula el valor de la intensidad lumínica en luxes usando la información de la luz visible e infrarroja. Finalmente, el valor de la intensidad lumínica y el índice UV son almacenados en una estructura de tipo data_SI1145.

SPI

SPI (Serial Peripheral Interface) es un protocolo síncrono, de datos seriales, que trabaja en modo full duplex, es decir, los dispositivos pueden actuar como transmisor y receptor al mismo tiempo.

En una conexión SPI siempre hay un maestro, que será el dispositivo encargado de transmitir información a los esclavos y de generar las señales de reloj y control. Un dispositivo esclavo se encarga de recibir y enviar información al maestro. Los dispositivos esclavos son controlados por la línea selectora, llamada Slave Select (SS), la cual se encarga de habilitar un esclavo. Usualmente hay tres líneas comunes tanto para maestros como esclavos, la línea MISO (Master In

Slave Out), utilizada para enviar datos que provienen del esclavo hacia el maestro, línea MOSI (Master Out Slave In), para enviar datos que provienen del maestro hacia en esclavo y la línea CLK (Clock), es la señal de reloj generada por el maestro para la sincronización de los dispositivos [31].

En este caso, el microcontrolador se comunica con el convertor análogo digital (ADC) a través del bus SPI. En la figura 17 se muestra el esquema de conexión.

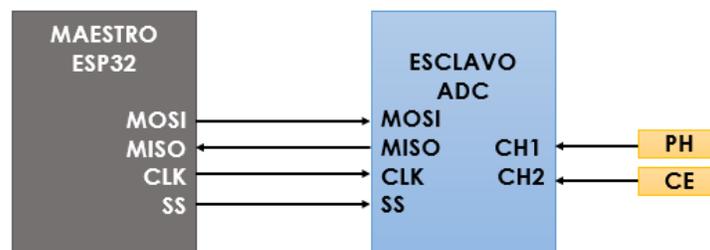


Figura 17. Esquema de conexión para el bus SPI.

La librería I2C es suministrada por el fabricante del microcontrolador ESP32, para su configuración y funcionamiento se utilizaron las funciones que son mostradas en la tabla 15.

Tabla 15. Funciones usadas para el manejo del periférico SPI.

Función	Descripción
<code>SPI.Settings(uint32_t clock, uint8_t bitOrder, uint8_t dataMode)</code>	Configura parámetros del bus SPI. <code>clock</code> es la velocidad de la señal de reloj, <code>bitOrder</code> define que bit se transmite primero, si toma el valor de 1 envía primero el bit más significativo, si es 0 envía primero el bit menos significativo y <code>dataMode</code> define cuatro modos de operación, los cuales son determinados por la señal de reloj cuando es bus está inactivo y el flanco en el cual se leen los datos.
<code>void SPI.begin(int8_t sck=-1, int8_t miso=-1, int8_t mosi=-1, int8_t ss=-1)</code>	Inicializa el periférico SPI definiendo una frecuencia de transmisión de 1MHz por defecto y los pines para las líneas del bus.
<code>void SPI.beginTransaction(SPISettings settings)</code>	Configura el periférico SPI para iniciar una comunicación, utilizando las <code>settings</code>

	configuraciones definidas por settings.
<code>void SPI.write(uint8_t data)</code>	Escribe un dato tipo entero sin signo de 8 bits en el bus SPI.
<code>void SPI.write16(uint16_t data)</code>	Escribe un dato tipo entero sin signo de 16 bit en el bus SPI.
<code>void SPI.write32(uint32_t data)</code>	Escribe un dato tipo entero sin signo de 32 bit en el bus SPI.

Librería ADC

Esta librería fue desarrollada para configurar el conversor análogo digital AD7792, que se encarga de la adquisición de datos de los sensores de PH y CE. En la tabla 16 se muestran las funciones contenidas en la librería.

Tabla 16. Funciones de la librería ADC.

Función	Descripción
<code>void ADC_begin()</code>	Inicializa el ADC enviado un comando de reset, seleccionado un rango de medida de 0V – 5V, tomando una ganancia 1 y activando el filtro de 50 Hz y 60 Hz. Finalmente, se realiza una calibración de medición para los dos canales.
<code>data_ADC ADC_Read_Data()</code>	Guarda en una estructura de tipo <code>data_ADC</code> , el valor en milivoltios medido por los canales 1 y 2 del ADC. Primero se envía un comando para que el ADC realice una conversión, luego se lee el registro de datos y finalmente se realiza la conversión del dato leído a milivoltios.
<code>uint8_t ADC_Read_Status()</code>	Permite leer el registro de estado, el cual contiene un bit que indica cuando se ha realizado una conversión del ADC.

ONE WIRE

One Wire es un protocolo serie diseñado por Dallas Semiconductor, la cual utiliza una sola línea de datos y referencia a tierra para su comunicación. Este bus solo permite un maestro para controlar e intercambiar datos con uno o varios esclavos conectados al mismo bus. Cada dispositivo tiene un número de identificación único e inalterable de 64 bits.

La librería fue creada para tener una comunicación con el sensor de temperatura DS18B20. En la tabla 17 se muestra la función utilizada.

Tabla 17. Función usada para comunicaciones One Wire.

Función	Descripción
<code>float readTemp_DS18B20()</code>	<p>Retorna al valor entregado por el sensor DS18B20 de la temperatura en grados centígrado, realizando los siguientes pasos:</p> <ul style="list-style-type: none">• Enviar el comando de reset al dispositivo• Enviar el comando que le indica al dispositivo que realice una conversión de temperatura.• Lee los datos entregados por el sensor.• Realiza la conversión del dato leído a grados centígrados.

Aplicación Principal

La aplicación puede arrancar en modo en configuración o en modo monitoreo. Para ingresar al modo configuración se debe poner en bajo el pin 26 del microcontrolador, a través de un pulsador ubicado en el circuito y luego presionar el botón de reset.

En modo configuración el microcontrolador corre un servidor web, el cual almacena una página html que puede ser descargada desde cualquier dispositivo conectada a la red WIFI generada por el ESP32, accediendo a la ip 192.168.4.1. En la figura 18 se muestra el diseño de página web para el nodo ambiente, en donde se pueden ingresar parámetros de ajustes para el sensor de luz, modificar el periodo de muestreo e ingresar parámetros de configuración para el módulo LoRa y la plataforma Sentilo. Y en la figura 19 se observa el diseño de página web para el nodo agua, la cual es similar a la mostrada en la figura 18, con la diferencia de que en esta se puede realizar calibraciones para los sensores de PH y CE.

NODO AMBIENTE

Calibrar sensor de luz

Ingresar el valor del sensor en ausencia de luz.

IR:

VIS:

Valor de IR en oscuridad:

Valor de VIS en oscuridad:

ENVIAR

Periodo de muestreo

Periodo de muestreo (segundos):

ENVIAR

Parámetros LoRa

Identificación de red (0 - 255):

Canal (0-80):

Canal Gateway(0-80):

Dirección (0 - 65534):

Dirección Gateway (0 - 65534):

Velocidad de transmisión inalámbrica (0 - 7):

Encriptación (0 - 65535):

ENVIAR

Parámetros Sentilo

Proveedor:

Nombre del sensor de temperatura:

Nombre del sensor de humedad relativa:

Nombre del sensor de luminosidad:

Nombre del sensor de índice UV:

ENVIAR

Figura 18. Diseño de la página web para el nodo ambiente.

NODO AGUA

Calibración PH

INICIAR

PH ÁCIDO

PH ACALINO

FINALIZAR

Calibración CE

INICIAR

CONFIRMAR

FINALIZAR

Periodo de muestreo

Periodo de muestreo (segundos):

ENVIAR

Parámetros LoRa

Identificación de red (0 - 255):

Canal (0-80):

Canal Gateway(0-80):

Dirección (0 - 65534):

Dirección Gateway (0 - 65534):

Velocidad de transmisión inalámbrica (0 - 7):

Encriptación (0 - 65535):

ENVIAR

Parámetros Sentilo

Proveedor:

Nombre del sensor de temperatura:

Nombre del sensor de conductividad eléctrica:

Nombre del sensor de PH:

ENVIAR

Figura 19. Diseño de la página web para el nodo agua.

En modo monitoreo la aplicación ejecuta tres máquinas de estados, las cuales se relacionan entre sí. La primera máquina de estados (**ahorro de energía**) se encarga de controlar el periodo de muestreo y poner al microcontrolador en modo de bajo consumo, cuando las demás máquinas de estados están inactivas. Esta máquina inicia en el estado Sleep, colocando al microcontrolador en bajo consumo y dejando activo el temporizador encargado de contar el periodo de muestreo y los pines que activan las interrupciones externas. La máquina cambiará de estado solamente si se cumple el periodo de muestreo o si hay una interrupción externa causada por la recepción de datos del módulo LoRa. Si el cambio de estado es debido a que se cumple el periodo de muestreo, se activará una bandera que le indica a la máquina de estado encargada del muestreo, que debe comenzar a recopilar los datos de los sensores para su posterior envío, si por el contrario el cambio de estado es causado por la interrupción externa, se activa una bandera que le indica la máquina de estados encargada de la comunicación, que debe recibir los datos y procesar los comandos. En la figura 20 se muestra el diagrama de estados de la máquina de ahorro de energía.

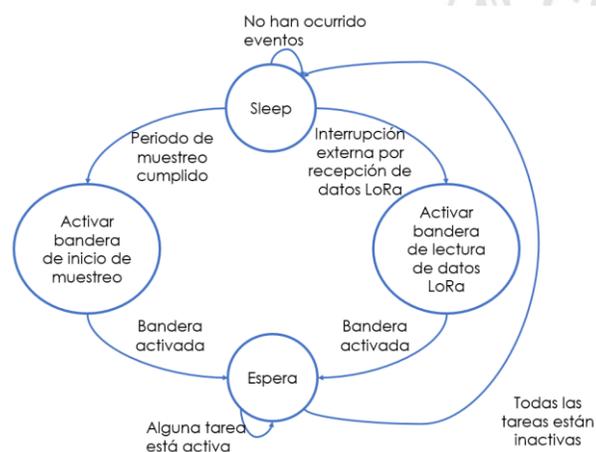


Figura 20. Máquina de estados ahorro de energía.

La máquina de **muestreo** inicia en un estado inactivo, a la espera de que se active la bandera que le indica que se ha cumplido un periodo de muestreo, una vez activada la bandera pasa a leer los datos de los sensores para luego almacenarlos en formato Json y transmitirlos al Gateway a través del protocolo de comunicación LoRa. Luego, la máquina de estados se queda esperando la confirmación de recibido del Gateway, si pasado 5 segundos no recibe la confirmación realiza una retransmisión de los datos, esta operación la repite durante tres veces consecutivas. La máquina pasa al estado inactivo nuevamente, si recibe la confirmación del Gateway o si después de tres intentos de transmisión, no recibió dicha confirmación. En la figura 21 se muestra el diagrama de estados de la máquina de muestreo.

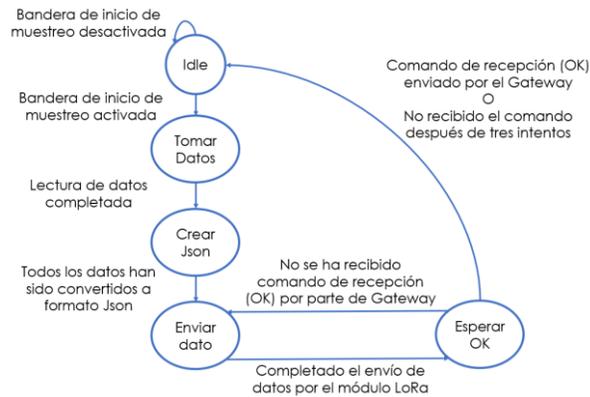


Figura 21. Máquina de estados muestreo.

La máquina de **comunicación** inicia en un estado inactivo, si se activa la bandera de configuración, la máquina pasa a modificar los parámetros del módulo LoRa, luego regresa nuevamente al estado inactivo. Si se activa la bandera de envío de datos, la máquina pasa a un estado de transmisión de datos en formato Json. Pero si se activa la bandera de recepción de datos, la máquina pasa a un estado de espera hasta cuando haya un dato disponible, cuando se confirma que hay un dato válido se procede a la lectura e interpretación de los datos. Si la trama recibida comienza con el carácter '0' se activa la bandera de confirmación de recepción de dato. Si la trama recibida comienza con el carácter '1' o '2' se activa o desactiva el rele 1 o el rele 2 respectivamente, dependiendo del valor del segundo carácter recibido. Si la trama recibida comienza con el carácter '3' se modifica el periodo de muestreo. Después de cada uno de los anteriores casos la máquina regresa al estado inactivo. En la figura 22 se muestra el diagrama de estados de la máquina de ahorro de comunicación.

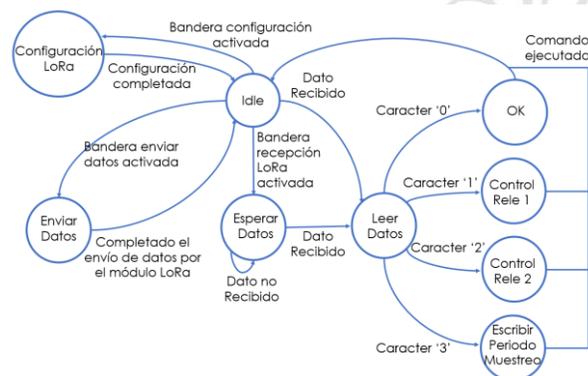


Figura 22. Máquina de estados comunicación.

5.3. Implementación de la red de sensores basado en protocolo de comunicación LoRa.

El sistema para el monitoreo de cultivos hidropónicos está formado por dos dispositivos de medida llamados nodos sensores. La información enviada por los nodos es recibida por un servidor central. En la figura 23 se muestra un diagrama de bloques de la red inalámbrica de sensores.

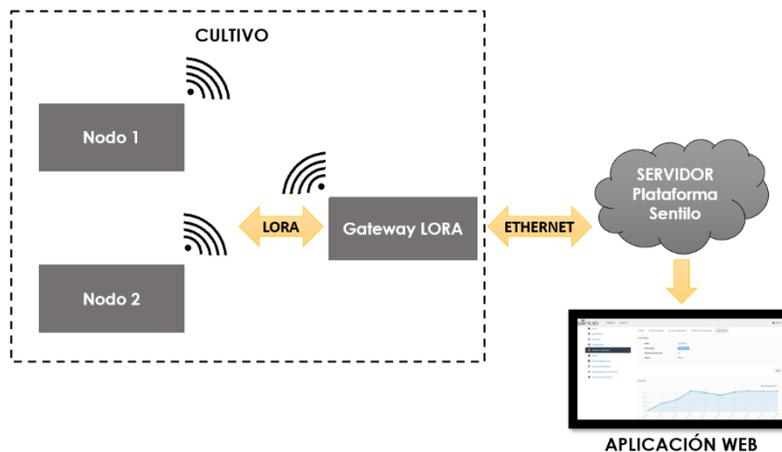


Figura 23. Diagrama de bloques de la red de sensores.

La información transmitida por los nodos a través del protocolo de comunicación LoRa es recibida por el **Gateway**, el cual actúa como puente para enviar los datos a una red superior como internet. Este es implementado mediante un ordenador de placa reducida (SBC, Simple Board Computer) desarrollado por la Fundación Raspberry Pi. El modelo utilizado es la Raspberry Pi 4, mostrado en la figura 24, fue lanzado al mercado en junio de 2019, dentro de sus características encontramos un procesador quad-core Cortex-A72 con frecuencia de reloj de 1.5GHz, 2GB de memoria RAM y posibilidad de conexión inalámbrica a través de WiFi y Bluetooth. A la Raspberry se le conecta un shield LoRa, basado en el módulo E22-900T22, el mismo que fue usado en los nodos sensores.



Figura 24. Raspberry pi 4 con shield LoRa.

La raspberry pi utiliza como sistema operativo Raspbian, una distribución de Linux basada en Debian. Para el desarrollo del software ejecutado por el Gateway se utiliza NODE-RED, la cual es una herramienta de programación basada en JavaScript, este se programa utilizando flujos que se ejecutan de forma paralela, usando como interfaz de desarrollo un navegador web.

El programa que se desarrolló para el Gateway recibe los datos transmitidos por el nodo sensor, luego convierte la información recibida a un objeto Json, posteriormente se toman diferentes acciones dependiendo del tipo de información que ha sido enviada por el nodo sensor, si es de tipo **datos** es porque contiene las medidas de los sensores, entonces se transmiten estas medidas a la plataforma IOT Sentilo, si es de tipo **respuesta** es porque contiene el estado de los relés o el valor del periodo de muestreo, en estos casos se publica dicha información a través del protocolo MQTT.

Además, si el Gateway recibe a través de la red ethernet una petición de activación o desactivación de los relés o un cambio en el periodo de muestreo, el Gateway se encarga de enviar la información a los nodos sensores correspondientes. En la figura 25 se observa el diagrama de flujo implementado en NODE-RED.

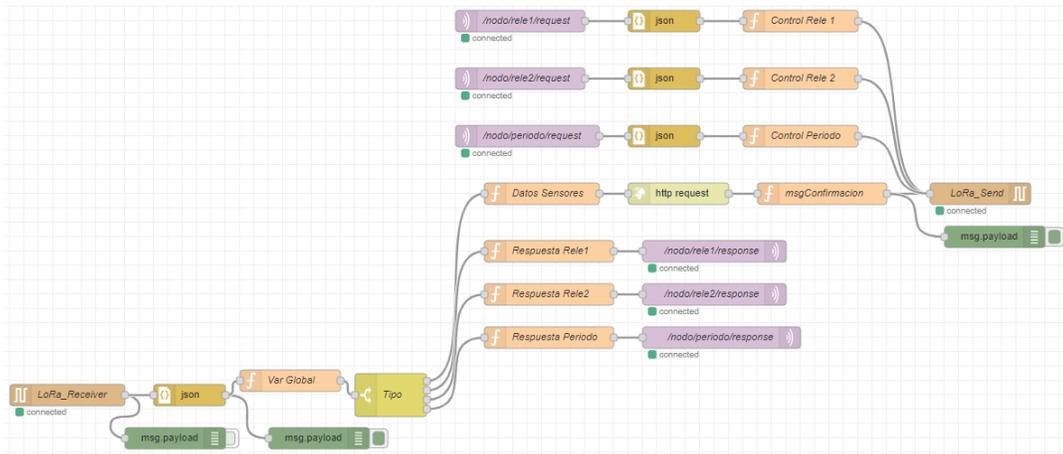


Figura 25. Diagrama de flujo para el Gateway.

5.4. Implementación de la plataforma IoT basada Sentilo.

La plataforma de Sentilo fue instalada en un servidor Linux que pertenece al grupo de investigación SISTEMIC, ubicado en la sede central de la Universidad de Antioquia. Para su instalación se utilizó una máquina virtual que se descargó de la página oficial de Sentilo.

Sentilo está compuesto por un catálogo web, a través de este se crearon los sensores, componentes, proveedores y aplicaciones. En la figura 26 se muestra la aplicación web de Sentilo, en el menú lateral izquierdo se pueden acceder a los recursos ofrecidos por la plataforma, en la parte derecha se observa los elementos creados por el usuario y la posibilidad de crear o eliminar.

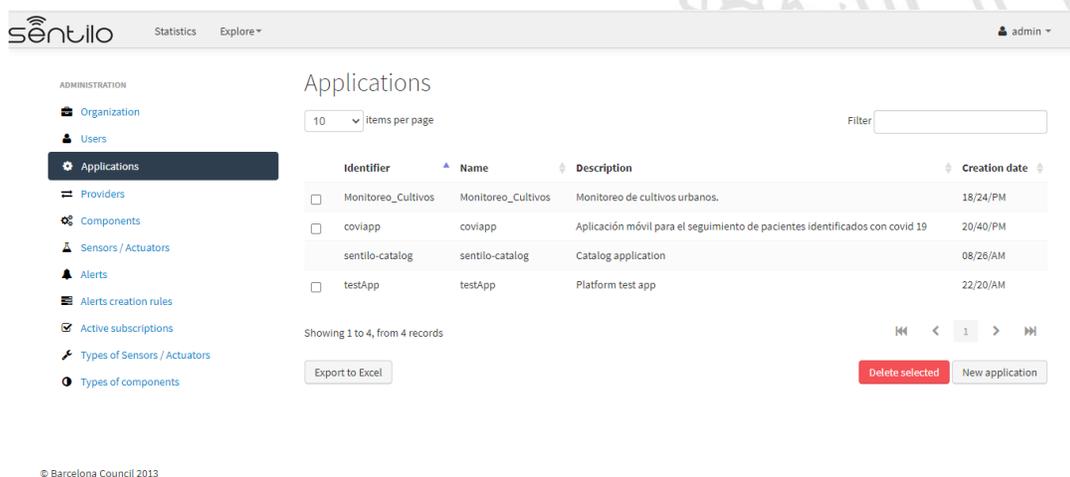


Figura 26. Aplicación web de Sentilo.

En la figura 27 se muestra los elementos que se crearon en la plataforma Sentilo, la cual está compuesta por una aplicación, un proveedor, dos componentes y siete sensores. Para identificar componentes, proveedores y sensores se tienen en cuenta las siguientes reglas:

- El identificador del proveedor comienza con la palabra "Cultivo" seguido de un código que representa el número del cultivo. Ejemplo: Cultivo1
- A cada proveedor se le asocian dos componentes, se deben nombrar comenzado con la palabra "Nodo" seguido del código asociado al proveedor al cual pertenece, luego le asigna el tipo de componente que representa, en donde puede tomar dos opciones: ser un nodo tipo ambiente escribiendo "_1" o nodo tipo agua "_2". Por ejemplo, "Nodo1_1" indica que el componente pertenece al proveedor Cultivo1 y es un nodo tipo ambiente.
- Para nombrar los sensores estos deben comenzar con el nombre del componente al cual pertenecen, seguido del tipo de variable que está midiendo el sensor, las cuales pueden ser: "_TEMP" sensor de temperatura, "_HR" sensor de humedad relativa, "_LUX" sensor de intensidad lumínica, "_UV" sensor de índice UV, "_PH" sensor de PH y "_CE" sensor de conductividad eléctrica. Por ejemplo, "Nodo1_2_PH" indica que el sensor pertenece al componente "Nodo1_2" y está midiendo la variable de PH.

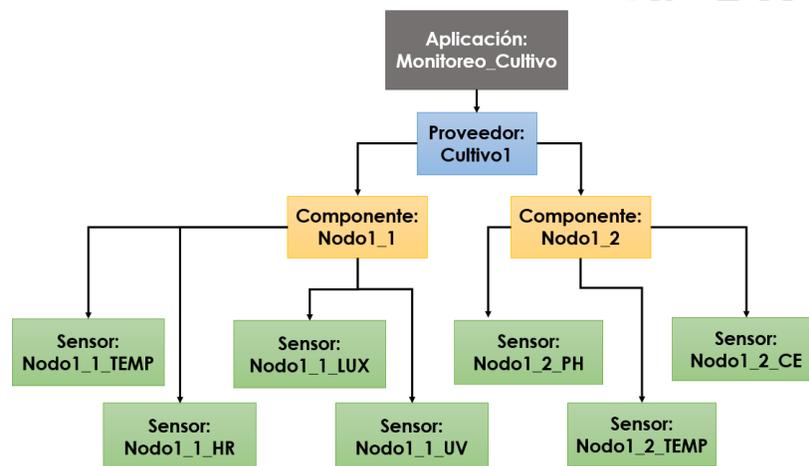


Figura 27. Estructura de la aplicación en la plataforma Sentilo.

Sentilo ofrece una API de código abierto basada en interfaces Representational State Transfer (REST) y la comunicación con Sentilo fue a través del protocolo de transferencia de hipertexto (HTTP). Los métodos del protocolo HTTP utilizados por

Sentilo son: GET para solicitar información, POST enviar datos nuevos, PUT actualizar datos existentes y DELETE para eliminar datos.

Después de realizar una petición, la plataforma responde con un código que indica el estado de la solicitud. En la tabla 18 se detalla los códigos de respuesta.

Tabla 18. Códigos de respuesta ante una petición HTTP.

Código	Descripción
200	Solicitud aceptada y procesada correctamente.
4xx	Error de cliente.
401	Solicitud no autorizada: credencial vacía o no válida.
403	Prohibido: no autorizado para la acción solicitada.
5xx	Error del servidor.

En el sistema de monitoreo del cultivo se hace uso del servicio data, el cual permite leer, escribir o eliminar observaciones de los sensores. Todas las solicitudes para este servicio tienen el siguiente formato:

`http://<your_api_server.com>/data/<provider_id>/<sensor_id>`

En donde `<your_api_server.com>` es el dominio del servidor que ejecuta la plataforma, `<provider_id>` y `<sensor_id>` corresponden a los identificadores del proveedor y del sensor respectivamente con el cual se desea realizar una operación.

Las operaciones que se utilizaron en este servicio fueron:

- **Publicar observaciones de sensores de un proveedor**

Esta acción permite a un proveedor escribir observaciones realizadas por más de un sensor en un solo mensaje y tiene el siguiente formato:

`http://<your_api_server.com>/data/<provider_id>`

En este caso se utiliza el método PUT, el cuerpo del mensaje es en formato Json y se debe tener permiso de escritura. Esta acción devuelve alguno de los códigos de respuesta descritos anteriormente en la tabla 18.

Por ejemplo, si se desea publicar observaciones de los sensores de PH y CE en el proveedor "Cultivo1" se debe usar el siguiente formato:

```
http://sistemic.udea.edu.co:4000/data/Culitvo1
```

y en el cuerpo del mensaje:

```
{"sensors":[
  {
    "sensor":"Nodo1_2_PH",
    "observations":[
      {"value":"6.5"}
    ]
  },{
    "sensor":"Nodo1_2_CE",
    "observations":[
      {"value":"1.75"}
    ]
  }
]}
```

- **Leer las observaciones de los sensores de un proveedor**

Esta acción permite leer las últimas observaciones de los sensores asociados a un proveedor. Además, tiene la opción de indicar el número máximo de datos a recuperar y/o filtrar por un periodo de tiempo determinado. En este caso se debe tener permiso de lectura y utilizar el método GET. Esta acción devuelve en formato Json el valor de las observaciones, la fecha y hora en el cual el dato fue realizado. Tiene el siguiente formato:

```
http://<your_api_server.com>/data/<provider_id>?<parameter>=<value>
```

En donde <parameter>=<value> son los parámetros que relacionan el número de observaciones a leer y el periodo de tiempo a consultar. Por ejemplo, si se desea recuperar las últimas 10 observaciones de los sensores asociados al proveedor "Cultivo1", a partir de una fecha determinada se utiliza el siguiente formato:

```
sistemic.udea.edu.co:4000/data/Cultivo1?limit=10&from=10/09/2020T20:00:00
```

6. Resultados y análisis

6.1. Nodo sensor.

Se realiza el diseño de una tarjeta de circuito impreso (PCB, printed circuit board) para la implementación física de los dos nodos sensores. El diseño del PCB es a dos capas para cada nodo. En la figura 28, se muestra el PCB del nodo ambiente con una vista desde las dos capas. El tamaño es de 7.7 cm x 8.2 cm. Además, se muestra los sensores y la antena conexión Lora. En la figura 29 se muestra el PCB construido, con una vista desde las dos capas para el nodo agua, el cual tiene una dimensión de 7.7 cm x 8.2 cm. Cada nodo tiene dos relés, con objeto de integrar la capacidad de implementar acciones de control como encender o apagar la bomba de suministro de agua para el sistema de riego.

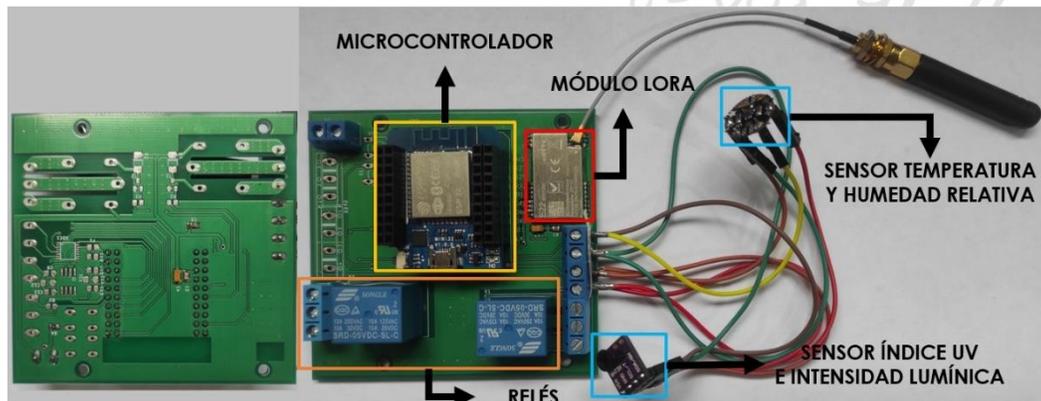


Figura 28. Circuito final del nodo ambiente.

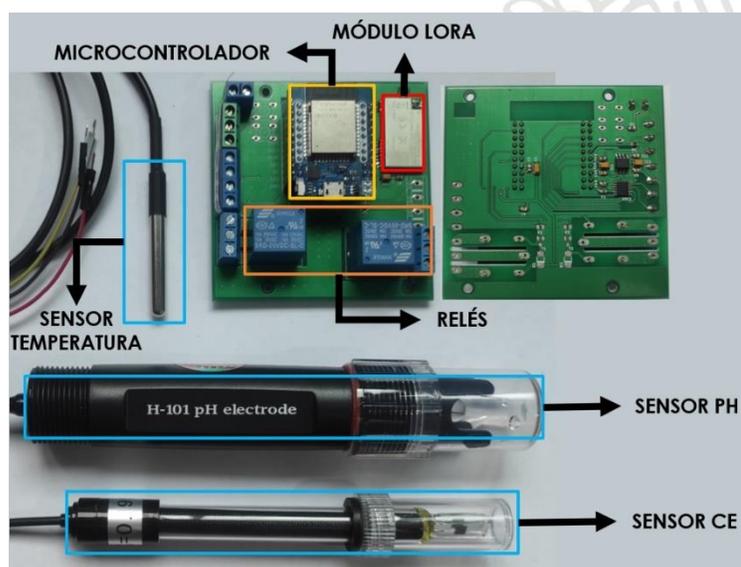
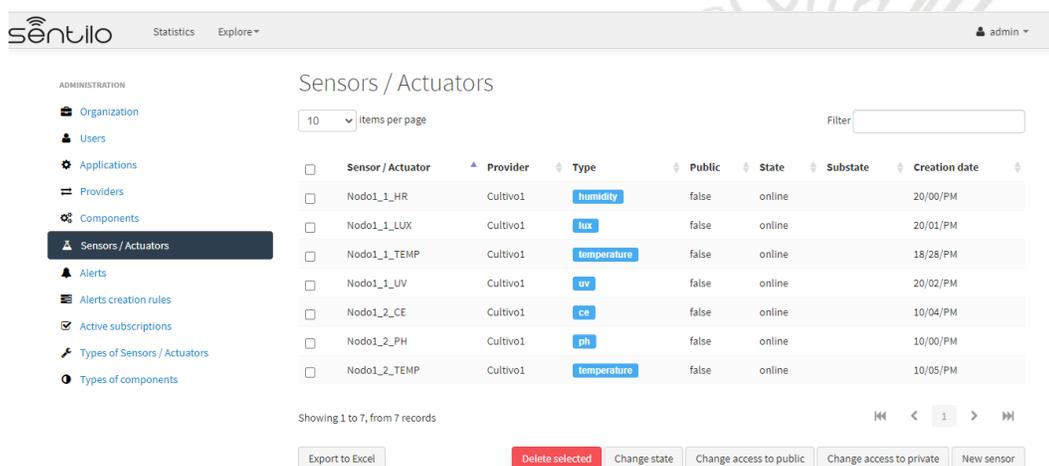


Figura 29. Circuito final del nodo agua.

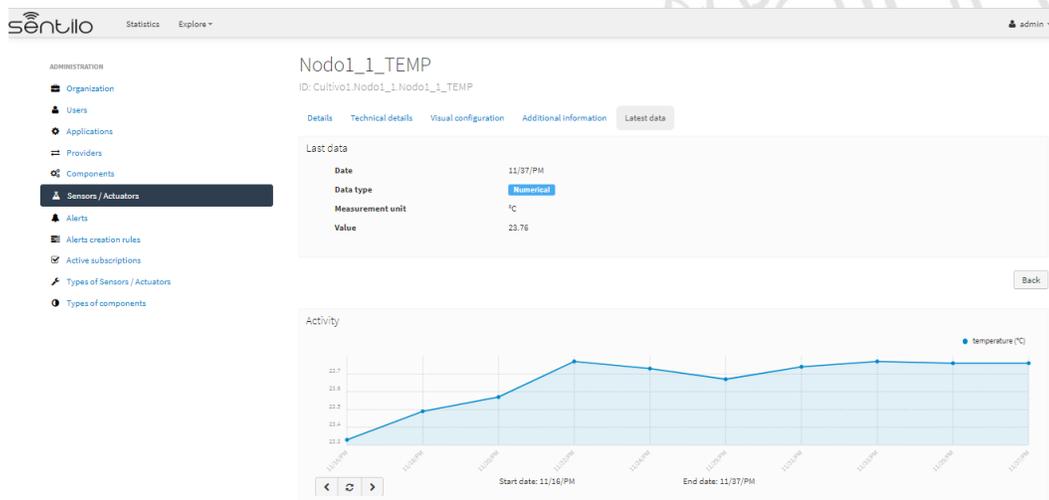
6.2. Plataforma IoT basada en Sentilo

La interfaz de la plataforma IOT con Sentilo se muestra en la figura 30, en la cual se puede apreciar una lista de los sensores configurados para el cultivo. A través de esta plataforma se logra almacenar los datos enviados por los sensores y tener acceso remoto a ellos, permitiendo de esta manera monitorear el cultivo desde cualquier lugar. Además, la plataforma permite visualizar mediante una gráfica en tiempo real los últimos datos tomados por cada sensor, en la figura 31 se observa los datos asociados al sensor de temperatura para el nodo ambiente.



<input type="checkbox"/>	Sensor / Actuator	Provider	Type	Public	State	Substate	Creation date
<input type="checkbox"/>	Nodo1_1_HR	Cultivo1	humidity	false	online		20/00/PM
<input type="checkbox"/>	Nodo1_1_LUX	Cultivo1	lux	false	online		20/01/PM
<input type="checkbox"/>	Nodo1_1_TEMP	Cultivo1	temperature	false	online		18/28/PM
<input type="checkbox"/>	Nodo1_1_UV	Cultivo1	uv	false	online		20/02/PM
<input type="checkbox"/>	Nodo1_2_CE	Cultivo1	ce	false	online		10/04/PM
<input type="checkbox"/>	Nodo1_2_PH	Cultivo1	ph	false	online		10/00/PM
<input type="checkbox"/>	Nodo1_2_TEMP	Cultivo1	temperature	false	online		10/05/PM

Figura 30. Plataforma IOT con Sentilo, para procesamiento de datos.



Latest data

Date	11/37/PM
Data type	Numerical
Measurement unit	°C
Value	23.76

Activity

Start date: 11/16/PM End date: 11/37/PM

Figura 31. Plataforma Sentilo visualización del sensor de temperatura para el nodo ambiente

6.3. Prueba de funcionamiento del sistema.

El sistema de monitoreo fue puesto a prueba, midiendo los parámetros ambientales y del agua. En las figuras 32, 32, 34 y 35 se muestran los valores obtenidos en una prueba, por los sensores entre las 12 del medio día y las 6 de la tarde, tomando datos cada 7 minutos, en un día sin lluvia. El lugar del cultivo está protegido por un techo traslúcido, por tal motivo la radiación UV presente en dicho cultivo es baja, como se puede observar en la figura 33.

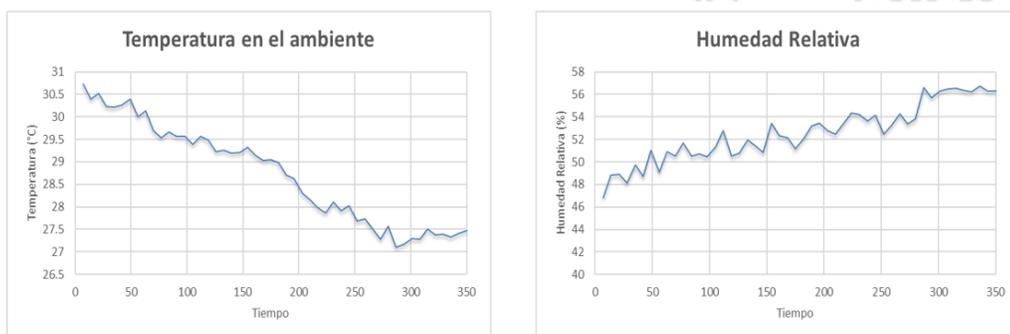


Figura 32. Gráfica de las medidas de temperatura y humedad relativa del ambiente.

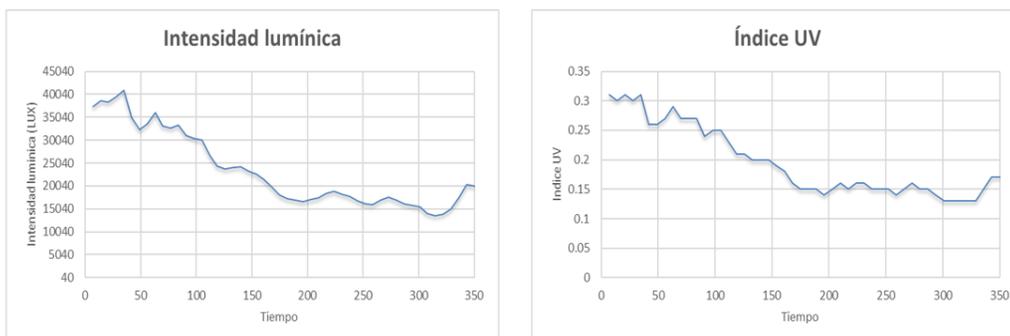


Figura 33. Gráfica de medidas de la intensidad luminosa e índice UV.

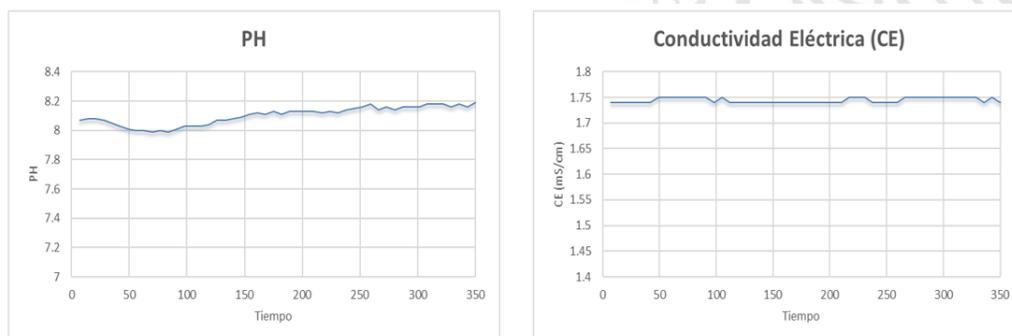


Figura 34. Gráfica de medidas de PH y CE

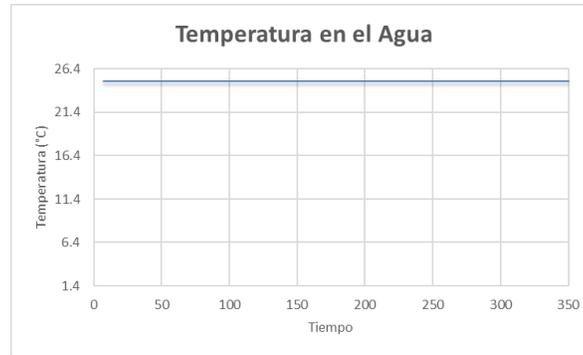


Figura 35. Gráfica de medias de la temperatura en el agua.

6.4. Prototipo del cultivo vertical.

Se desarrolla un prototipo a pequeña escala de un cultivo vertical hidropónico piramidal con sistema NTF, tiene un tamaño de 80 cm de ancho, 175 cm de largo y 200 cm de alto.

La estructura que soporta los tubos de siembra está hecha en madera y tiene forma piramidal, aprovechando el espacio vertical. El sistema cuenta con 12 tubos de PVC de 2" con un largo de 150 cm, cada tubo tiene 7 perforaciones de 3.8 cm de diámetro y una separación entre ellas 17 cm, dentro de cada perforación se encuentra la canastilla que sostiene la planta durante su crecimiento. Los tubos están interconectados entre sí mediante tubería de $\frac{3}{4}$ ". El agua con nutrientes, que recircula a través de la tubería, es almacenada en un recipiente con capacidad de 50 litros aproximadamente y es impulsada mediante una electrobomba de $\frac{1}{2}$ hp. El prototipo del cultivo se muestra en la figura 36.



Figura 36. Prototipo del cultivo vertical hidropónico.

7. Conclusiones

En este trabajo se presenta el desarrollo de un sistema IoT para el manejo de cultivos de hortalizas verticales. El sistema fue diseñado para integrar la capacidad de detección de los parámetros más importantes para el manejo de cultivos. Se diseñaron dos nodos sensores, uno para monitorear las condiciones del agua (PH, conductividad y temperatura) que conduce los nutrientes a las plantas y, el otro para monitorear las condiciones ambientales (temperatura, humedad relativa, índice UV e intensidad de luz). Se construyó una plataforma que permite tener datos en tiempo real e integrar protocolos de acción basados en información de cultivos para un manejo efectivo. Se realizaron pruebas de laboratorio que demostraron el correcto funcionamiento del sistema. Se construyó un piloto a pequeña escala de un cultivo vertical para llevar a cabo pruebas a nivel de producción del cultivo, antes de escalarlo a sistemas de producción a pequeños escala.

Como trabajo futuro, se propone la construcción de sistemas inteligentes a partir de la información suministrada por los sensores y del desarrollo de los cultivos, la construcción de sistemas inteligentes (basados en aprendizaje de máquinas) de gestión autónoma basados en Inteligencia Artificial.

8. Referencias Bibliográficas

- [1] J. Beltrano and D. O. Gimenez, *Cultivo en hidroponía*. Editorial de la Universidad de La Plata, 2015.
- [2] D. S. Maria José Palma Lampreia, "Smart cities and urban areas - Aquaponics as innovative urban agriculture," *Urban For. Urban Green.*, vol. 20, pp. 402–406, Dec. 2016, doi: 10.1016/j.ufug.2016.10.004.
- [3] J. Birkby, "Vertical Farming," *ATTRA Sustain. Agric.*, pp. 1–12, 2016.
- [4] T. Changmai, S. Gertphol, and P. Chulak, "Smart Hydroponic Lettuce Farm using Internet of Things," in *2018 10th International Conference on Knowledge and Smart Technology: Cybernetics in the Next Decades, KST 2018*, Aug. 2018, pp. 231–236, doi: 10.1109/KST.2018.8426141.
- [5] B. A. Sheikh, "Hydroponics: key to sustain agriculture in water stressed and urban environment," *Pak. J. Agric., Agril. Eng., Vet. Sci.*, vol. 22 (2), pp. 53–57, 2006.
- [6] "URVEG Cultivo Hidropónico Urbano." <http://urveg.blogspot.com/p/hidroponia-metodos-hidroponicos.html> (accessed Sep. 14, 2020).
- [7] "Tipos de sistemas hidropónicos para cultivar | Generación Verde." <https://generacionverde.com/blog/hidroponia/tipos-de-sistemas-hidroponicos/> (accessed Sep. 14, 2020).
- [8] "Home Sentilo - Sentilo." <https://www.sentilo.io/wordpress/#> (accessed Sep. 14, 2020).
- [9] E. Touloupakis, L. Giannoudi, S. A. Piletsky, L. Guzzella, F. Pozzoni, and M. T.

- Giardi, "A multi-biosensor based on immobilized Photosystem II on screen-printed electrodes for the detection of herbicides in river water," in *Biosensors and Bioelectronics*, Apr. 2005, vol. 20, no. 10 SPEC. ISS., pp. 1984–1992, doi: 10.1016/j.bios.2004.08.035.
- [10] M. Zennaro and ICTP Trieste-Italy, "Introduccion to wireless sensor networks," 2012.
- [11] D. F. Morales Castaño, "Desarrollo de una red inalámbrica de biosensores para la detección de contaminantes tóxicos en aguas," Universidad de Antioquia, 2019.
- [12] L. Chamorro and E. Pietrosevoli, "Redes Inalámbricas para el desarrollo en América Latina y el Caribe," 2008. Accessed: Sep. 15, 2020. [Online]. Available: <https://www.flacsoandes.edu.ec/pt-br/agora/redes-inalambricas-para-el-desarrollo-en-america-latina-y-el-caribe>.
- [13] H. Ali, J. K. Soe, and S. R. Weller, "A real-time ambient air quality monitoring wireless sensor network for schools in smart cities," in *2015 IEEE 1st International Smart Cities Conference, ISC2, 2015*, pp. 1–6, doi: 10.1109/ISC2.2015.7366163.
- [14] R. Lara, D. Benítez, A. Caamaño, M. Zennaro, and J. L. Rojo-Álvarez, "On real-time performance evaluation of volcano-monitoring systems with wireless sensor networks," *IEEE Sens. J.*, vol. 15, no. 6, pp. 3514–3523, Jun. 2015, doi: 10.1109/JSEN.2015.2393713.
- [15] D. Villón Valdiviezo, "Diseño de una red de sensores inalámbrica para agricultura de precisión," Pontificia Universidad Católica del Perú, 2009.
- [16] N. Vikram, K. S. Harish, M. S. Nihaal, R. Umesh, A. Shetty, and A. Kumar, "A low cost home automation system using wi-fi based wireless sensor network incorporating internet of things (IoT)," in *2017 IEEE 7th International Advanced Computing Conference, IACC, Jul. 2017*, pp. 174–178, doi: 10.1109/IACC.2017.0048.
- [17] D. L. Mogrovejo Narváez, "Monitoreo de variables industriales basadas en protocolo de comunicación LoRa," Universidad Israel, Quito, 2019.
- [18] R. K. Kodali and S. R. Soratkal, "MQTT based home automation system using ESP8266," in *2016 IEEE Region 10 Humanitarian Technology Conference, R10-HTC, Apr. 2016*, pp. 1–5, doi: 10.1109/R10-HTC.2016.7906845.
- [19] M. I. Alipio, A. E. M. Dela Cruz, J. D. A. Doria, and R. M. S. Fruto, "A smart hydroponics farming system using exact inference in Bayesian network," in *2017 IEEE 6th Global Conference on Consumer Electronics, GCCE, Dec. 2017*, vol. 2017-January, pp. 1–5, doi: 10.1109/GCCE.2017.8229470.
- [20] D. S. Domingues, H. W. Takahashi, C. A. P. Camara, and S. L. Nixdorf, "Automated system developed to control pH and concentration of nutrient solution evaluated in hydroponic lettuce production," *Comput. Electron. Agric.*, vol. 84, pp. 53–61, Jun. 2012, doi: 10.1016/j.compag.2012.02.006.
- [21] "Espressif Systems - Wi-Fi and Bluetooth chipsets and solutions." <https://www.espressif.com/> (accessed Sep. 20, 2020).
- [22] "SHT3x (RH/T) - Digital Humidity Sensor | Sensirion." <https://www.sensirion.com/en/environmental-sensors/humidity-sensors/digital-humidity-sensors-for-various-applications/> (accessed Sep. 28, 2020).
- [23] S. Labs, "Si1145/46/47 Proximity/UV/Ambient light sensor," 2020. <https://www.silabs.com/documents/public/data-sheets/Si1145-46-47.pdf> (accessed Sep. 28, 2020).
- [24] "DS18B20." <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf> (accessed Sep. 28, 2020).
- [25] "Industrial_pH_electrode_SKU_FIT0348_-DFRobot." https://wiki.dfrobot.com/Industrial_pH_electrode_SKU_FIT0348_ (accessed Sep. 28, 2020).

- [26] "Analog_EC_Meter_SKU_DFR0300-DFRobot."
https://wiki.dfrobot.com/Analog_EC_Meter_SKU_DFR0300 (accessed Sep. 28, 2020).
- [27] "AD7792." https://www.analog.com/media/en/technical-documentation/data-sheets/AD7792_7793.pdf (accessed Sep. 28, 2020).
- [28] "E22-900M22S." <http://www.ebyte.com/en/product-view-news.aspx?id=437> (accessed Sep. 28, 2020).
- [29] "What is Arduino? | Arduino."
<https://www.arduino.cc/en/Guide/Introduction> (accessed Sep. 15, 2020).
- [30] "GitHub - espressif/arduino-esp32: Arduino core for the ESP32."
<https://github.com/espressif/arduino-esp32> (accessed Sep. 28, 2020).
- [31] B. de A. Ruiz Morón, "Adquisición de datos con microcontroladores para aplicaciones en el internet de las cosas," UNIVERSIDAD CARLOS III DE MADRID, 2017.

