

Máster Universitario en Internet de las Cosas:
Tecnologías Aplicadas Curso 2022-2023

Trabajo Fin de Máster

“Red IoT para la monitorización de la salud de los bosques de coníferas”

Manuel Álvarez Herrera

Tutor/es

José Antonio García Souto

Puerta de Toledo, Septiembre



[Incluir en el caso del interés de su publicación en el archivo abierto]

Esta obra se encuentra sujeta a la licencia Creative Commons

Reconocimiento – No Comercial – Sin Obra Derivada

Título: Red IoT para la monitorización de la salud de los bosques de coníferas

Autor: Manuel Álvarez Herrera

Director: José Antonio García Souto

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día ____ de ____ del 2023 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Desde este espacio quiero transmitir mi más sincero afecto, gratitud y cariño a todas las personas que me han dado fuerza y apoyo, las cuales menciono a continuación.

Quiero dar las gracias, en primer lugar, a mi tutor el Dr. José Antonio García Souto, por la forma en que me ha guiado, aconsejado, dedicándome su tiempo y facilitándome la realización de este Trabajo Fin de Máster. Gracias por toda la comprensión y empatía.

Quiero agradecer a la empresa Hydra Space Systems y mis compañeros de trabajo por las facilidades que me han proporcionado para realizar el presente proyecto.

Un especial reconocimiento a la profesión de maestro y a todos los profesionales que, con sus enseñanzas y ejemplo, me han ayudado, formándome como profesional y como ser humano.

Estoy muy agradecido con todos los compañeros del laboratorio de la UC3M que me han sabido guiar y siempre han estado dispuestos a tenderme una mano amiga. He de nombrar a Cindy Báez y Yuliay Moreno por su especial atención e inestimable apoyo.

Quiero mostrar mi gratitud a mis amigos, especial mención a Raúl y Vladimir por su gran amistad e inestimable ayuda al leer el presente TFM con infinita paciencia y aportarme su más sincera opinión, ¡un millón de gracias! Tampoco puedo olvidar a Melchior, Alejandro, Cristian, Alexander, Ulises, ... También quería mencionar a mis compañeros de la UC3M, sobre todo a mis amigos Juanjo, Guille, Marcos, Javier, Delia, Tomás, Marta... A todos gracias por ser apoyo en momentos difíciles, hemos superado mil obstáculos. Gracias por transmitirme vuestra fuerza, siempre espero contar con vosotros y sabed que siempre contareis conmigo.

He de dedicar este TFM a mi prima Amanda, a mis tíos María Cruz, Amalia y Vicente, a mis abuelos y, en especial, a mis padres que me han dedicado todo su cariño, tiempo, y esfuerzo con el objetivo de educarme y formarme lo mejor posible para afrontar la vida, por su paciencia, apoyo y amor incondicional que me guía siempre.

A todos, infinitas gracias.

Resumen

En este Trabajo Fin de Máster se ha implementado una red IoT para la monitorización de un bosque de coníferas, para lo cual se han utilizado diversos nodos que proporcionarán información del bosque a partir de magnitudes medioambientales y algunas magnitudes que indican el estado y la salud del ejemplar monitorizado.

Se describen los nodos utilizados para montar la red, comprobando su consumo de potencia y estudiando las diferentes formas de suministro de energía. Para la red de comunicación entre los diferentes nodos se utiliza tanto LoRa como NB-IoT. Se ha realizado un estudio de los rangos de cobertura de los nodos del sistema implementado. Todo esto se ha hecho con el fin de desarrollar una red eficiente y optimizada para el presente estudio.

Por último, una vez establecida la red a utilizar, se han comprobado distintos métodos de procesamiento de los datos obtenidos y se han usado distintas bases de datos de secuencias temporales para almacenar y graficar toda la información recabada a lo largo del tiempo de estudio.

Palabras clave: cambio climático, sistema de monitorización, magnitudes arbóreas, magnitudes climáticas, sensorización bosques, Arduino, Raspberry Pi, Dragino, nodos, red IoT, LoRa, NB-IoT, bases de datos, InfluxDB, The Things Network, Home Assistant.

Abstract

In this Master Thesis an IoT network will be implemented to monitor a coniferous forest, using different nodes that provide information about the forest from environmental magnitudes and some that indicate the state and health of the monitored specimen.

The nodes used to assemble the network will be indicated and their power consumption will be checked. Different energy supply forms are studied as well. Both LoRa and NB-IoT will be used for the communication network between the different nodes. A study of the individual nodes' coverage range inside the implemented system will be carried out. All this is done to try to develop the most efficient and optimal network for the present study.

Finally, once the network to be used has been established, different methods of processing the data obtained will be tested. Different time sequence database will be used to store and graph all the information collected throughout the study time.

Keywords: climate change, monitoring system, tree magnitudes, climate magnitudes, forest sensing, Edge Computing, Arduino, Raspberry Pi, Dragino, nodes, IoT network, LoRa, NB-IoT, databases, InfluxDB, The Things Network, Home Assistant.

Abreviaturas

Notación	Significado
BD	Bases de Datos
CNAF	Centro Nacional de Atribución de Frecuencias
C_2H_4	Etileno
CO_2	Dióxido de carbono
CSV	Valores Separados por Comas (Comma Separated Values)
CoAP	Constrained Application Protocol
GSM	Sistema Global para las comunicaciones Móviles (Global System for Mobile communications)
GW	GateWay
HTTP	Protocolo de Transferencia de HiperTexto (HyperText Transfer Protocol)
IoT	Internet de las Cosas (Internet of Things)
JSON	Notación de Objeto de JavaScript (JavaScript Object Notatio)
LoRa	Largo Alcance (Long Range)
LPWAN	Redes de Área Amplia y de Baja Potencia (Low Power Wide Area Network)
MQTT	Message Queuing Telemetry Transport
N	Nitrógeno
NB	Banda Estrecha (Narrow Band)
O_2	Oxígeno
ONU	Organización de las Naciones Unidas
PC	Ordenador personal (Personal Computer)
PCB	Placa de circuito impreso (Printed Circuit Board)
SIM	Módulo de Identificación de Abonado (Subscriber Identity Module)
SoC	Sistema en chip (System On a Chip)
SQL	Lenguaje de Consulta Estructurada (Structured Query Language)
TCP	Protocolo de Control de Transmisión (Transmission Control Protocol)
TFM	Trabajo de Fin de Máster
TTL	Lógica Transistor a Transistor (Transistor-Transistor Logic)
UDP	Protocolo de datagramas de usuario (User datagram protocol)
UE	Unión Europea
URL	Localizador de Recursos Uniforme (Uniform Resource Locator)
μC	Microcontrolador
μP	Microprocesador

X

Índice general

1.	Introducción	1
1.1.	Motivación	1
1.2.	Objetivos	2
1.3.	Impacto	3
1.4.	Fases de desarrollo y organización del trabajo	3
1.5.	Organización de la memoria	6
2.	Estado del arte	7
2.1.	Coníferas	7
2.2.	Cambio climático	10
2.3.	Sistemas de monitorización	11
2.4.	Redes de comunicación.....	11
2.5.	Bases de datos.....	13
2.6.	Acciones para la sostenibilidad del bosque	14
3.	Nodos sensores	16
3.1.	Magnitudes a medir y sensores utilizados	16
3.2.	Implementación de los nodos	19
3.2.1.	Primer tipo de nodo sensor implementado en (3).....	19
3.2.2.	Segundo tipo de nodo sensor implementado en (3).....	22
3.2.3.	Implementación del tercer tipo de nodo sensor.....	23
3.3.	Integración en el entorno.....	23
3.3.1.	Integración del primer nodo sensor.....	23
3.3.2.	Integración del segundo nodo sensor	24
3.3.3.	Integración del tercer nodo sensor	24
4.	Transmisión y recepción de los datos	26
4.1.	Primer nodo.....	26
4.2.	Segundo nodo	31
4.2.1.	NBSN95.....	31
4.2.2.	LSN50.....	34
4.3.	Tercer nodo	38
5.	Procesado y almacenamiento de datos	40
5.1.	Primer nodo.....	40
5.2.	Segundo nodo	43
5.3.	Tercer nodo	45

6.	Representación gráfica y estudio de las mediciones	48
6.1.	Gráficas, tratamiento de la información y alertas	48
6.1.1.	InfluxDB	48
6.1.2.	The Things Board.....	56
6.1.3.	Bases de datos meteorológicas.....	58
6.2.	Resultados	63
6.2.1.	Magnitudes propias del árbol	64
6.2.2.	Magnitudes meteorológicas.....	70
7.	Conclusiones y trabajo futuro	80
7.1.	Conclusiones.....	80
7.2.	Trabajo futuro	81
	Bibliografía	83
	Apéndice A	88
	Temperatura estomática.....	88
	Humedad de la hoja	89
	Diámetro del tallo, pecíolo y tronco	90
	Tamaño de fruto.....	91
	Sensores de Humedad	91
	Flujo de savia.....	92
	Oquedad del árbol.....	93
	Presión del jugo celular sobre la hoja	93
	Oxígeno en el suelo	94
	Oxígeno en la hoja.....	95
	Potencial hidráulico.....	95
	Dióxido de carbono en la hoja	96
	Dióxido de carbono en el suelo	97
	NPK.....	98
	pH	99
	Humedad del suelo	100
	Estación meteorológica.....	101
	Apéndice B	102
	Esquema electrónico de las conexiones del primer tipo de nodo implementado	102
	Esquema electrónico de las conexiones del segundo tipo de nodo implementado	103
	Apéndice C.....	107
	Cálculo rango de cobertura primer nodo.....	107
	Cálculo rango de cobertura segundo nodo.....	107
	Apéndice D	108

ping.c.....	108
pong.c.....	110
Apéndice E.....	112
transmit_implicit.c	112
receive_implicit.c	114
Apéndice F.....	116
sensores.c.....	116
main.c.....	132
Apéndice G	145
Apéndice H	147
Apéndice I.....	150

Índice de figuras

Figura 2.1.1 Fósil de Cordaites lungatus	7
Figura 2.1.2 Hyperion, la conífera más alta del mundo	8
Figura 2.1.3 Debilitamiento de las coníferas tras filomena	9
Figura 2.4.1 Resumen comparativo de las características de las tecnologías Sigfox, LoRa y NB-IoT	12
Figura 2.5.1 Máquina tabuladora o censadora	13
Figura 2.5.2 Tarjeta perforada	13
Figura 3.2.1 Esquema conceptual primer nodo	19
Figura 3.2.2 Esquema de conexión del primer nodo	20
Figura 3.2.3 Final del archivo 08_2023.json	21
Figura 3.2.4 Principio del archivo 09_2023.json	21
Figura 3.2.5 Modos de funcionamiento creados por el fabricante	22
Figura 3.2.6 Modos de funcionamiento diseñados para la creación del nodo	22
Figura 4.1.1 Módulo LoRa Ra-02 SX1278	26
Figura 4.1.2 Antena 433 MHz y 3 dBi de Dolla Tek	26
Figura 4.1.3 Esquema conceptual del primer sistema implementado	27
Figura 4.1.4 Esquema conceptual del primer sistema implementado	28
Figura 4.1.5 Esquema conceptual del primer sistema implementado	29
Figura 4.1.6 Ejecución del código ping.c en el nodo sensor	30
Figura 4.1.7 Ejecución del código pong.c en el nodo central	30
Figura 4.1.8 Ejecución del código transmit_implicit.c en el nodo sensor	30
Figura 4.1.9 Ejecución del código recieve_implicit.c en el nodo sensor	30
Figura 4.2.1 Funcionamiento del NBSN95	31
Figura 4.2.2 Conexión de la NBSN95 al PC a través del módulo TTL	32
<i>Figura 4.2.3 Conexión de la NBSN95 al PC a través del módulo TTL</i>	34
Figura 4.2.4 2JW0315-868-C675B	34
<i>Figura 4.2.5 Características propias para el rango de frecuencias requerido del CNAF</i>	35
<i>Figura 4.2.6 Frecuencias utilizadas por el GW</i>	35
<i>Figura 4.2.7 Potencia radiada aparente para las frecuencias utilizadas</i>	36
<i>Figura 4.2.8 Potencias máximas transmitidas por el GW</i>	36
Figura 4.3.1 Pantalla de la estación meteorológica WS3500	39
Figura 4.3.2 Configuración para el “Home Assistant”	39
Figura 5.1.1 Bucket utilizado en la BD InfluxDB	40
Figura 5.1.2 Token que enlaza el bucket junto con los permisos	41
Figura 5.1.3 Parte de la toma de datos que recoge InfluxDB Cloud	43

Figura 5.2.1 Datos recibidos de la estación meteorológica y GPS 1	44
Figura 5.2.2 Datos recibidos de la estación meteorológica y GPS 2	44
Figura 5.2.3 Datos recibidos de la estación meteorológica y GPS 3	44
Figura 5.3.1 Recepción de los datos en la BD Ecowitt	45
Figura 5.3.2 Recepción de los datos en la BD Weather Underground.....	45
Figura 5.3.3 Recepción de los datos en la BD Weathercloud.....	46
Figura 5.3.4 Integrar estación meteorológica en Home Assistant.....	46
Figura 5.3.5 Datos a introducir en la aplicación para la integración.....	47
Figura 5.3.6 Recepción de los datos en la BD Home Assistant	47
Figura 6.1.1 Explorador de datos	48
Figura 6.1.2 Datos de la humedad de las hojas.....	49
Figura 6.1.3 Dashboard de los sensores de hoja.....	49
Figura 6.1.4 Dashboard de todos los sensores.....	49
Figura 6.1.5 Script creado para mostrar la humedad de hoja	50
Figura 6.1.6 Check de la magnitud temperatura ambiente con un WARN y un CRIT.....	51
Figura 6.1.7 Creación una aplicación para Slack	51
Figura 6.1.8 Elección de opción “From Scratch” para crear la aplicación.....	52
Figura 6.1.9 Elección de nombre a la aplicación y asignación de workspace	52
Figura 6.1.10 Creación de un webhooks	53
Figura 6.1.11 Creación de un canal	53
Figura 6.1.12 Asignación del webhook al workspace	54
Figura 6.1.13 Asignación del canal al canal de Slack	54
Figura 6.1.14 Con el webhook creado se copia el enlace	55
Figura 6.1.15 Creación del Notification Endpoint	55
Figura 6.1.16 Datos de la temperatura ambiental recopilada por el segundo nodo.....	56
Figura 6.1.17 Datos de la humedad ambiental recopilada por el segundo nodo	56
Figura 6.1.18 Panel para generar las alertas para la temperatura para el segundo nodo	57
Figura 6.1.19 Algunos de los datos exportados de la BD The Things Boards.....	57
Figura 6.1.20 Gráfica de la temperatura en Ecowitt	58
Figura 6.1.21 Creación de la alerta en Ecowitt	58
Figura 6.1.22 Notificación de Ecowitt al saltar la alerta.....	59
Figura 6.1.23 Mapa de temperatura en España con Ecowitt.....	60
Figura 6.1.24 Informe semanal del tercer nodo en Weather Underground.....	60
Figura 6.1.25 Temperatura y punto de rocío en Weather Underground	61
Figura 6.1.26 Mapa de temperatura en España con Weather Underground	61
Figura 6.1.27 Archivo CSV generado del estudio en Weathercloud	61
Figura 6.1.28 Informe semanal del tercer nodo en Weathercloud	62

Figura 6.1.29 Sistema de elección de magnitudes a graficar proporcionada por Weathercloud	62
Figura 6.1.30 Temperatura interior y exterior del tercer nodo en Weathercloud	63
Figura 6.1.31 Mapa de temperatura en España con Weathercloud.....	63
Figura 6.2.1 Evolución de la humedad de las hojas	64
Figura 6.2.2 Datos de la humedad de hoja del pino durante el primer periodo de medición...	65
Figura 6.2.3 Datos de la humedad de hoja del pino durante un día del primer periodo	65
Figura 6.2.4 Datos de la humedad de hoja del pino durante todo el segundo periodo de medición	65
Figura 6.2.5 Datos de la humedad de hoja del pino un día del segundo periodo	65
Figura 6.2.6 Datos del <i>O₂</i> de hoja del pino durante todo el primer periodo de medición	66
Figura 6.2.7 Datos del <i>O₂</i> de hoja del pino durante un día del primer periodo	66
Figura 6.2.8 Datos del <i>O₂</i> humedad de hoja del pino durante todo el segundo periodo de medición	66
Figura 6.2.9 Datos del <i>O₂</i> humedad de hoja del pino durante un día del segundo periodo.....	66
Figura 6.2.10 Evolución del <i>CO₂</i> del aire en un día en una ciudad y en una zona rural.....	67
Figura 6.2.11 Evolución del <i>CO₂</i> del pino durante todo el primer periodo de medición.....	67
Figura 6.2.12 Evolución del <i>CO₂</i> del pino durante un día del primer periodo	68
Figura 6.2.13 Evolución del <i>CO₂</i> del pino durante todo el segundo periodo de medición	68
Figura 6.2.14 Evolución del <i>CO₂</i> del pino durante un día del segundo periodo	68
Figura 6.2.15 Evolución del pH del suelo en una zona controlada y en otra quemada	69
<i>Figura 6.2.16 Evolución del pH del pino durante todo el primer periodo de medición</i>	69
Figura 6.2.17 Evolución del pH del pino durante un día del primer periodo.....	69
Figura 6.2.18 Evolución del pH del pino durante todo el segundo periodo de medición.....	70
Figura 6.2.19 Evolución del pH del pino durante un día del segundo periodo	70
Figura 6.2.20 Datos relacionados con la temperatura y la humedad de Meteosierra	71
Figura 6.2.21 Datos relacionados con la temperatura exterior del tercer tipo de nodo representados en Ecowitt	71
Figura 6.2.22 Datos relacionados con la temperatura interior del tercer tipo de nodo representados en Ecowitt	71
Figura 6.2.23 Datos relacionados con la temperatura del tercer tipo de nodo representados en Weather Underground	72
Figura 6.2.24 Datos relacionados con la temperatura interior del tercer tipo de nodo representados en Weathercloud.....	72
Figura 6.2.25 Datos relacionados con la humedad exterior del tercer tipo de nodo representados en Ecowitt	73
Figura 6.2.26 Datos relacionados con la humedad interior del tercer tipo de nodo representados en Ecowitt	73
Figura 6.2.27 Datos relacionados con la humedad del tercer tipo de nodo representados en Weathercloud.....	73

Figura 6.2.28 Datos relacionados con la dirección y velocidad del viento de Meteosierra.....	74
Figura 6.2.29 Datos relacionados con la dirección del viento del tercer tipo de nodo representados en Ecowitt	74
Figura 6.2.30 Datos relacionados con la velocidad del viento del tercer tipo de nodo representados en Ecowitt	74
Figura 6.2.31 Datos relacionados con la dirección del viento del tercer tipo de nodo representados en Weather Underground.....	74
Figura 6.2.32 Datos relacionados con la velocidad del viento del tercer tipo de nodo representados en Weather Underground.....	75
Figura 6.2.33 Datos relacionados con la dirección del viento del tercer tipo de nodo representados en Weathercloud.....	75
Figura 6.2.34 Datos relacionados con la velocidad del viento del tercer tipo de nodo representados en Weathercloud.....	75
Figura 6.2.35 Datos relacionados con la radiación solar e índice ultravioleta de Meteosierra..	76
Figura 6.2.36 Datos relacionados con la radiación solar e índice ultravioleta del tercer tipo de nodo representados en Ecowitt	76
Figura 6.2.37 Datos relacionados con la radiación solar del tercer tipo de nodo representados en Weather Underground	76
Figura 6.2.38 Datos relacionados con el índice ultravioleta del tercer tipo de nodo representados en Weather Underground.....	76
Figura 6.2.39 Datos relacionados con la radiación solar e índice ultravioleta del tercer tipo de nodo representados en Weathercloud	77
Figura 6.2.40 Datos relacionados con la lluvia de Meteosierra	77
Figura 6.2.41 Datos relacionados con la lluvia del tercer tipo de nodo representados en Ecowitt	77
Figura 6.2.42 Datos relacionados con la lluvia del tercer tipo de nodo representados en Weather Underground	78
Figura 6.2.43 Datos relacionados con la lluvia del tercer tipo de nodo representados en Weathercloud	78
Figura 6.2.44 Datos relacionados con la presión atmosférica del tercer tipo de nodo representados en Ecowitt	79
Figura 6.2.45 Datos relacionados con la presión atmosférica del tercer tipo de nodo representados en Weather Underground.....	79
Figura 6.2.46 Datos relacionados con la presión atmosférica del tercer tipo de nodo representados en Weathercloud.....	79

Índice de tablas

Tabla 1.4.1 Diagrama de Gantt	5
Tabla 3.1.1 Selección de los sensores a utilizar.....	18
Tabla 4.1.1 Pines para la integración del módulo Ra-02 en la Raspberry Pi 4	27
Tabla 4.1.2 Datos de las antenas y módulos de comunicación para ver el balance de enlace del primer nodo.....	28
Tabla 4.2.1 Datos de las antenas y módulos de comunicación para ver el balance de enlace del segundo nodo	38

1. Introducción

En este apartado se abordarán los conceptos fundamentales, se expondrá el motivo que ha guiado la creación del presente Trabajo Fin de Máster (TFM), junto con los objetivos que se aspiran conseguir.

Se explorarán algunas de las utilidades que emanen del presente proyecto. También, se mostrará la organización temporal delineada en el diagrama de Gantt para realizar este trabajo. Se concluirá con la organización de la memoria.

1.1. Motivación

Con la llegada, a mediados del siglo XVIII, de la primera revolución industrial, se aceleró sin límite la sobreexplotación de los recursos naturales, la producción de gases de efecto invernadero y de productos contaminantes. Estos han sido vertidos sin control y sin depurar a la atmósfera, a la tierra, a los ríos, a los acuíferos, al mar..., hecho que ha ido en aumento de forma exponencial y que aún sigue casi sin freno ni límites.

La comunidad científica, desde casi los principios de la revolución industrial, y las organizaciones ecologistas, desde mediados del siglo XX, llevan estudiando los efectos que tienen la sobreexplotación de los recursos naturales, la superproducción de los gases de efecto invernadero y el inmenso vertido incontrolado de productos contaminantes, sobre el clima, la naturaleza y el ser humano.

Una de las principales conclusiones aceptadas es que hemos conseguido cambiar el clima en un tiempo récord, con consecuencias dañinas y nocivas, como las que se están observando a través de las catástrofes relacionadas directamente con el cambio climático. Por lo tanto, en la época actual, el cambio climático es una de las mayores preocupaciones a las que se enfrenta la humanidad. Se estima que a corto y medio plazo producirá grandes y graves cambios en los patrones climáticos, generando condiciones medioambientales perjudiciales, tanto para la fauna, la flora como para la sociedad humana, debido a que las grandes variaciones de las temperaturas afectan negativamente a la naturaleza y, por ende, al sistema.

Aunque el ser humano lleva modificando el ecosistema en el que habitamos desde que dejó de ser nómada, este hecho se ha agravado con la era industrial, donde se produce sin descanso y se desechara lo producido a gran velocidad. El medioambiente por sí solo no tiene la capacidad de regenerarse a la velocidad que el ser humano lo destruye (1)(2).

El mayor problema que hay que afrontar en estos momentos consiste en proteger la salud del planeta y de todos los seres vivos que lo habitan. Se trata, por tanto, de buscar todos los medios humanos, tecnológicos y económicos para revertir el daño causado.

Lo que motiva este trabajo es alcanzar un adecuado desarrollo y estado de salud de los bosques, ya que son un vector necesario para la solución de este problema. Se pretende hacer uso de los medios y conocimientos que el Internet de las Cosas (IoT) puede aportar en este campo, como es informar del estado de los bosques para mejorar el aprovechamiento de los escasos recursos hídricos, de los abonos, de los pesticidas... y crear alertas para avisar de situaciones anómalas o extremas.

INTRODUCCIÓN

Este proyecto está motivado por la propuesta *MANAGE4FUTURE* dentro de la convocatoria de Proyectos de Transición Ecológica y de Transición Digital correspondiente al Plan de Recuperación, Transformación y Resiliencia. A su vez, la Unión Europea (UE) está interesada en luchar contra el calentamiento global, esto pasa por la protección de todos los bosques al ser grandes almacenes de carbono, suministradores de riqueza a la economía y a la sociedad. Este proyecto pretende ser parte activa de la lucha contra el cambio climático, aportando su pequeño grano de arena.

1.2. Objetivos

El objetivo principal de este proyecto es estudiar la evolución del cambio climático a través de la monitorización de bosques de coníferas y otros entornos. En este estudio se pretende observar cómo varían múltiples magnitudes que pueden afectar al desarrollo y la salud de estos árboles.

Para ello se tendrán en cuenta las magnitudes que pueden afectar directa o indirectamente a las coníferas y al entorno en el que se encuentran, y se implementará un sistema de IoT que monitorice el estado del bosque de coníferas, permita observar su evolución y facilite los datos y gráficas a los diferentes expertos.

Los objetivos de este proyecto son complejos, amplios y ambiciosos, lo que hace que sean inabarcables en un solo TFM. Debido a esta razón, el presente trabajo fin de máster se considera complementario de otro realizado en paralelo. Las diferencias claves entre ellos son las siguientes:

- El trabajo realizado previamente tiene como objetivo el estudio, desarrollo e implementación de los distintos nodos de medición que serán parte integral del sistema de monitorización, esto se hizo para bosques de quercíneas. (3)
- El presente estudio pretende ser una continuación al tratar de implementar un sistema completo de IoT para la monitorización de una amplia región. Para ello, se creará un nodo central que se encargará de comunicarse con los distintos nodos sensores y con las bases de datos con el fin de recopilar, procesar, almacenar y graficar todos los datos de los diversos nodos.

Por consiguiente, los propósitos de ambos trabajos van encaminados a un objetivo compartido, que es la monitorización, protección y mejora de salud de nuestros entornos naturales. Sin embargo, cada proyecto posee un enfoque claramente diferenciado del otro, pero complementario. Una diferencia adicional viene dada por el tipo de bosques que es objeto de estudio en cada caso. Así, en el presente TFM el estudio se centra en coníferas.

El trabajo está estructurado a través de los siguientes subobjetivos:

- Estudiar e investigar las distintas formas de desarrollar el presente proyecto.
- Plantear los nodos centrales que se encargarán de aunar la información recopilada por los sensores de medida.
- Estudiar e implementar técnicas de comunicación a largas distancias entre los nodos sensores y los nodos centrales.
- Diseñar las funcionalidades que van a poseer los nodos centrales.
- Estudiar las distintas bases de datos (BD).

- Almacenar y, si es necesario, procesar los datos en las BD.
- Presentar gráficamente los datos procesados en las BD para realizar un estudio.
- Extraer conclusiones.

1.3. Impacto

Este trabajo está enfocado a la monitorización de entornos naturales, con el objetivo de hacer un seguimiento del cambio climático y su impacto en los mismos. Adicionalmente, este TFM puede tener utilidad en otros ámbitos:

La agricultura: Al recoger datos medioambientales y estar pensado para el estudio de la flora, se podría aplicar a la agricultura y la jardinería. De esta forma se contaría con entornos más controlados en donde sería posible mejorar y rentabilizar los escasos recursos hídricos, optimizar el uso de abonos y reducir el impacto de pesticidas.

La botánica: La utilidad en este caso proviene de que la información proporcionada corresponde a una escala de tiempo prolongada y sobre distintos entornos y ambientes en estudio. Se podrían realizar ensayos para seleccionar los ejemplares más adecuados por su adaptación a cada tipo de terreno y que, por tanto, requieren menos cuidados. De esta forma, podría tener un impacto medioambiental y también sobre la biodiversidad.

Las ciencias medioambientales y la ecología: Este trabajo puede ser de utilidad para la preservación del medioambiente y prevención de desastres climáticos, incendios, enfermedades de las plantas... al disponer de sensores que proporcionan información a tiempo real. Con esta tecnología se podrían utilizar alarmas para casos extremos, y alimentar de datos a sistemas de aprendizaje máquina para predecir posibles catástrofes ambientales.

El Big Data: Con todos los datos que se pueden obtener con la explotación del sistema implementado, sería posible realizar análisis más complejos y rigurosos haciendo uso de técnicas de *machine learning*, inteligencia artificial o estrategias propias del *Big Data* para lograr recabar información útil según la demanden los expertos.

El IoT: Este TFM es un sistema de monitorización basado en redes de internet de las cosas. Al utilizar distintos sensores, nodos, sistemas de comunicación, BD, etc., puede servir para futuros proyectos y fomentar nuevas ideas de investigación.

1.4. Fases de desarrollo y organización del trabajo

Como el proyecto que se quiere llevar a cabo tiene un alto nivel de complejidad, es necesario dividir el trabajo en diversas fases de desarrollo. Haciendo uso de esta estrategia, se logra simplificar, ordenar y estructurar las tareas. A continuación, se muestran las diferentes fases en las que se desarrollará el TFM:

- **Estudio inicial:** Para la mejor comprensión del presente trabajo, se hace necesario la realización de un breve resumen de lo desarrollado en el anterior proyecto. Además, se hace imprescindible realizar un estudio pormenorizado de los distintos elementos que van a intervenir en el desarrollo y evolución del TFM.

INTRODUCCIÓN

- **Investigación de distintas tecnologías de comunicación:** Para conseguir una correcta transmisión de datos dentro de la red, se hará imprescindible una elección correcta de las tecnologías de comunicación a utilizar.
- **Creación de los nodos centrales:** Una vez desarrollados los distintos nodos de medida, se hace necesario implementar los distintos nodos centrales que se encarguen de aunar los datos recopilados. Cada uno de ellos puede tener distintas funcionalidades para llevar a cabo su cometido.
- **Funcionalidades de los nodos centrales:** Cada nodo tendrá distintas tareas asignadas según su cometido. Algunos procesarán los datos o los almacenarán de forma temporal, pero todos ellos transmitirán la información a distintas BD.
- **Bases de datos:** Se estudiarán distintas BD existentes y se seleccionarán las que más se ajusten a las necesidades propias del presente TFM.
- **Tratamiento de datos:** Se hará uso de varias de las diferentes funcionalidades, que tienen incorporadas las distintas BD, utilizadas para tratar y graficar toda la información recopilada.
- **Redacción de la memoria:** Para finalizar el proyecto, se procede a la elaboración de la presente memoria para reflejar las distintas tareas realizadas junto con los datos obtenidos.

Previamente a la elaboración de este TFM se realizó una estimación del tiempo y tareas necesarias para completar el proyecto. Todo esto viene recogido en el siguiente diagrama de Gantt:



Diagrama de Gantt

Red IoT para la monitorización de la salud de los bosques de coníferas

UC3M										
Autor: Manuel Álvarez Herrera Tutor: Jose Antonio García Souto										
Fecha de inicio del proyecto:		02/07/2022								
Meses	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto		
Descripción del hito	Inicio	Días								
Estudio inicial	20/01/2022	40								
Investigación de distintas tecnologías de comunicación	20/01/2022	20								
Creación de los nodos centrales	16/03/2022	55								
Funcionalidades de los nodos centrales	18/03/2022	70								
Bases de datos	01/03/2022	90								
Tratamiento de datos	01/05/2022	75								
Redacción de la memoria	01/01/2022	184								

Tabla 1.4.1 Diagrama de Gantt

1.5. Organización de la memoria

El contenido de la actual memoria se va a segmentar en los apartados que se detallan a continuación:

Apartado 2

Estado del arte: En este apartado se expondrá una breve historia de las coníferas, su localización y sus múltiples valores. También se expondrán distintos elementos previos para analizar los sistemas de monitorización en IoT.

Apartado 3

Nodos implementados: En este apartado se expondrá los distintos nodos implementados en el TFM anterior, junto con las magnitudes útiles para la medición del sistema, así como la forma de integrar el nodo en el entorno de medición. Además, se explican los nuevos nodos centrales que se han implementado en este proyecto.

Apartado 4

Transmisión y recepción: En este apartado se expondrá el estudio de las distintas técnicas de comunicación a largas distancias, como la elección de estas para transmitir los datos de cada uno de los tipos de nodos de medida a sus correspondientes nodos centrales.

Apartado 5

Procesado y almacenaje en las bases de datos: En este apartado se expondrá el proceso de guardar los datos de manera ordenada en las BD, para su futuro tratamiento y posterior consulta.

Apartado 6

Representación gráfica y estudio de las mediciones: En este apartado se expondrá la forma de representar los datos con las funcionalidades presentadas por las BD, para obtener información y realizar el posterior análisis y comparativa.

Apartado 7

Conclusiones y trabajo futuro: En este apartado se expondrán los resultados obtenidos tras el desarrollo e implementación del presente TFM. También, se explicarán las distintas aplicaciones, modificaciones y mejoras que se podrían realizar en un futuro.

Para finalizar, el documento incluirá la bibliografía y diversos apéndices que completan los contenidos de esta memoria.

2. Estado del arte

Es innegable que el mundo está cambiando, los gases de efecto invernadero, los distintos vertidos contaminantes junto con el incesante cambio climático está afectando a la salud de todos los seres vivos y de nuestros entornos forestales. Se puede observar que cada año hay una mayor cantidad de incendios, junto con talas indiscriminadas y contaminación de los entornos naturales.

Se hace, por tanto, fundamental conseguir preservar estos entornos naturales, para ello se ha ideado este proyecto el cual intenta aportar una de las múltiples acciones que se pueden realizar para favorecer la conservación del medio ambiente. Con este trabajo se pretende realizar un sistema de observación de los efectos del cambio climático para monitorizar el estado de salud de los tan importantes bosques de coníferas.

Para conseguir una mejor comprensión de la presente memoria, se procede a exponer los conceptos previos y a observar el estado actual de los sistemas de monitorización para poder desarrollar el sistema deseado de forma adecuada.

2.1. Coníferas

Las coníferas son árboles y arbustos que se cree que derivan de las cordaites, datan de finales del carbonífero, hace unos 300 millones de años aproximadamente. Este tipo de árboles son los que más proliferaron tras la extinción masiva del Pérmico-Triásico hace 252 millones de años (4).



Figura 2.1.1 Fósil de *Cordaites lungatus*

Las coníferas se identifican por ser plantas leñosas que secretan resina para protegerse de insectos, hongos y heridas. La mayoría de ellas son árboles con crecimiento

ESTADO DEL ARTE

monopodial, es decir, un solo tronco recto con ramas laterales. Se caracterizan por la forma cónica de su ramaje y frutos. Algunos ejemplos de coníferas serían cedros, abetos, cipreses, enebros, kauri, alerces, pinos, secuoyas y tejos.

En su gran mayoría son árboles de hoja perenne, debido a que la hoja tiene una vida entre dos y cuarenta años, esto facilita la integración de los sensores, la forma de la hoja es de escama o aguja. En diversas especies de coníferas, las hojas se encuentran dispuestas en forma de espiral con las bases de las hojas torcidas para presentarlas en un plano, para así captar el máximo de luz.

Las hojas en altas latitudes o bajo la umbría de un dosel forestal es de color verde oscuro, para poder absorber de forma más eficiente la poca luz solar que les llega. En cambio, las coníferas situadas en zonas cálidas y con gran cantidad de luz solar, suelen poseer hojas de un color verde claro o verde amarillento.

Otra característica de las coníferas es que son gimnospermas porque tienen sus semillas al descubierto. Las flores de las coníferas no tienen pétalos y pueden ser macho o hembra, según la especie pueden ser monoicas (flores macho y hembra se encuentran en la misma planta) o dioicas (solo se encuentran flores macho o solamente flores hembra en la misma planta). La fertilización depende de que el viento sople de tal forma que el polen producido por las flores masculinas llegue a las flores femeninas y de esta forma desarrolle la semilla en el estróbilo, que es el cono protector (5).

La altura de las coníferas varía entre treinta centímetros y poco más de cien metros, en ejemplares adultos, siendo este taxón el que tiene el honor de contar entre sus ejemplares con los árboles más longevos, ejemplos de ellos son: el más antiguo Matusalén que cuenta con más de 4800 años de vida (6) y el más alto Hyperion que cuenta con una altura de 115,55 metros (7) del mundo.



Figura 2.1.2 Hyperion, la conífera más alta del mundo (8)

Una de sus características más relevantes es que poseen una fácil adaptación a climas secos y fríos. En el hemisferio norte hay bosques boreales de coníferas llamados taigas, están compuestos por pinos, pináceas y abetos, tienen forma cónica, estrecha y sus ramas se inclinan hacia abajo para deshacerse más fácilmente de la nieve, también modifican su bioquímica para volverse resistentes a la congelación. En los lugares con mayor temperatura los bosques de coníferas están integrados por secuoyas, cedros y cipreses (9).

Las coníferas son el grupo más importante de las gimnospermas, destaca su gran importancia tanto ecológica como económica. Hay que recalcar que son el principal sumidero de carbono terrestre, por tanto, son capaces de absorber el CO_2 de la atmósfera y fijarlo para mantener los niveles idóneos. Es importante conocer sus múltiples utilidades, algunas de ellas se reflejan a continuación:

- **Madera:** Su madera es muy apreciada en diversos sectores como la construcción de muebles, carpintería, ebanistería, embalaje, silvicultura, papelera, fabricación de instrumentos musicales...
- **Resina:** Es muy apreciada como aislante, para quirófanos, salas estériles, laboratorios. También se utiliza en la industria farmacéutica, alimentaria, mecánica, en la construcción...
- **Semillas:** Los piñones son muy apreciados en el ámbito gastronómico y se utilizan en la industria alimentaria, farmacéutica...
- **Cáscaras y piña:** Con los restos se puede fabricar biomasa que es muypreciada en la industria energética.
- **Metabolitos secundarios:** En la actualidad se han creado nuevas técnicas para aprovechar la totalidad del árbol, se pueden extraer del serrín, de las astillas, entre otros. Estos metabolitos se pueden utilizar como adsorbente natural, también para la producción de fermentos del etanol, además son muy apreciados en la industria química y farmacéutica... (10)

Desde finales del último siglo, se ha descubierto que existen zonas boscosas, tanto en Europa como en América, con claros síntomas de debilitamiento. Esto provoca en los árboles la pérdida de fuerza, retracción de las raíces, decoloración y caída precoz de las hojas (defoliación) y, por último, el árbol se seca y muere, debido a que no tiene el suficiente vigor para soportar heladas, sequías o enfermedades. A esto se le denominó, en la década de los ochenta en Alemania, «Waldsterben» (11).



Figura 2.1.3 Debilitamiento de las coníferas tras filomena

ESTADO DEL ARTE

En los años ochenta, en quince países europeos, unas siete millones de hectáreas boscosas se vieron afectadas por este debilitamiento, así como en el norte de América.

El estudio realizado por varios expertos de este fenómeno ha achacado este debilitamiento a la contaminación y cambio climático, pero parece ser más complejo, ya que existen una gran cantidad de fenómenos que afectan a los bosques como son la lluvia ácida, dióxido de azufre, déficit de ozono, calentamiento global, óxido de nitrógeno, suelos contaminados, etc. Otros expertos son más prudentes y solicitan ampliar las investigaciones.

2.2. Cambio climático

En 1859 el físico irlandés John Tyndall descubrió que los gases como el CO₂, el metano y el vapor de agua, eran los creadores del efecto invernadero (12). Más tarde, en 1896, el físico-químico Svante Arrhenius, se basó en los estudios de Tyndall, para cuantificar las consecuencias en el clima de dicho efecto, acuñando el concepto de cambio climático y prediciendo la fecha en que ocurriría, la predicción difirió debido a que el ritmo de producción de estos gases han ido aumentando paulatinamente. (13)

El geólogo estadounidense Wallace Smith Broecker, en 1975, fue la primera persona en publicar sobre la instauración definitiva del cambio climático, sobre el calentamiento global del planeta en su conjunto y sobre sus graves consecuencias. (14)

El cambio climático implica la modificación de los patrones tanto de temperatura como de las características climáticas propias de una zona geográfica o del planeta en general. Estos cambios pueden ser tanto:

- Naturales: debido a los ritmos del clima, de la rotación de la tierra...
- Actividades del ser humano: como la quema de combustibles fósiles, la creación de gases y productos potencialmente nocivos para el medioambiente, que acaban en la atmósfera, ríos, mares, espacios naturales....

Estos cambios se están produciendo cada vez a velocidades más vertiginosas, provocando que la situación sea especialmente preocupante, sobre todo al tener en cuenta que estas se deben principalmente a la actividad humana.

A partir de la revolución industrial se empezó un modelo de producción masivo, haciendo uso de quema de combustible fósil para la obtención de energía. Esto provocó la liberación de gases de efecto invernadero como el metano o dióxido de carbono, que actúan como una especie de cúpula que envuelve el planeta y no deja que salga el calor, por tanto, provoca un aumento paulatino de las temperaturas denominado calentamiento global.

La desidia al no intervenir para evitar los diversos desastres medioambientales ocurridos, a lo largo de la historia, ha sido propiciada por la mentalidad de que los costes para evitarlos se calificaban como costes demasiado onerosos, estimulado por la idea ingenua de que la naturaleza sería capaz, por ella misma, de asimilar los productos nocivos que se generan de los procesos productivos. Debido a la rápida evolución del cambio climático, se hace más difícil la adaptación de toda la Naturaleza y, por ende, del ser humano a estas alteraciones del medio.

2.3. Sistemas de monitorización

La monitorización se encarga de vigilar de cerca el estado de un sistema en estudio haciendo uso de sensores, por ende, se trata de un sistema pasivo que se encarga de adquirir, tratar y almacenar datos relevantes del sistema estudiado. Estos datos son de gran importancia al aportar información relevante para el análisis e investigación del sistema. La monitorización, en consecuencia, está ligada tanto al análisis de datos como a los sistemas de control.

Para que el sistema de monitorización pueda interactuar de forma correcta con el entorno, se utilizan los dispositivos denominados sensores. Estos son capaces de captar los cambios provenientes del entorno, es decir, son transductores que se encargan de hacer interpretables las magnitudes físicas y/o químicas medidas. En la mayoría de los casos transforman estos datos en magnitudes eléctricas para que sean interpretables por ordenadores y otros sistemas electrónicos.

En 1724, Daniel Gabriel Fahrenheit, desarrolla lo que actualmente se conoce como el primer sensor de la historia. Este sensor se encarga de medir la temperatura y es conocido como termómetro, su diseño consta de un tubo de vidrio donde se aloja mercurio líquido tintado, la temperatura influye en la expansión de este metal líquido en el frasco. Posteriormente, en 1874, fue desarrollado por un equipo de científicos franceses una estación meteorológica que proporcionaba datos a grandes distancias. (15)

Actualmente, para el desarrollo de los sistemas de monitorización son prácticamente imprescindibles los sistemas embebidos. Estos consisten en sistemas electrónicos desarrollados para ejecutar tareas específicas, para ello se utilizan microprocesadores (μP) y/o microcontroladores (μC) (16) (17). Los sistemas embebidos pueden ser autónomos, pero, los sistemas de monitorización suelen estar integrados al ser parte de un sistema mayor.

En 1970 se desarrolla el primer sistema empotrado para los cálculos de vuelo del avión F-14 Tomcat (18). Estos dispositivos son esenciales en la actualidad en diversos ámbitos y campos, como dispositivos médicos, sistemas de comunicación y son parte fundamental en el IoT.

Se hacen uso de sistemas de control para conseguir que estos complejos sistemas de monitorización se les puedan añadir un sistema de actuación. De esta forma, logra propiciar el desarrollo del sistema en estudio y comprobar el buen funcionamiento del propio sistema de monitorización. Esto se puede realizar de forma predeterminada (lazo abierto) o a través del estudio de la evolución del sistema (lazo cerrado). En consecuencia, se tienen que coordinar de forma adecuada y deben de mantener una continua interacción con el entorno para poder dar respuesta ante los estímulos.

2.4. Redes de comunicación

Para la realización de sistemas de monitorización complejos, será necesario tener la capacidad de conectar miles o millones de dispositivos inalámbricos de forma eficiente. Para llevar a cabo esta comunicación inalámbrica se pueden utilizar distintas tecnologías y medios como radiofrecuencia, microondas, infrarrojos, luz, entre otras muchas.

ESTADO DEL ARTE

Gracias al descubrimiento de las ondas electromagnéticas en el siglo XIX por Heinrich Hertz, Guillermo Marconi desarrolló el primer sistema de comunicación inalámbrico haciendo uso de ondas electromagnéticas de radiofrecuencia para transmitir información entre dos puntos distantes. Desde ese momento ha cambiado completamente la forma de realizar las comunicaciones. (19)

Será importante para redes IoT tener sistemas escalables, de coste reducido, fácil integración e instalación y que no se necesite un gran mantenimiento. En este proyecto, al desarrollarse en un entorno forestal, será de gran utilidad redes que sean capaces de transmitir y recibir información a grandes distancias.

La comunicación inalámbrica actual difiere de la primigenia, actualmente se hacen uso de tecnologías que varían en función de la forma en la que se transmite, la velocidad de transferencia de datos, la latencia existente en la comunicación, alcance de la comunicación...

A parte de las tecnologías expuestas previamente, existen soluciones especializadas en el marco del IoT. Los grandes sistemas de IoT cuentan con una cantidad abrumadora de dispositivos, por ende, reciben un inmenso volumen de datos. Además, suelen ser sensores que precisan cierta autonomía, por lo cual, será necesario que tengan un bajo consumo a la hora de adquirir y transmitir la información recopilada. Muchas de las aplicaciones más frecuentes en IoT son en grandes entornos de estudio.

Para cumplir con los requisitos expuestos en el párrafo anterior, se hará uso de técnicas de comunicación LPWAN que están expresamente diseñadas para este menester. Algunos de los ejemplos más destacados son:

- **Sigfox:** Red global de larga distancia y bajo consumo diseñada expresamente para generar soluciones IoT. El inconveniente de esta red es que es de pago y, además, de pago por unidad, por tanto, no se utilizará para este proyecto al no ser la mejor opción.
- **NB-IoT:** tecnología de comunicación celular que utiliza las bandas de telefonía móvil consiguiendo un bajo consumo energético. Diseñada para soluciones IoT, consiguiendo una gran penetración en interiores. Esta será de utilidad para el proyecto, por tanto, se desarrollará más en apartados posteriores.
- **LoRa:** tecnología de comunicación inalámbrica de largo alcance y bajo consumo de potencia que posee una cobertura extraordinariamente extensa. Estas características lo hacen ideal para el desarrollo del presente proyecto, por tanto, se utilizará en una de las posibles soluciones implementadas.

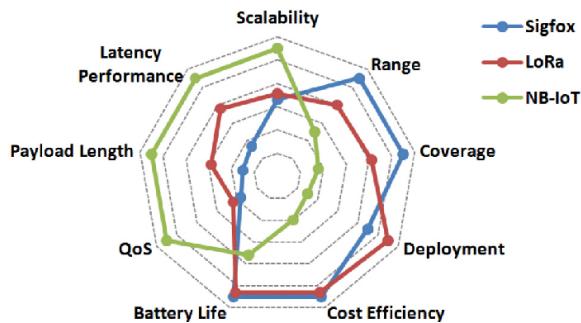


Figura 2.4.1 Resumen comparativo de las características de las tecnologías Sigfox, LoRa y NB-IoT (20)

Un pequeño resumen de las capacidades que tienen este tipo de tecnología LPWAN se muestra en la figura 2.4.1. Estas tecnologías son pilares para el desarrollo de la comunicación inalámbrica, eficiente y confiable para redes de IoT.

2.5. Bases de datos

El ser humano ha atesorado la información de la que dispone desde el origen de la escritura. Se podrían considerar a las bibliotecas y toda clase de registros como las primeras BD. Estas fundamentalmente se utilizaban para recopilar los datos de la cosecha y realizar censos de la población.

La primera BD mecanizada, fue creada en 1884 por Herman Hollerith, se denominó máquina tabuladora, esta se muestra en la figura 2.5.1. Esta máquina utiliza tarjetas perforadas, como la que se muestra en la figura 2.5.2, para la recolección de datos censales.



Figura 2.5.1 Máquina tabuladora o censadora (21)

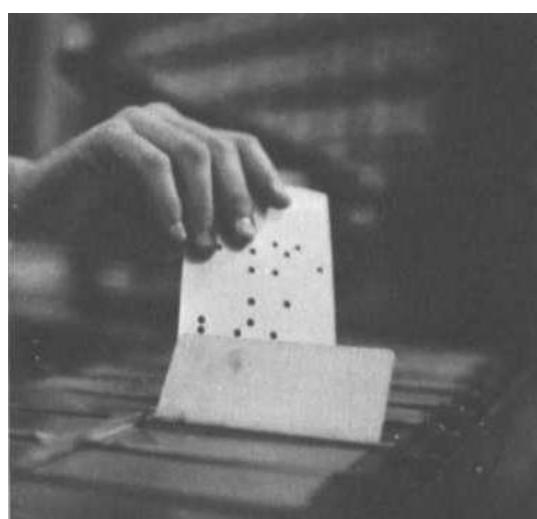


Figura 2.5.2 Tarjeta perforada (21)

La evolución de las BD está estrechamente relacionada con el desarrollo del HW y de las computadoras. En la década de 1950 se da origen a las cintas magnéticas, estas cintas eran capaces de almacenar hasta cientos de kilobytes de información, pero solamente de manera secuencial. Para el almacenamiento de estos datos se hacían uso de códigos para procesar la información recogida.

En la década de 1960 se crea el primer sistema de gestión de BD, este fue desarrollado por Charles W. Bachman. IBM, para no quedarse atrás, también desarrolló un sistema de gestión de BD: IMS. Ambos fueron los antecesores de las primeras BD de navegación.

Se utilizaron diversos lenguajes de programación para el almacenamiento y uso de las nuevas BD creadas. Sin embargo, no hubo un estándar fijo, esto lo llevó a cabo ANSI en la década de los setenta. Todo lo mencionado, junto con el descenso del precio de las computadoras y la mejora en velocidad y flexibilidad, produjo una popularización de las computadoras en las empresas.

Durante la década de los setenta, el científico e informático Edgar Frank Codd cansado de que no se pudieran buscar datos de forma sencilla en las BD primigenias, creó la teoría de BD relacionales. Basándose en Codd, Lawrence J. Ellison desarrolló el Relational Software System o, como es más conocido, *Oracle*.

Siguió la evolución natural de estas BD, en la década de los ochenta el incremento de ventas de BD relacionales, SQL se convirtió en el estándar de la industria al utilizar un sistema de tablas con forma matricial, que fue un claro competidor de las BD jerárquicas y de red.

En los noventa con la aparición de Internet, *World Wide Web*, aparecen herramientas como *Excel*, *Access*... esto hace que las BD se desarrollen poco a poco dando origen a las que conocemos actualmente.

En la actualidad hay una gran diversidad de BD, en función de sus características, utilidades, campo de estudio... Al ser el sistema a implementar en el presente proyecto una red IoT, se tendrá que utilizar una BD acorde a las necesidades de este, que sea capaz de tratar con un gran flujo de datos en distintos instantes de tiempo.

Las BD de series temporales son las más adecuadas para el propósito descrito previamente, se encargan de tratar datos que tienen marcas temporales, facilitando el procesamiento, el almacenamiento y la consulta de forma rápida y eficiente. Algunos de los ejemplos de este tipo de BD son *InfluxDB*, *Prometheus*, *Graphite*, *AWS Timestream*, *The Things Network*...

2.6. Acciones para la sostenibilidad del bosque

Los entornos naturales se ven cada vez más amenazados por el cambio climático, la deforestación y los cambios socioculturales. Esto provoca un abandono de estas zonas. El gobierno de España cada vez más se preocupa por el bienestar de sus bosques, principalmente para la mejora de la salud medioambiental y fomentar la conservación de estos para tener una mejora en la industria forestal. Algunos de estos ejemplos son

ESTADO DEL ARTE

los “Criterios e Indicadores de gestión forestal sostenible en los bosques españoles” (22) y la “Estrategia Forestal Española Horizonte 2050” (23).

A nivel internacional hay otras iniciativas como el “Pacto Verde Europeo” (24) (25), el “Acuerdo de París” (26), el “Objetivo de Desarrollo Sostenible de la Agenda 2030” (27), “Convenio de Diversidad Biológica” (28) y el “Protocolo de Kioto” (29). En todos estos se promueve la lucha contra el cambio climático, haciendo uso de los bosques para paliar sus efectos.

Hay diversos proyectos de IoT para la monitorización de grandes espacios naturales. Todo esto para mejorar la salud de los bosques o de entornos naturales terrestres. Algunos de estos múltiples ejemplos son:

- "Uniendo macro y microclima en paisajes de montaña: una aproximación conceptual e instrumental" (30): Es un artículo que tiene como objetivo estudiar cómo el cambio climático afecta a la adaptación de la biodiversidad en entornos montañosos. Haciendo hincapié en la comprensión y la caracterización de los microclimas para entender el macroclima de la región montañosa.
- “*A Framework for Knowledge Discovery from Wireless Sensor Networks in Rural Environments: A Crop Irrigation Systems Case Study*” (31): Este proyecto refleja un sistema multiagente basado en organizaciones virtuales, hace uso de sensores inalámbricos para realizar una red que estudie y actúe sobre los entornos rurales.

3. Nodos sensores

Antes de explicar el montaje del sistema completo, es necesario ahondar en las distintas partes que van a integrar el proyecto, para así poder obtener una visión más amplia del funcionamiento de la red.

En este apartado se procederá a exponer los nodos de medición que se han implementado tanto en el presente proyecto como en el anterior (3) para conformar la red IoT. Explicando desde las magnitudes de medida, pasando por los sensores que las toman, hasta la implementación e integración de los nodos en el entorno en estudio.

3.1. Magnitudes a medir y sensores utilizados

Para poder implementar un buen sistema de monitorización es necesario entender las magnitudes que se han de medir. Todo ello para analizar correctamente, no solo el árbol en estudio, si no para ser capaz de deducir el estado del bosque completo y estudiar la evolución del cambio climático.

Para lograr estudiar de forma adecuada el desarrollo del cambio climático en el entorno forestal no solamente se han de medir las magnitudes relacionadas con los árboles, también será necesario estudiar todas aquellas que influyan directamente en su desarrollo. Por lo cual, se hará uso de magnitudes propias del árbol, como del suelo o las magnitudes medioambientales que influyen en su adaptación al medio.

A continuación, se enumeran algunos de los múltiples parámetros que influyen en el correcto desarrollo del árbol, estos se dividirán en función de si estudian al ejemplar o si estudian las condiciones climáticas entorno a este.

Magnitudes que afectan directamente al árbol:

- Temperatura en los estomas.
- Humedad emitida por la hoja.
- Diámetro de tallo, pecíolo y tronco.
- Tamaño de fruto.
- Flujo de savia.
- Oquedad del árbol (A través de sonido percutido).
- Resistencia de la madera (A través de corriente eléctrica).
- Presión del jugo celular sobre la hoja.
- Oxígeno (O_2).
- Dióxido de carbono (CO_2).
- Nitrógeno (N).
- Etileno (C_2H_4).
- Clorofila.
- pH del suelo.
- Humedad del suelo.
- Observación del estado de la planta.

Magnitudes ambientales:

- Viento.
- Temperatura ambiente.
- Luz incidente e índice de radiación ultravioleta (UVI).
- Humedad del suelo y relativa del aire.
- Precipitaciones.
- Presión atmosférica y del vapor de agua.

También es imprescindible realizar el estudio de aquellos sensores que pueden medir de forma apropiada las magnitudes anteriormente mencionadas. En el caso de las magnitudes propias del árbol se han de buscar sensores que puedan realizar mediciones en entornos abiertos y hostiles como puede ser un entorno forestal. Para la obtención de los datos de las magnitudes ambientales se podría utilizar directamente estaciones meteorológicas. Todo este estudio se puede observar en el [apéndice A](#).

Una vez estudiadas las magnitudes y los sensores se proceden a elegir cuales son los más idóneos para la monitorización de coníferas al tener unas características específicas, como tamaño de hoja, entre otras. Es importante recalcar que, tanto para la elección de magnitudes como de sensores, al estar interrelacionados, se han de tomar en cuenta distintos parámetros como son:

- Condiciones climáticas que deben soportar los sensores.
- Precio.
- Distribuidor.
- Disponibilidad.
- Tiempo de envío.
- Precisión.
- Respuesta temporal.
- Dimensiones.
- Facilidad a la hora de colocarlo.
- Durabilidad.
- Empaque y protección.
- Tipo de salida.
- Protocolo de comunicación.
- Software (SW) compatible.
- Distancia de comunicación.
- Consumo.
- Tipo de alimentación.
- Neutro para el medio ambiente.

Todos estos parámetros se discutieron de una manera más detallada en el TFM anterior (3). Finalmente, los sensores con las magnitudes asociadas que se eligieron para el desarrollo de los nodos sensores son:

Elección de sensores

Modelo	Magnitud	Página web
LAT-B3	Tº estomática	https://ecomatik.de/site/assets/files/21750/usermanual_lat-b3.pdf
PYTHOS-31	Humedad hoja	https://library.metergroup.com/Manuals/20434_PHYTOS31_Manual_Web.pdf
SEN0322	Oxígeno hoja	https://wiki.dfrobot.com/Gravity_I2C_Oxygen_Sensor_SKU_SEN0322
MH-Z19C	CO2 Hoja	https://www.winsen-sensor.com/d/files/mh-z19c-pins%26terminal-type-co2-manual(ver1_2).pdf
Soil NPK Sensor	NPK	https://www.renkeer.com/product/soil-npk-sensor/
SEN0249	pH	https://wiki.dfrobot.com/Gravity__Analog_Spear_Tip_pH_Sensor__Meter_Kit__For_Soil_And_Food_Applications_SKU_SEN0249
314990620	Humedad suelo	https://files.seeedstudio.com/products/101990668/res/RS485%20Soil%20Moisture%20&%20Temperature%20Sensor%20(S-Soil%20MT-02)-Datasheet.pdf
Sensecap	EC, salinidad...	https://files.seeedstudio.com/wiki/Soil_Moisture_Temperature_EC_Sensor/SoilMoisture_Temperature_ECSensorUserManual-S-Temp&VWC&EC-02.pdf
DS18B20	Tº y humedad suelo	https://www.analog.com/media/en/technical-documentation/data-sheets/ds18b20.pdf
SHT20	Tº y humedad aire	https://wiki.dfrobot.com/SHT20_I2C_Temperature_%26_Humidity_Sensor__Waterproof_Probe_SKU_SEN0227
NEO-6M	GPS	https://www.tme.eu/es/details/neo-6m-0/modulos-gnss-gps/u-blox/neo-6m-0-001/
YF-S201	Caudal	https://www.adafruit.com/product/828
SEN0313	Distancia	https://cdn-shop.adafruit.com/product-files/828/C898+datasheet.pdf
Sainlogic WS 3500 Plus	Estación meteorológica	https://estacionmeteorologicaprofesional.com/sainlogic/sainlogic-ws3500-10-en-1/

Tabla 3.1.1 Selección de los sensores a utilizar (3)

3.2. Implementación de los nodos

Se procede a explicar cómo se realizaron los distintos nodos de medición que se utilizarán en este TFM para implementar la red IoT. Esto será de gran utilidad para lograr entender de forma adecuada el sistema implementado.

En cada uno de estos se explicará tanto el concepto previo de diseño como la implementación final llevada a cabo. Exponiendo las políticas propias de cada tipo de nodo, el sistema de alimentación, el desarrollo a nivel HW y SW, y como se prepararán los datos para ser transmitidos a los nodos centrales.

3.2.1. Primer tipo de nodo sensor implementado en (3)

El primer nodo implementado tiene como objetivo tomar solamente datos provenientes del ejemplar. Esto lo hará a través de unas placas de circuito impreso (PCB) que se encargarán de alimentar y, si es necesario, acondicionar los distintos sensores. Estas placas se han separado en función de la posición que tomen los sensores en el ejemplar, es decir, existe una PCB encargada de la alimentación y acondicionamiento de los sensores que van situados en el follaje del árbol y otra que tendrá una función similar a la anterior, pero para los sensores de suelo.

Al ser un sistema pensado para ser eficiente energéticamente hablando, para alimentarlo se hará uso de una batería de gran capacidad. Esta proporcionará el suministro de energía suficiente para poder tomar medidas por un largo periodo de tiempo.

La figura 3.2.1 es un esquema conceptual del diseño implementado para el sistema. Se puede observar que se adquirirán y se preprocesaran los datos con un microprocesador *Arduino Uno*, este será el encargado de transferir los datos por protocolo I2C a la *Raspberry Pi* que se le encomendará la transmisión de los datos al nodo central.

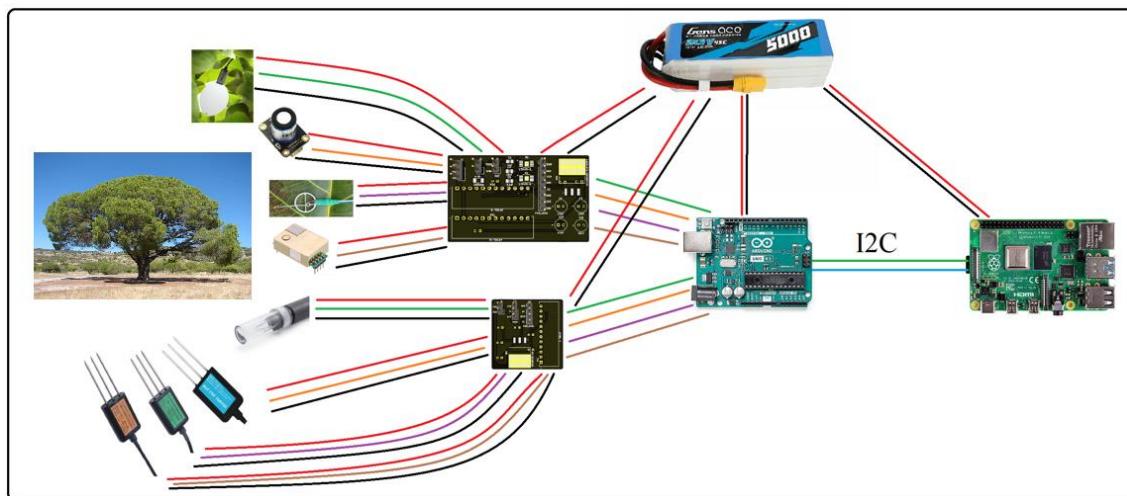


Figura 3.2.1 Esquema conceptual primer nodo (3)

Para poder alimentar adecuadamente los microprocesadores se decide crear otra PCB para suministrarle correctamente energía a sendos dispositivos. Para el diseño de las PCB se empleó la herramienta SW KiCad 7.0, posteriormente se imprimieron y se comprobaron que tanto la alimentación como el acondicionamiento funcionaban de la forma preestablecida. Una vez realizado este estudio previo se procedió a implementar este nodo.

NODOS IMPLEMENTADOS

Debido a una serie de inconvenientes con los sensores o debido al protocolo de comunicación que utilizaban, se decide reducir el número de sensores para no complicar el sistema. Finalmente, se resuelve hacer uso de los sensores PYTHOS-31, SEN0322, MH-Z19C y SEN0249.

En la figura 3.2.2 se muestra un esquema del circuito implementado para la toma de datos por el primer nodo. Se refleja, en dicha imagen, cómo se deben conectar los sensores y los microcontroladores para recibir de forma adecuada la alimentación y acondicionamiento preciso. Además, tanto en la placa como en la *Raspberry Pi* existe un pin de control (CTR) que se encarga de encender y apagar el sistema para lograr ahorrar recursos energéticos.

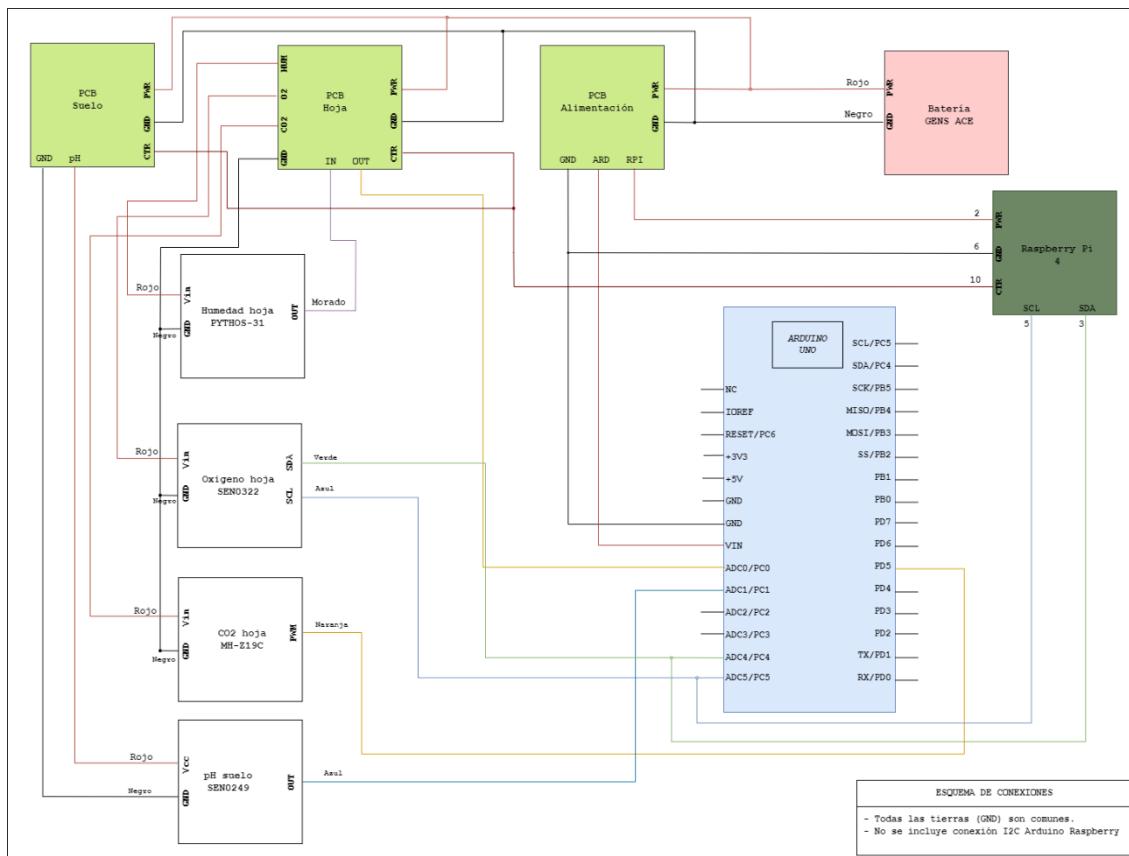


Figura 3.2.2 Esquema de conexión del primer nodo (3)

Una vez implementada la parte física del sistema, se desarrollaron los códigos de *Arduino Uno* y *Raspberry Pi* para poder adquirir, preprocesar y almacenar de forma temporal los datos. Cada vez que se quiera almacenar un dato (normalmente cada veinte minutos), la *Raspberry* le enviará una petición al *Arduino* para que esta adquiera los datos, los trate y los envíe.

Para la adquisición y preprocesamiento se hace uso del *Arduino*. Este contendrá dos tipos de códigos que se encargarán de la adquisición y preprocesamiento de datos:

- El primer modo se encarga de tomar datos cada vez que la *Raspberry Pi* le envíe la petición, pero no hará ningún preprocesamiento de las medidas.
- El segundo modo se encarga de tomar los datos durante cierto tiempo (normalmente treinta segundos) cada vez que la *Raspberry Pi* le envíe la petición, con estos datos tomados se realizará una media winsorizada o truncada

NODOS IMPLEMENTADOS

para obtener una medida representativa que mitigue los efectos de los valores atípicos.

Una vez la *Raspberry* envía la petición y recibe el dato, este lo almacenará en un archivo JSON cuyo nombre es el mes y el año en el que se ha tomado la medida, cada mes se actualiza el nombre del archivo. Para tomar una mayor cantidad de datos, se utiliza el primer modo enviando una petición cada diez segundos.

En la figura 3.2.3 se muestra el final del archivo de los datos tomados en el mes de agosto, comprobando con el *timestamp* que la última medida tomada fue a las 23:59:59. En la figura 3.2.4 se expone el principio del fichero de toma de datos del mes de septiembre, comprobando que se tomó la medida a las 00:00:09.

```

8_2023.json - /home/tfmanuel/Desktop/Codigo - Geany
Archivo Editar Buscar Ver Documento Proyecto Construir Herramientas Ayuda
Símbolos tx_data.json ✘ tx_implicit_example.c ✘ i2cmasterM2vfin.py ✘ Prueba2.py ✘ i2cmastervfin.py ✘ 9_2023.json ✘ 8_2023.json ✘
Miembros
54644 {"timestamp": "16935189918", "humidity": 17, "co2": 1978, "oxygen": "20.10", "pH": 7.04}
54645 {"timestamp": "16935189928", "humidity": 8, "co2": 1993, "oxygen": "20.10", "pH": 8.16}
54646 {"timestamp": "16935189938", "humidity": 18, "co2": 1993, "oxygen": "20.10", "pH": 8.16}
54647 {"timestamp": "16935189948", "humidity": 28, "co2": 1993, "oxygen": "20.10", "pH": 8.32}
54648 {"timestamp": "16935189958", "humidity": 29, "co2": 1993, "oxygen": "20.10", "pH": 8.08}
54649 {"timestamp": "16935189968", "humidity": 28, "co2": 1993, "oxygen": "20.10", "pH": 8.32}
54650 {"timestamp": "16935189978", "humidity": 24, "co2": 1993, "oxygen": "20.10", "pH": 8.0}
54651 {"timestamp": "16935189988", "humidity": 24, "co2": 1993, "oxygen": "20.10", "pH": 8.16}
54652 {"timestamp": "16935189998", "humidity": 24, "co2": 1993, "oxygen": "20.10", "pH": 8.08}
54653 {"timestamp": "16935190008", "humidity": 0, "co2": 1993, "oxygen": "20.10", "pH": 8.32}
54654 {"timestamp": "16935190018", "humidity": 0, "co2": 1993, "oxygen": "20.10", "pH": 8.08}
54655 {"timestamp": "16935190028", "humidity": 17, "co2": 1993, "oxygen": "20.10", "pH": 8.24}
54656 {"timestamp": "16935190038", "humidity": 18, "co2": 1993, "oxygen": "20.10", "pH": 7.84}
54657 {"timestamp": "16935190048", "humidity": 16, "co2": 1993, "oxygen": "20.10", "pH": 8.0}
54658 {"timestamp": "16935190058", "humidity": 18, "co2": 1993, "oxygen": "20.10", "pH": 7.84}
54659 {"timestamp": "16935190068", "humidity": 16, "co2": 1993, "oxygen": "20.10", "pH": 7.44}
54660 {"timestamp": "16935190078", "humidity": 16, "co2": 1993, "oxygen": "20.10", "pH": 7.2}
54661 {"timestamp": "16935190088", "humidity": 16, "co2": 1993, "oxygen": "20.10", "pH": 7.12}
54662 {"timestamp": "16935190098", "humidity": 16, "co2": 1993, "oxygen": "20.10", "pH": 7.04}
54663 {"timestamp": "16935191008", "humidity": 16, "co2": 1993, "oxygen": "20.10", "pH": 7.28}
54664 {"timestamp": "16935191108", "humidity": 15, "co2": 1993, "oxygen": "20.10", "pH": 7.28}
54665 {"timestamp": "16935191128", "humidity": 34, "co2": 1993, "oxygen": "20.10", "pH": 7.28}
54666 {"timestamp": "16935191138", "humidity": 17, "co2": 1993, "oxygen": "20.10", "pH": 6.72}
54667 {"timestamp": "16935191148", "humidity": 16, "co2": 1993, "oxygen": "20.10", "pH": 7.28}
54668 {"timestamp": "16935191158", "humidity": 16, "co2": 1993, "oxygen": "20.10", "pH": 7.04}
54669 {"timestamp": "16935191168", "humidity": 17, "co2": 1993, "oxygen": "20.10", "pH": 4.8}
54670 {"timestamp": "16935191179", "humidity": 18, "co2": 1993, "oxygen": "20.10", "pH": 8.32}
54671 {"timestamp": "16935191189", "humidity": 26, "co2": 1993, "oxygen": "20.10", "pH": 8.16}
54672 {"timestamp": "16935191199", "humidity": 22, "co2": 1993, "oxygen": "20.10", "pH": 8.16}

linea 54672 col:14 sel:10 INS TAB mode LF codificación:UTF-8 tipo de archivo:JSON ámbito: desconocido

```

Figura 3.2.3 Final del archivo 08_2023.json

```

*_9_2023.json - /home/tfmanuel/Desktop/Codigo - Geany
Archivo Editar Buscar Ver Documento Proyecto Construir Herramientas Ayuda
Símbolos tx_data.json ✘ tx_implicit_example.c ✘ i2cmasterM2vfin.py ✘ Prueba2.py ✘ i2cmastervfin.py ✘ 9_2023.json ✘ 9_2023.json ✘
Miembros
1 {"timestamp": "1693519209", "humidity": 32, "co2": 1993, "oxygen": "20.10", "pH": 8.08}
2 {"timestamp": "1693519219", "humidity": 22, "co2": 1993, "oxygen": "20.10", "pH": 8.16}
3 {"timestamp": "1693519229", "humidity": 0, "co2": 1993, "oxygen": "20.10", "pH": 8.24}
4 {"timestamp": "1693519239", "humidity": 0, "co2": 1993, "oxygen": "20.10", "pH": 8.24}
5 {"timestamp": "1693519249", "humidity": 0, "co2": 1993, "oxygen": "20.10", "pH": 8.16}
6 {"timestamp": "1693519259", "humidity": 3, "co2": 1993, "oxygen": "20.10", "pH": 8.16}
7 {"timestamp": "1693519269", "humidity": 7, "co2": 1993, "oxygen": "20.10", "pH": 8.16}
8 {"timestamp": "1693519279", "humidity": 22, "co2": 1993, "oxygen": "20.10", "pH": 8.08}
9 {"timestamp": "1693519289", "humidity": 23, "co2": 1993, "oxygen": "20.10", "pH": 8.08}
10 {"timestamp": "1693519299", "humidity": 26, "co2": 1993, "oxygen": "20.10", "pH": 8.0}
11 {"timestamp": "1693519309", "humidity": 23, "co2": 1993, "oxygen": "20.10", "pH": 8.08}
12 {"timestamp": "1693519319", "humidity": 23, "co2": 1993, "oxygen": "20.10", "pH": 8.16}
13 {"timestamp": "1693519329", "humidity": 16, "co2": 1993, "oxygen": "20.10", "pH": 7.12}
14 {"timestamp": "1693519339", "humidity": 17, "co2": 1993, "oxygen": "20.10", "pH": 6.96}
15 {"timestamp": "1693519349", "humidity": 16, "co2": 1993, "oxygen": "20.10", "pH": 7.2}
16 {"timestamp": "1693519359", "humidity": 17, "co2": 1993, "oxygen": "20.10", "pH": 7.04}
17 {"timestamp": "1693519369", "humidity": 16, "co2": 1993, "oxygen": "20.10", "pH": 7.12}
18 {"timestamp": "1693519379", "humidity": 16, "co2": 1993, "oxygen": "20.10", "pH": 7.12}
19 {"timestamp": "1693519389", "humidity": 17, "co2": 1956, "oxygen": "20.10", "pH": 7.28}
20 {"timestamp": "1693519399", "humidity": 18, "co2": 1956, "oxygen": "20.10", "pH": 7.2}
21 {"timestamp": "1693519409", "humidity": 17, "co2": 1956, "oxygen": "20.10", "pH": 6.88}
22 {"timestamp": "1693519419", "humidity": 0, "co2": 1956, "oxygen": "20.10", "pH": 7.28}
23 {"timestamp": "1693519429", "humidity": 0, "co2": 1956, "oxygen": "20.10", "pH": 7.12}
24 {"timestamp": "1693519439", "humidity": 16, "co2": 1956, "oxygen": "20.10", "pH": 6.96}
25 {"timestamp": "1693519449", "humidity": 16, "co2": 1956, "oxygen": "20.10", "pH": 6.64}
26 {"timestamp": "1693519459", "humidity": 17, "co2": 1956, "oxygen": "20.10", "pH": 7.04}
27 {"timestamp": "1693519469", "humidity": 18, "co2": 1956, "oxygen": "20.10", "pH": 6.96}
28 {"timestamp": "1693519479", "humidity": 18, "co2": 1956, "oxygen": "20.10", "pH": 6.96}
29 {"timestamp": "1693519489", "humidity": 20, "co2": 1956, "oxygen": "20.10", "pH": 8.16}
30 {"timestamp": "1693519499", "humidity": 29, "co2": 1956, "oxygen": "20.10", "pH": 8.24}
31 {"timestamp": "1693519509", "humidity": 26, "co2": 1956, "oxygen": "20.10", "pH": 8.16}

linea 1 / 6259 col:0 sel:0 INS TAB MOD mode LF codificación:UTF-8 tipo de archivo:JSON ámbito: desconocido

```

Figura 3.2.4 Principio del archivo 09_2023.json

NODOS IMPLEMENTADOS

3.2.2. Segundo tipo de nodo sensor implementado en (3)

En el segundo nodo se hace uso, para la adquisición, preprocesamiento y almacenamiento temporal de los datos adquiridos por los sensores, de dos dispositivos de *Dragino*. Estos dispositivos son el NBSN95 (32) que tiene como protocolo de transmisión de datos NB-IoT en GSM y el microprocesador LSN50 (33) usa LoRa como protocolo de comunicación.

El código implementado en ambos dispositivos tiene el mismo cometido. Se crearon distintos modos de funcionamiento, los cuales se encargarán de obtener los datos de distintos sensores, debido a que cada configuración se podrá situar en un entorno distinto para la medición del entorno forestal.

Sendos dispositivos venían con unos modos de funcionamiento propios, los cuales hacían uso de otros sensores que pueden ser de utilidad para el presente sistema. Estos modos son fácilmente intercambiables a través de los comandos AT. Los nuevos modos se implementaron sin eliminar a los modos anteriores, otorgando más funcionalidades al sistema.

Tanto los modos proporcionados por *Dragino* como los creados para el desarrollo de la presente red vienen reflejados en las siguientes tablas, donde se relaciona que sensores, se utilizan en cada modo y que tipo de continuación utilizan:

		DRAGINO	Modos de funcionamiento						Tipo de comunicación					
			1	2	3	4	5	6	RS485	RS232	I2C	ADC	GPIO	1-WIRE
	Sensor													
	DS18B20		1	1		3	1	1						X
	Sensores ADC		1	1	3	1	1	1					X	
	Contador de pulsos								1					X
	SHT31/AHT20			1										
	Sensores I2C					1/-						X		
Uno u otro	Entrada digital		1	1	1	1	1	1						X
	Entrada interrupción		1	1	1	1	1	1						X
Uno u otro	Lidar			1								X		
	Ultrasonido				1							X		
	HX711/PESO						1							X
	Total		4	4	4/-	5	4	4						

Figura 3.2.5 Modos de funcionamiento creados por el fabricante (3)

		DISEÑO	Modos de funcionamiento							Tipo de comunicación					
ID Sensor	Magnitud		7	8	9	10	11	12	13	RS485	RS232	I2C	ADC	GPIO	1-WIRE
A	Tº Suelo	DS18B20					1								X
B	-	Sensores ADC											X		
C	Caudal	Contador de pulsos			1	1	1	1	3						X
D	Tº, hum aire	SHT31/AHT20			1		4			1					X
E	Posición	GPS	1	1	1	1	1					X			
F	Lluvia	Pluviómetro		1								X			
G	Tº, hum suelo	Sensecap		? 3	3							X			
H	Ambiente	Estación meteorológica (WH24C)	1									X			
		Total	2	4/7	5	6	3	3	5						
		¿Es viable?	Sí	Sí	Sí	Sí	Sí	Sí	Sí						

Figura 3.2.6 Modos de funcionamiento diseñados para la creación del nodo (3)

Por problemas de incompatibilidad, no se hizo uso del pluviómetro. Igualmente, se procedió a ver qué datos proporcionaban cada uno de los sensores, algunos daban tanto datos de configuración como de medición. En el presente sistema, para simplificarlo y

NODOS IMPLEMENTADOS

ahorrar tiempo de cómputo y batería, solamente se van a transmitir los datos útiles de medición, el resto se descarta.

El funcionamiento del sistema es tomar medidas cada cierto tiempo (normalmente veinte minutos), se encienden durante un pequeño periodo de tiempo los sensores para ir tomando datos y se preprocesan. Estos datos se almacenarán de forma temporal en unas estructuras, esperando a ser transmitidos y a que se reinicen sus valores.

En el [apéndice B](#) se muestran los esquemas de los circuitos necesarios para el diseño del nodo con relación al modo de funcionamiento utilizado.

Para la alimentación de este nodo se hará uso de una placa fotovoltaica de 5 V, 60 mA de un tamaño pequeño para encajarla en la carcasa del dispositivo, además de hacer uso de una batería Li/SOCl2.

3.2.3. Implementación del tercer tipo de nodo sensor

Este nodo sensor es más sencillo que los implementados con anterioridad. Se hará uso de las funcionalidades propias, y tremadamente útiles, que tiene la estación meteorológica Sainlogic WS3500.

Este nodo se encargará de recopilar los datos obtenidos de las distintas magnitudes medioambientales para transmitirlas directamente a diversas bases de datos, esto facilitará su estudio y comparación con datos recopilados por otras estaciones.

El único inconveniente es que se ha de utilizar en lugares cercanos a los nodos centrales o en entornos rurales con conexión, al precisar de una red Wifi para poder hacer uso de estas funcionalidades. Por tanto, podría ser muy útil para estudiar y comparar los datos meteorológicos recibidos del bosque con los de las poblaciones cercanas a este entorno.

Para alimentar el sistema se pueden utilizar baterías 2XAA, pero esto no proporcionará demasiada autonomía. Además, como ha de estar cerca de un sitio con red Wifi, se podría modificar el sistema eliminando la carcasa donde se incluyen las baterías y conectar sus cables a un conversor de tres voltios. Esto proporcionaría una autonomía prácticamente total, siempre que haya red eléctrica y Wifi.

3.3. Integración en el entorno

Se procede a exponer, en esta sección, cómo se han de colocar los distintos nodos en el entorno en estudio. Se hace fundamental explicar de forma adecuada esta integración al tener un papel crítico en la adquisición de unos valores reales y precisos.

3.3.1. Integración del primer nodo sensor

Se va a llevar a cabo una explicación a grandes rasgos de la integración del sistema montado, es decir, se explicará la integración del sistema con los sensores utilizados. Es importante recalcar que este sistema está diseñado con dos PCB que van a tener como misión alimentar y acondicionar los sensores de hoja o de suelo.

Los microprocesadores que se encargarán de la adquisición, procesamiento y transmisión de los datos han de situarse relativamente cerca unos de otros. Además, la

NODOS IMPLEMENTADOS

antena de la que disponen ha de estar orientada de forma adecuada, intentando tener línea directa con la antena del nodo central y en una posición prácticamente vertical respecto al suelo para disminuir los efectos de las pérdidas de polarización.

Las PCB deben situarse en un punto medio entre los sensores y los microprocesadores, de tal forma que la PCB que va a alimentar y acondicionar a los sensores de hoja ha de colocarse a media altura del tronco, para que los sensores puedan situarse de forma adecuada en el follaje de los ejemplares. La PCB de suelo, como su propio nombre indica, ha de dejarse a nivel de suelo, cerca de los sensores que medirán la tierra entorno al árbol. Sería conveniente proteger a las PCB haciendo uso de carcchas protectoras, así podrán soportar las inclemencias y ataques de la fauna y flora circundante.

Cada sensor que va situado a las hojas tiene su complicación, la mayoría de ellos se tendrán que situar en una zona muy poblada de follaje para poder obtener la mejor medición posible. Si esto último no es posible, es necesario conocer que los datos obtenidos podrían estar en gran parte alterados por el medioambiente que les rodea.

Los sensores que se encargan de medir las magnitudes propias del suelo, en el caso de querer tomar medidas esporádicas, simplemente se han de clavar sus puntas metálicas en el suelo. Sin embargo, para mediciones prolongadas en el entorno forestal, es muy recomendable enterrarlas por completo, cerca de las raíces a menos de medio metro de profundidad.

Se hace necesario explicar que el sensor de pH utilizado cuenta de base con un vial con un contenido para calibrar el sensor, para integrarlo en el sistema se ha de quitar este vial, intentando no volcar el contenido. Posteriormente, se ha de quitar el anillo de plástico que está enganchado al sensor, para poder meter la punta metálica y sujetarla con el anillo previamente quitado.

3.3.2. Integración del segundo nodo sensor

El segundo nodo tiene distintos modos de funcionamiento, la instalación del sistema en el entorno dependerá totalmente del modo utilizado. La mayoría de los sensores sirven para el control de las magnitudes de suelo, por tanto, para instalarlas en largos períodos de tiempo se deben enterrar cerca de las raíces.

Los modos que hacen uso del sensor de GPS deben estar situados en una zona boscosa, pero donde el follaje no impida que el sensor se comunique con los satélites. A su vez, se pueden utilizar los sensores de humedad y temperatura SHT20 para poder estudiar dichas magnitudes, estos se deben situar en una zona frondosa con abundantes hojas.

Se puede emplear también el tipo de estación meteorológica WS3500 para poder tomar las medidas ambientales. Estas se situarán en zonas boscosas, cosa que difiere con el próximo nodo.

3.3.3. Integración del tercer nodo sensor

Este nodo, al estar solamente compuesto por la estación meteorológica, únicamente se ha de procurar la correcta instalación de esta en el entorno de medición que, en esta ocasión, es en zonas rurales cercanas al bosque en estudio.

NODOS IMPLEMENTADOS

Hay que destacar que se cuenta con dos estaciones meteorológicas idénticas, una se utiliza en este nodo (tercer nodo) y la otra en el nodo anterior (segundo nodo). Para completar esta sección se explicará con detalle la integración de la estación meteorológica en el entorno.

Inicialmente, se ha de montar la estación meteorológica haciendo uso de cada uno de los componentes que completan a los sensores:

- Para el sensor que mide el direccionamiento del viento, se ha de incorporar la veleta en el soporte metálico mellado.
- El anemómetro posee una pieza con forma de molino con aspas cóncavas que se ha de situar en el soporte metálico.
- Al pluviómetro se le ha de integrar un cuenco, para ello se han de poner las pestañas en las ranuras destinadas para este fin y girarlo.

Una vez integrado cada componente, es necesario hacer uso de las mordazas para ajustarlo al mástil donde irá situada la estación meteorológica. También es fundamental que esta estación esté orientada hacia el norte, para ello existe una muesca en forma de triángulo en la carcasa que nos indica que su punta se ha de dirigir al norte. A su vez, posee un nivel para informar de que la estación está nivelada.

Sería adecuado dejar la estación meteorológica a una distancia próxima a los cien metros del terminal de recepción, siempre y cuando no haya demasiados obstáculos en su línea de visión.

4. Transmisión y recepción de los datos

En este apartado se estudiará la forma en la que se han de transmitir los datos y cuál sería el nodo de recepción relativo a cada tipo de nodo de medida. Esta parte se hace necesaria para el correcto entendimiento de las características propias de la red como son sus protocolos de comunicación, rangos de cobertura, gestión en la forma de tratamiento e intercambio de los datos, entre otras.

4.1. Primer nodo

Este nodo está diseñado para localizarse en un entorno abierto, alejado de las zonas pobladas. Los datos recibidos de los nodos de medición se transmitirán a una BD en la nube, por tanto, se precisará de un nodo central que realice el papel de intermediario para poder almacenar los datos recibidos y que tenga conexión a Internet.

Al precisar de un sistema de largo alcance, es necesario utilizar un protocolo de comunicación a larga distancia para que, de este modo, el nodo central proporcione un amplio rango de cobertura a los diferentes nodos que se implementarían en el entorno forestal. Sin embargo, es fundamental que no se consuma demasiada energía en el proceso de transmisión de datos, al precisar de nodos de medición autónomos y al ser de difícil sustitución por su localización.

Para el cumplimiento de este fin se va a hacer uso del protocolo de comunicación de amplio alcance y bajo consumo LoRa. A su vez, este puede emplear distintos rangos de frecuencias para la transmisión de datos como son las bandas de 433 MHz, 868 MHz, 915 MHz (34). Para obtener un mayor rango de cobertura, se hará uso de la frecuencia de 433 MHz al tener menores pérdidas de trayectoria en espacio libre.

La *Raspberry* no posee ningún módulo ni antena propios para transmisión LoRa, por tanto, será necesario incorporarlos al sistema final para que el nodo central y el de medición se puedan comunicar de forma idónea. Tras una selección, se eligió el módulo *Ra-02 SX1278*, ver figura 4.1.1, que tiene la capacidad de integrarse a la *Raspberry*, además de que se le puede añadir la antena de 433 MHz y 3dBi de *Dolla Tek* mostrada en la figura 4.1.2.

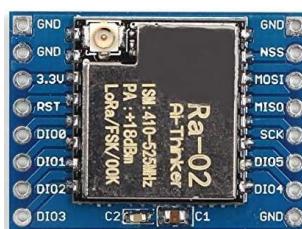


Figura 4.1.1 Módulo LoRa Ra-02 SX1278 (35)

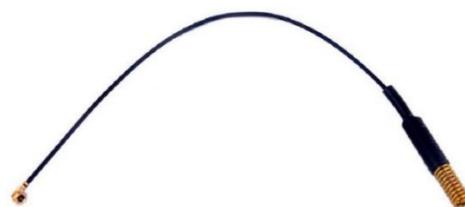


Figura 4.1.2 Antena 433 MHz y 3 dBi de Dolla Tek (36)

TRANSMISIÓN Y RECEPCIÓN DE LOS DATOS

Una vez recibidos por el nodo central los datos a través del protocolo LoRa, este creará un fichero para tener una copia de seguridad de los datos y, posteriormente, los transmitirá a la BD en la nube *InfluxDB Cloud*. El esquema conceptual de todo el sistema desarrollado en esta sección se muestra en la siguiente figura:

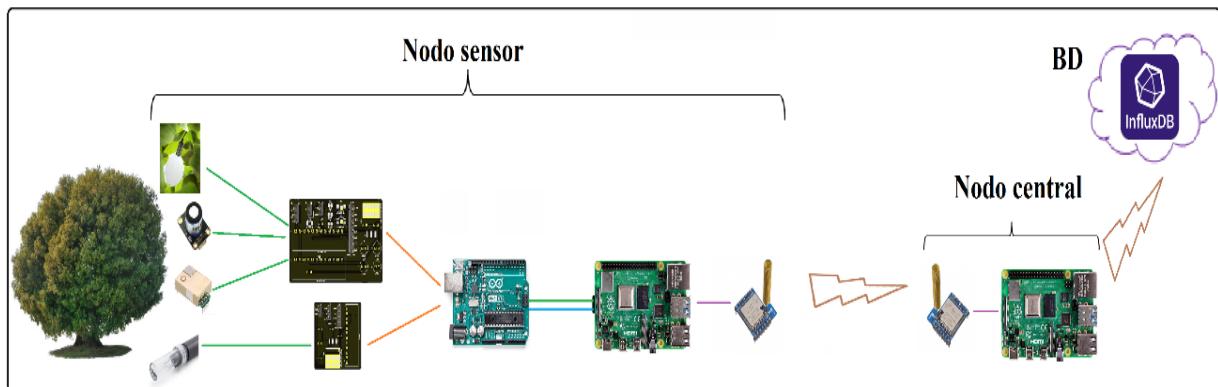


Figura 4.1.3 Esquema conceptual del primer sistema implementado

Para garantizar una operación eficaz y una integración efectiva entre la *Raspberry* y el módulo adquirido, se hizo esencial utilizar el repositorio de *Github* de Yandiev Ruslan *sx1278-LoRa-RaspberryPi* (37). Para comunicar ambos dispositivos es necesario realizar las conexiones que se muestran en la siguiente tabla:

Raspberry Pi 4 GPIO	Módulo Ra-02 GPIO
GPIO4, pin7	Reset
GPIO17, pin 11	DIO0
MOSI (GPIO10, pin 19)	MOSI
MISO (GPIO9, pin 21)	MISO
CLK (GPIO11, pin 23)	SCK
SPI_CE0 (GPIO8, pin24)	NSS

Tabla 4.1.1 Pines para la integración del módulo Ra-02 en la Raspberry Pi 4 (37)

Estudiando las hojas de características de la antena (36) y el módulo LoRa (38) se recopilaron los datos necesarios para hacer el estudio del balance de enlace del sistema. Todo esto para comprobar el rango de cobertura disponible del sistema.

En la tabla 4.1.2 se observan tanto los datos característicos del sistema implementado para realizar el cálculo del balance de enlace, como los factores que influyen en las pérdidas del sistema. Se quiere indicar que se han elegido las características más adversas para establecer, de esta manera, el límite en el que el sistema funcionaría correctamente.

Símbolo de la medida	Nombre	Valor	Unidades
f	Frecuencia	433	MHz
λ	Longitud de onda	692,84	mm
P _t	Potencia transmitida	24,87	dBm
P _r	Sensibilidad del Receptor	-125	dBm

TRANSMISIÓN Y RECEPCIÓN DE LOS DATOS

G_t	Ganancia de la antena Transmisora	3	dBi
G_r	Ganancia de la antena Receptora	3	dBi
ROE_t	Relación de onda estacionaria de la antena Transmisora	1,50	V/V
ROE_r	Relación de onda estacionaria de la antena Receptora	1,50	V/V
$\rho_t = \frac{ROE_t - 1}{ROE_t + 1}$	Coeficiente de reflexión de la antena Transmisora	0,20	V/V
$\rho_r = \frac{ROE_r - 1}{ROE_r + 1}$	Coeficiente de reflexión de la antena Receptora	0,20	V/V
$1 - \rho_t^2$	Pérdidas por desadaptación de la antena Transmisora	0,96	V^2/V^2
$1 - \rho_r^2$	Pérdidas por desadaptación de la antena Receptora	0,96	V^2/V^2
$ \hat{e}_t \hat{e}_r^* ^2$	Pérdidas por polarización	0,9	
L_{rg}	Pérdidas por Ruido Galáctico	1,20	dB/km
L_{rf}	Pérdidas por Ruido de Fondo	0,035	dB/km
L_{zb}	Pérdidas por Zona Boscosa	0,09	dB/km

Tabla 4.1.2 Datos de las antenas y módulos de comunicación para ver el balance de enlace del primer nodo

Los primeros datos de la tabla, vista previamente, muestran los valores de los parámetros propios de las antenas y módulos que se muestran en sus hojas de características y la transformación a parámetros útiles para la realización del estudio del balance de enlace.

En cuanto a las pérdidas de polarización, el sistema está ideado, originalmente para que el técnico que se encargue de instalar los nodos en el entorno a monitorizar sitúe las antenas de tal forma que no haya una degradación de la señal debida a la polarización. Sin embargo, no sería extraño que, al estar situado en un entorno tan hostil, a lo largo del tiempo las antenas se vean ligeramente desalineadas. De esta forma, se decidió utilizar un parámetro por pérdidas ligeramente inferior al ideal al poderse ver modificada la posición de estas antenas.

Otros posibles parámetros que pueden influir en la atenuación del sistema son las pérdidas debidas a condiciones atmosféricas o por ruido cósmico, para el estudio del balance de enlace se va a utilizar la figura 4.1.4. Los ruidos que más van a influir en el entorno de medición a la frecuencia de trabajo son el ruido galáctico y el del *Big Bang*.

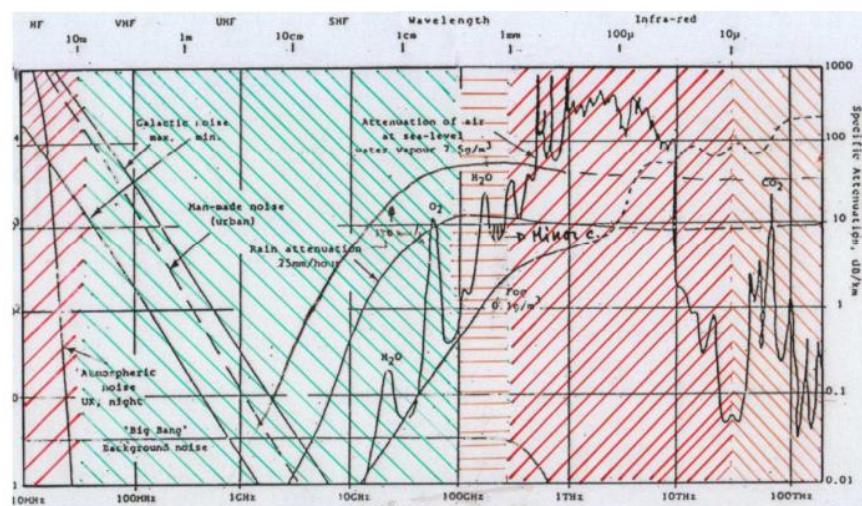


Figura 4.1.4 Esquema conceptual del primer sistema implementado (39)

TRANSMISIÓN Y RECEPCIÓN DE LOS DATOS

Además, al ser el entorno de medida un medio forestal, se hará uso de la gráfica patente en la figura 4.1.5 para poder tener una aproximación de la influencia de este entorno en la comunicación del sistema. Se tenía pensado de antemano hacer uso de una polarización vertical al ser menos susceptible a las posibles reflexiones originadas por el efecto del suelo en la comunicación.

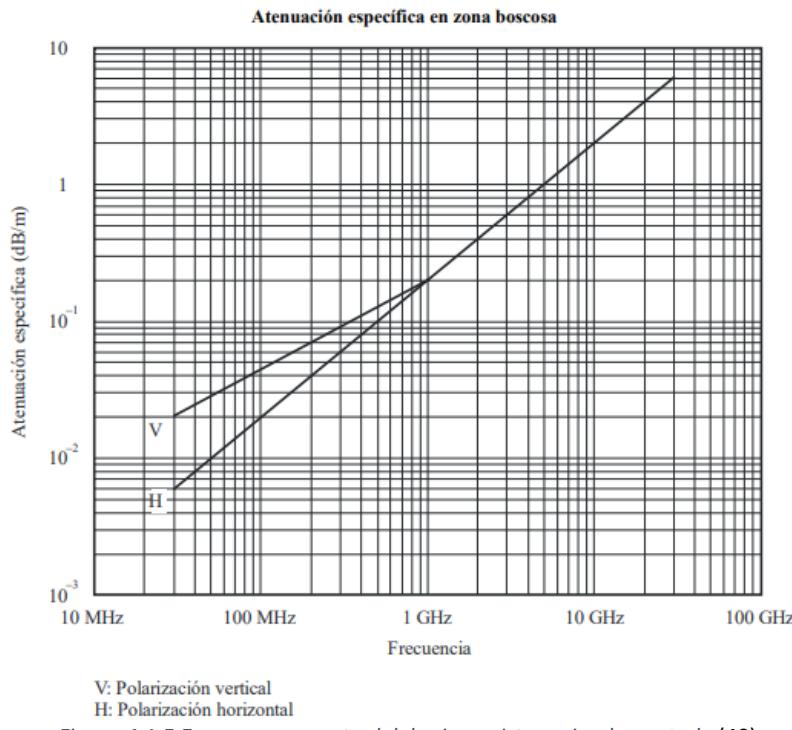


Figura 4.1.5 Esquema conceptual del primer sistema implementado (40)

Con toda la información recopilada para realizar el balance de enlace, se procede a utilizar la ecuación de Friis que se describe a continuación:

$$\begin{aligned}
 P_r(dBm) = & P_t(dBm) + D_t(dBi) + \eta_t(dB) + D_r(dBi) + \eta_r(dB) \\
 & + 20 \log_{10} \left(\frac{\lambda}{4\pi R} \right) (dB) + 10 \log_{10}(1 - |\Gamma_t|^2) (dB) \\
 & + 10 \log_{10}(1 - |\Gamma_r|^2) (dB) + 10 \log_{10}(|\hat{e}_t \hat{e}_r^*|^2) (dB) \\
 & - 1.2R \frac{dB}{km} - 0.035R \frac{dB}{km} - 0.09R \frac{dB}{km}
 \end{aligned}$$

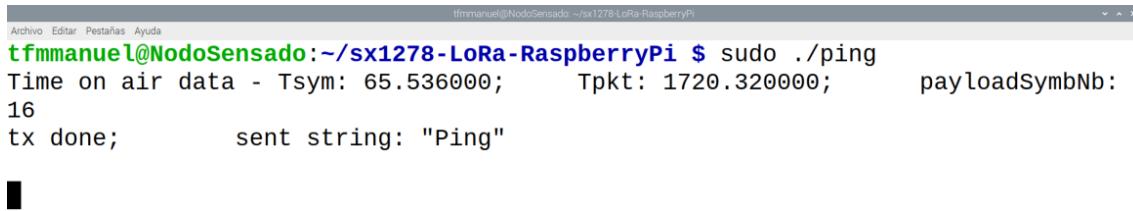
Se puede observar que esta fórmula no tiene una solución algebraica que se pueda despejar en términos de funciones elementales, por tanto, se hizo uso de la herramienta SW *Matlab* para realizar un código que despejara el radio de cobertura R ([apéndice C](#)). Con estos datos se obtiene un rango de cobertura de 29,83 km, por ende, es recomendable situar los nodos sensores a una distancia menor de la indicada respecto a los nodos centrales.

Una vez conectados sendos módulos al nodo de control y de medida, se comprobó que el sistema implementado transmitía y recibía la información de la forma esperada. El repositorio tiene dos códigos para comprobar la conectividad entre dos nodos que se muestran en el [apéndice D](#), el código *ping.c* se encarga de transmitir la trama “*ping*” y el código *pong.c* se encarga de esperar a recibir este mensaje. Además, estos códigos proporcionan diversos datos útiles para entender el estado de la comunicación, como el

TRANSMISIÓN Y RECEPCIÓN DE LOS DATOS

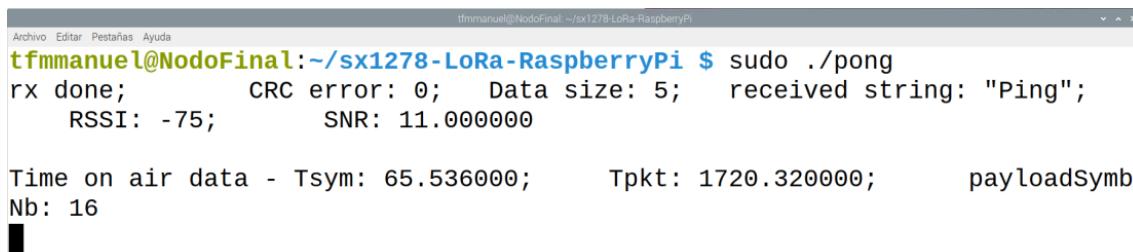
tiempo de símbolo, el tiempo de paquete, la relación señal a ruido y la potencia de la señal recibida, entre otros.

Se procede a realizar una prueba, ejecutando el código *ping.c* en el nodo sensor y el código *pong.c* en el nodo central, como se muestra en la figura 4.1.6 y 4.1.7 el sistema se comunica de la forma esperada.



```
tfmanuel@NodoSensado:~/sx1278-LoRa-RaspberryPi $ sudo ./ping
Time on air data - Tsym: 65.536000;      Tpkt: 1720.320000;      payloadSymbNb:
16
tx done;      sent string: "Ping"
```

Figura 4.1.6 Ejecución del código *ping.c* en el nodo sensor

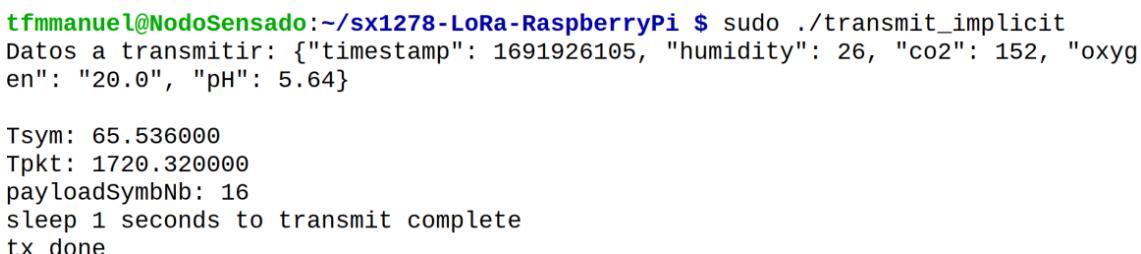


```
tfmanuel@NodoFinal:~/sx1278-LoRa-RaspberryPi $ sudo ./pong
rx done;      CRC error: 0;      Data size: 5;      received string: "Ping";
RSSI: -75;      SNR: 11.000000

Time on air data - Tsym: 65.536000;      Tpkt: 1720.320000;      payloadSymb
Nb: 16
```

Figura 4.1.7 Ejecución del código *pong.c* en el nodo central

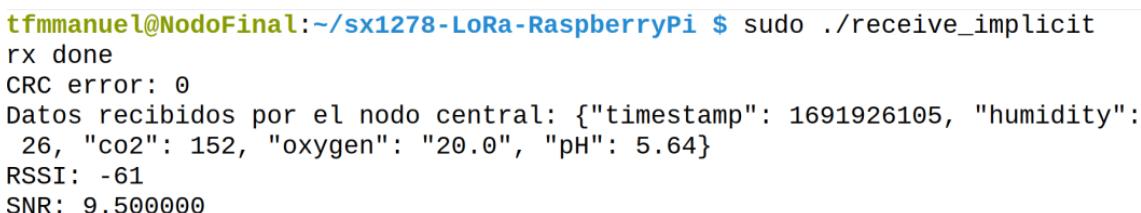
A partir de los códigos *tx_implicit_example.c* y *rx_implicit_example.c* del repositorio, se crearon los códigos *transmit_implicit.c* y *receive_implicit.c* que se utilizarán finalmente para el intercambio de información entre los nodos sensores y el nodo central. Se le añade la funcionalidad de almacenar los datos recibidos en el fichero “*rx_data.json*”, para después transmitirlos a *InfluxDB Cloud*. Estos códigos se ven reflejados en el [apéndice E](#), con ellos se produjo de forma correcta la transmisión de datos como se muestran en las siguientes figuras:



```
tfmanuel@NodoSensado:~/sx1278-LoRa-RaspberryPi $ sudo ./transmit_implicit
Datos a transmitir: {"timestamp": 1691926105, "humidity": 26, "co2": 152, "oxyg
en": "20.0", "pH": 5.64}

Tsym: 65.536000
Tpkt: 1720.320000
payloadSymbNb: 16
sleep 1 seconds to transmit complete
tx done
```

Figura 4.1.8 Ejecución del código *transmit_implicit.c* en el nodo sensor



```
tfmanuel@NodoFinal:~/sx1278-LoRa-RaspberryPi $ sudo ./receive_implicit
rx done
CRC error: 0
Datos recibidos por el nodo central: {"timestamp": 1691926105, "humidity": 26, "co2": 152, "oxygen": "20.0", "pH": 5.64}
RSSI: -61
SNR: 9.500000
```

Figura 4.1.9 Ejecución del código *receive_implicit.c* en el nodo sensor

Las figuras anteriores demuestran que el sistema de comunicación con protocolo LoRa funciona correctamente para la realización de este proyecto.

4.2. Segundo nodo

Como se ha comentado con anterioridad, para este segundo tipo de nodo se pueden hacer uso de dos microcontroladores de *Dragino*. A continuación, se desarrolla la forma en la que se transmiten los datos para cada uno de ellos.

4.2.1. NBSN95

El segundo nodo sensor NBSN95 es de código abierto y está diseñado para la toma de datos al aire libre con un firmware preinstalado. Es un dispositivo de bajo consumo, fiable a la hora de transmitir datos por NB-IoT y eficiente. Se caracteriza por hacer uso del microcontrolador de ultra baja potencia STM32I072. El funcionamiento de este dispositivo se recoge en la figura 4.2.1.

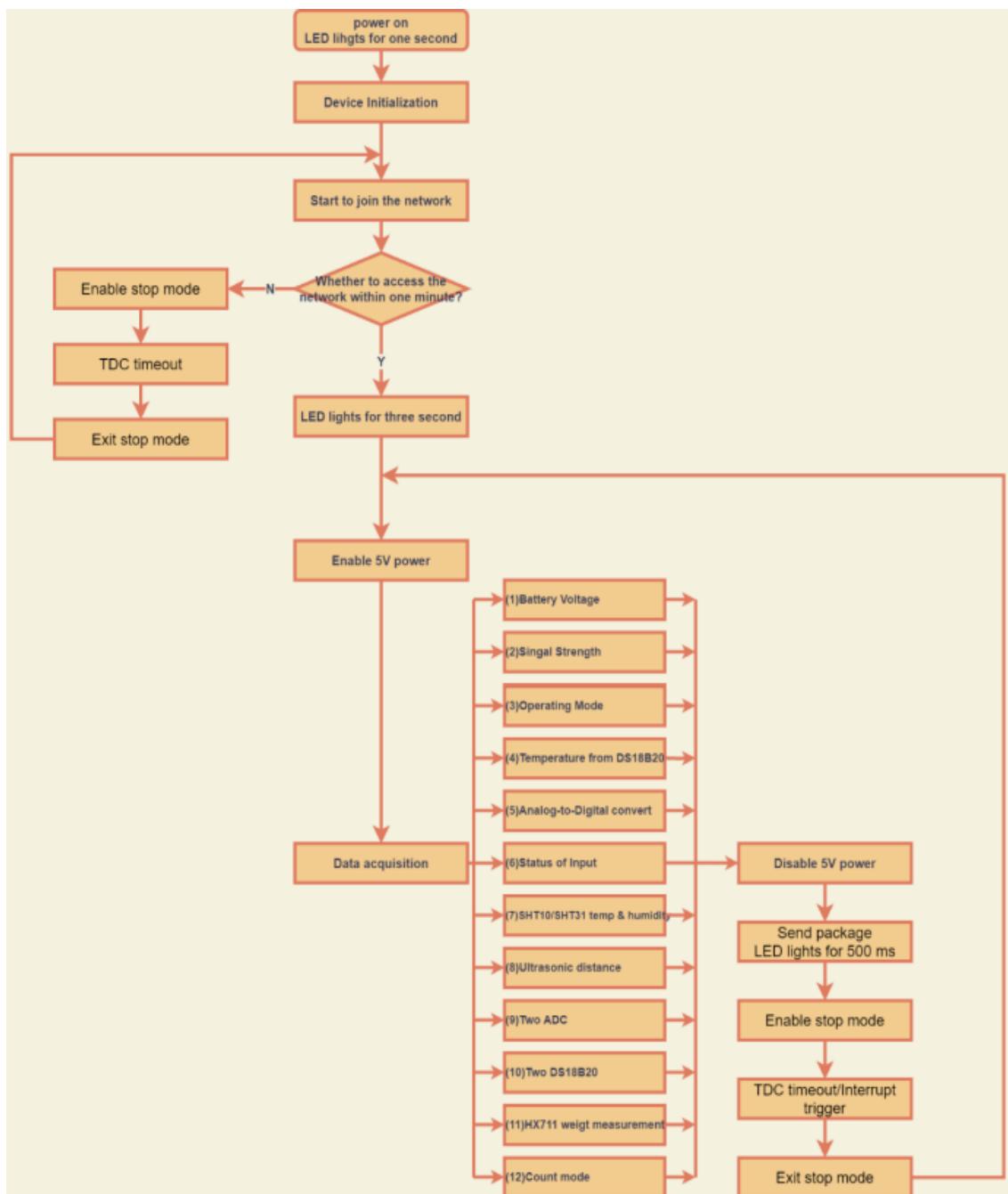


Figura 4.2.1 Funcionamiento del NBSN95 (32)

TRANSMISIÓN Y RECEPCIÓN DE LOS DATOS

Al hacer uso de la GSM, es necesario insertar una tarjeta SIM en el módulo de comunicación, esta tarjeta se usa para comunicación máquina a máquina. Una vez insertada, se debe añadir el módulo de comunicación TTL, como se muestra en la figura 4.2.2, para poder configurarla con la herramienta “Serial Port Utility”.

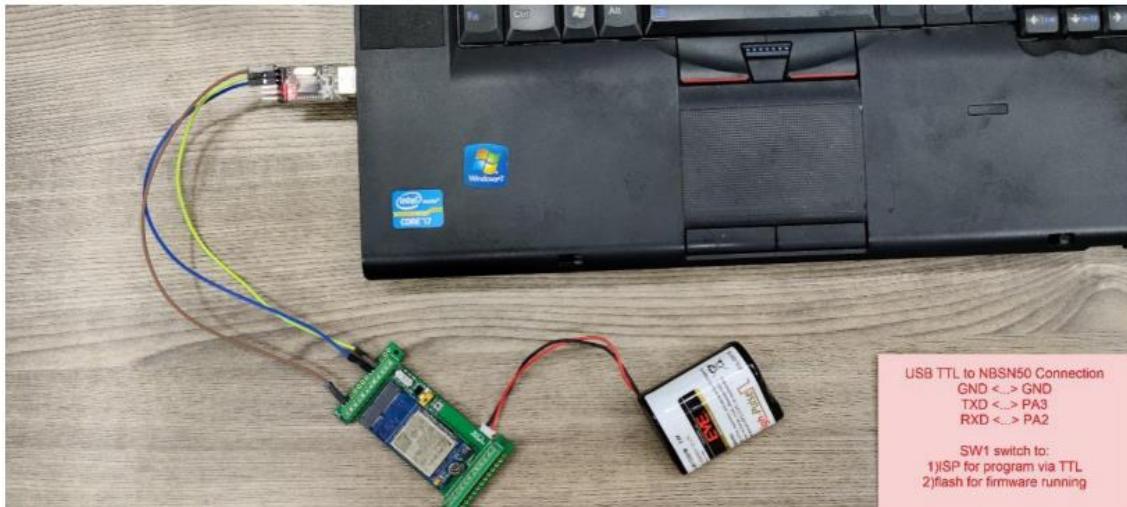


Figura 4.2.2 Conexión de la NBSN95 al PC a través del módulo TTL (32)

Para poder interactuar con el dispositivo hay que introducirle la contraseña. Posteriormente, hay que configurar el protocolo que se quiere utilizar para la transmisión de datos y, a su vez, configurar todos los parámetros necesarios para que esta se produzca.

Para poder cambiar el tiempo entre toma de las medidas, se debe utilizar el comando AT+TDC, a este se le tiene que introducir el tiempo en segundos.

Se procede a explicar los distintos protocolos de comunicación que se pueden utilizar y cómo se ha de configurar los parámetros pertinentes:

- **CoAP:** Es un protocolo de la capa de aplicación de internet con modelo cliente/servidor especificado en la RFC7252 (41), para que dispositivos limitados (pocos recursos, poca capacidad de procesamiento, poca memoria, poco coste...) se comuniquen con cualquier nodo de Internet.
Para configurar el dispositivo con CoAP se introducirán los siguientes comandos a través del “Serial Port Utility”:

```
AT+PRO=2 // Set to use UDP protocol to uplink  
AT+SERVADDR=120.24.4.116,5601 // to set UDP server address and port  
AT+CFM=1 //CoAP = 1
```

- **UDP:** Protocolo que se sitúa entre las capas de red y aplicación estipulado en la RFC768 (42). Altamente utilizado para la transmisión a tiempo real a cambio de no tener un sistema de control de flujo ni de confirmación de entrega.
Para realizar correctamente la configuración del dispositivo es necesario utilizar los siguientes comandos:

TRANSMISIÓN Y RECEPCIÓN DE LOS DATOS

AT+PRO=2 // Set to use UDP protocol to uplink
AT+SERVADDR=120.24.4.116,5601 // to set UDP server address and port
AT+CFM=1 //If the server does not respond, this command is unnecessary

- **MQTT:** Es un protocolo para dispositivos restringidos o para redes con poco ancho de banda, con configuración publicador/suscriptor (43). Es un protocolo que está recomendado su uso en IoT en la ISO/IEC 20922 (44), el cual proporciona un servicio de conexión ordenada, sin pérdidas y bidireccional.
A continuación, se enumeran los comandos necesarios para configurar el dispositivo:

AT+PRO=3 // Set to use MQTT protocol to uplink
AT+SERVADDR=120.24.4.116,1883 // to set MQTT server address and port
AT+CLIENT=CLIENT //Set up the CLIENT of MQTT
AT+UNAME=UNAME //Set the username of MQTT
AT+PWD=PWD //Set the password of MQTT
AT+PUBTOPIC=T1_PUB //Set the sending topic of MQTT
AT+SUBTOPIC=T1_SUB //Set the subscription topic of MQTT

- **TCP:** Protocolo que trabaja junto con IP que asegura un flujo ordenado de datos con configuración cliente/servidor reflejado en la RFC9293 (45). Para asegurar la transmisión de datos se debe establecer una conexión entre el cliente y el servidor, lo que aumenta la latencia del sistema.
Los siguientes comandos configuran al dispositivo para utilizar este protocolo:

AT+PRO=4 // Set to use TCP protocol to uplink
AT+SERVADDR=120.24.4.116,5600 // to set TCP server address and port

La configuración final establecida para subir los datos es la que se muestra a continuación, haciendo uso del protocolo MQTT para ello:

AT+CFGMOD=1 (Comando que selecciona el modo)
AT+SERVADDR=mqtt.vunkers.com,1883
AT+UNAME=gelada
AT+PWD=Aeph9jeeya
AT+PUBTOPIC=vunkers/iot/gelada/
AT+SUBTOPIC=vunkers/iot/gelada/
AT+TDC=300
AT+INTMOD=0
AT+PRO=3
AT+APN=internet.easym2m.eu
AT+CLIENT=NCK<AT+DEUI=?>
AT+CLIENT=NCK867787050462922

TRANSMISIÓN Y RECEPCIÓN DE LOS DATOS

Para que esta configuración se lleve a cabo, se ha de reiniciar el dispositivo, para ello se le puede desconectar y conectar la alimentación o se puede utilizar el comando ATZ.

Para que el presente nodo sensor funcione de manera adecuada se tuvo que integrar el código dado por el fabricante en *Github* (46). La integración del código es compleja y parte de ella está reflejada en el [apéndice F](#).

La cobertura alcanza una distancia máxima de diez kilómetros en entornos abiertos y con buena visibilidad con la estación base (47).

4.2.2. LSN50

Este dispositivo al ser un sistema hermanado con el anterior tiene prácticamente las mismas características, funcionalidades y, además, la configuración es muy similar. Por tanto, la configuración es la indicada en la sección previa, el problema estriba en que no utiliza una estación base GSM para comunicarse, si no que se necesita incluir al sistema un nodo LoRa para dar cobertura al sistema.

Para realizar esta red se ha utilizado un GW *Mikrotik LNOT LR8 kit* de LoRa (48) que se muestra en la figura 4.2.3 junto con la antena 2JW0315-868-C675B (49) mostrada en la figura 4.2.4, para configurar el dispositivo se hace uso de la herramienta *Winbox*.



Figura 4.2.3 Conexión de la NBSN95 al PC a través del módulo TTL (50)



Figura 4.2.4 2JW0315-868-C675B (51)

Para que funcione de forma adecuada, hay que conectarle la antena en el puerto SMA que se encuentra en el lateral de la carcasa y reiniciar el dispositivo pulsando el botón de *reset* mientras se conecta a la red eléctrica.

MikroTik deja a cargo al usuario, a la hora de instalar el GW, de que la instalación se haga de forma correcta. Tiene de base una ganancia de 0 dBi, para cumplir con los requisitos para el rango de frecuencias en torno a los 868 MHz hay que seguir lo indicado en el UN-39 por el Centro Nacional de Atribución de Frecuencias (CNAF).

TRANSMISIÓN Y RECEPCIÓN DE LOS DATOS

ATRIBUCIÓN A LOS SERVICIOS según el RR de la UIT			ATRIBUCIÓN NACIONAL		
460 - 890 MHz			460 - 890 MHz		
Región 1	Región 2	Región 3		USOS	OBSERVACIONES
790 - 862 FIJO MOVIL, salvo móvil aeronáutico 5.316B 5.317A RADIODIFUSIÓN 5.312 5.319			790 - 862 FIJO MOVIL, salvo móvil aeronáutico P P	5.316B 5.317A Sistemas terrenales capaces de prestar servicios de comunicaciones electrónicas (790-823/832-862 MHz) UN-151: Dispositivos PMSE (823-832 MHz) UN-153, UN-154	
862 - 890 FIJO MOVIL, salvo móvil aeronáutico 5.317A RADIODIFUSIÓN 5.322 5.319 5.323	806 - 890 FIJO MÓVIL 5.317A RADIODIFUSIÓN 5.317 5.318		862 - 890 FIJO MOVIL, salvo móvil aeronáutico •	5.317A 5.322 Sistemas terrenales capaces de prestar servicios de comunicaciones electrónicas (860-890 MHz) UN-39, UN-40, UN-41, UN-111 UN-115 UN-118: MICRÓFONOS SIN HILOS UN-135 RFID, UN-154 * Usos M y C (según notas UN)	

Figura 4.2.5 Características propias para el rango de frecuencias requerido del CNAF (34)

Se observa en la figura 4.2.6 con el “Serial Port Utility” los rangos de frecuencias a los que se intenta conectar el LSN50, en este caso son relevantes las que reciben, puesto que son las del GW:

```
***** UpLinkCounter= 0 *****
TX on freq 868.100 MHz at DR 5
txDone
RX on freq 868.100 MHz at DR 5
rxTimeOut
RX on freq 869.525 MHz at DR 0
rxTimeOut

***** UpLinkCounter= 0 *****
TX on freq 868.300 MHz at DR 5
txDone
RX on freq 868.300 MHz at DR 5
rxTimeOut

RX on freq 869.525 MHz at DR 0
rxTimeOut
```

Figura 4.2.6 Frecuencias utilizadas por el GW

Acudiendo al apartado UN-39 del CNAF, se observa que, para dispositivos de baja potencia a las frecuencias vistas en la imagen anterior, la potencia radiada aparente máxima es de 25 mW o 500 mW. Esto se muestra en la siguiente figura:

TRANSMISIÓN Y RECEPCIÓN DE LOS DATOS

UN-39	Banda 868-870 MHz
-------	-------------------

Aplicaciones de baja potencia con la consideración de uso común en el rango de frecuencias 868 a 870 MHz. (Ver figura 24).

Esta banda se destina para aplicaciones de baja potencia y de datos en general de acuerdo con la Decisión de Ejecución (UE) 2019/1345 de la Comisión, por la que se modifica la Decisión 2006/771/CE, y se actualizan las condiciones técnicas armonizadas en el ámbito del uso del espectro radioeléctrico para los dispositivos de corto alcance, así como la Recomendación 70-03 (anexos 1 y 7) de la CEPT, conforme a la siguiente clasificación de dispositivos.

Dispositivos de baja potencia no específicos:

▪ 868,000 - 868,600 MHz con 25 mW (p.r.a.) de potencia radiada aparente máxima.
Estos dispositivos deberán utilizar técnicas de acceso y mitigación de interferencias con rendimiento al menos equivalente a las técnicas descritas en las normas armonizadas según la Directiva 2014/53/UE, o alternativamente no sobrepasar el 1% de ciclo de trabajo.
▪ 868,700 - 869,200 MHz con 25 mW (p.r.a.) de potencia radiada aparente máxima.
Estos dispositivos deberán utilizar técnicas de acceso y mitigación de interferencias con rendimiento al menos equivalente a las técnicas descritas en las normas armonizadas según la Directiva 2014/53/UE, o alternativamente no sobrepasar el 0,1% de ciclo de trabajo.
▪ 869,400 - 869,650 MHz con 500 mW (p.r.a.) de potencia radiada aparente máxima.
Estos dispositivos deberán utilizar técnicas de acceso y mitigación de interferencias con rendimiento al menos equivalente a las técnicas descritas en las normas armonizadas según la Directiva 2014/53/UE, o alternativamente no sobrepasar el 10% de ciclo de trabajo.
▪ 869,700 - 870,000 MHz con 5 mW (p.r.a.) de potencia radiada aparente máxima.
Se permiten aplicaciones de voz con técnicas de mitigación avanzadas, excluyéndose otras aplicaciones de audio y de video.
▪ 869,700 - 870,000 MHz con 25 mW (p.r.a.) de potencia radiada aparente máxima.
Estos dispositivos deberán utilizar técnicas de acceso y mitigación de interferencias con rendimiento al menos equivalente a las técnicas descritas en las normas armonizadas según la Directiva 2014/53/UE, o alternativamente no sobrepasar el 1% de ciclo de trabajo.

Figura 4.2.7 Potencia radiada aparente para las frecuencias utilizadas (34)

Ahora es necesario hacer saber la potencia transmitida por el GW de LoRa, como se ve en la siguiente imagen la potencia máxima transmitida varía entre:

$$P_{tmin} = 21 \text{ dBm} - 2.7 \text{ dB} = 18.3 \text{ dBm}$$

$$P_{tmax} = 21 \text{ dBm} + 2.7 \text{ dB} = 23.7 \text{ dBm}$$

WLAN / Bluetooth / LTE

(EN) Operating Frequency / Maximum output power (DE) Betriebsfrequenz / maximale Ausgangsleistung (FR) Fréquence de fonctionnement / puissance de sortie maximale (IT) Frequenza operativa / massima potenza di uscita (ES) Frecuencia de funcionamiento / potencia de salida máxima (RU) Рабочая частота / максимальная выходная мощность	WLAN Bluetooth LTE FDD Band 1 LTE FDD Band 3 LTE FDD Band 8 LTE FDD Band 20 LTE FDD Band 28	2400-2483.5 MHz / 20 dBm 2400-2483.5 MHz / 20 dBm 2100 MHz / 21 dBm ± 2.7 dB 1800 MHz / 21 dBm ± 2.7 dB 900 MHz / 21 dBm ± 2.7 dB 800 MHz / 21 dBm ± 2.7 dB 700 MHz / 21 dBm ± 2.7 dB
--	--	--

Figura 4.2.8 Potencias máximas transmitidas por el GW (48)

TRANSMISIÓN Y RECEPCIÓN DE LOS DATOS

Antes de calcular si la ganancia de la antena es factible, se proceden a recopilar los datos de la potencia transmitida y radiada, tanto para el máximo como para el mínimo:

$$\begin{aligned} P_t &\rightarrow \text{Potencia transmitida por la antena del GW} \\ \text{Potencia transmitida mínima: } P_{t\min}(dBm) &= 18.3 dBm \\ \text{Potencia transmitida máxima: } P_{t\max}(dBm) &= 23.7 dBm \\ P_{ra} &\rightarrow \text{Potencia radiada aparente} \\ \text{Potencia radiada aparente mínima: } P_{r\min}(mW) &= 25 mW \\ \text{Potencia radiada aparente máxima: } P_{r\max}(mW) &= 500 mW \end{aligned}$$

Se procede a obtener la expresión de la ganancia (G_t) de la antena del GW de LoRa para poder calcular los valores para los casos máximos y mínimos:

$$\begin{aligned} \left\{ \begin{array}{l} P_{ra}(dBm) = 10 \cdot \log(P_{ra}(mW)) \\ PIRE(dBm) = P_{ra}(dBm) + 2.15 dB \rightarrow \\ PIRE(dBm) = P_t(dBm) + G_t(dBi) \end{array} \right. \\ P_t(dBm) + G_t(dBi) = 10 \cdot \log(P_{ra}(mW)) + 2.15 dB \rightarrow \\ G_t(dBi) = 10 \cdot \log(P_{ra}(mW)) + 2.15 dB - P_t(dBm) \end{aligned}$$

- **Caso 1:** Para el mejor caso, potencia radiada aparente máxima y potencia transmitida mínima:

$$G_t(dBi) = 10 \cdot 500 mW + 2.15 dB - 18.3 dBm = 10.8379 dBi > 0 dBi$$

Se cumple

- **Caso 2:** Potencia radiada aparente máxima y potencia transmitida máxima:

$$G_t(dBi) = 10 \cdot 500 mW + 2.15 dB - 23.7 dBm = 5.4397 dBi > 0 dBi$$

Se cumple

- **Caso 3:** Potencia radiada aparente mínima y potencia transmitida mínima:

$$G_t(dBi) = 10 \cdot 25 mW + 2.15 dB - 18.3 dBm = -2.1706 dBi \not> 0 dBi$$

No se cumple

- **Caso 4:** Para el peor caso, potencia radiada aparente mínima y potencia transmitida máxima:

$$G_t(dBi) = 10 \cdot 25 mW + 2.15 dB - 23.7 dBm = -7.5706 dBi \not> 0 dBi$$

No se cumple

Como no se cumple para los casos en los que se tiene una potencia radiada aparente se ha de cambiarlo en el Winbox. Para ello, hay que desactivar el GW y modificar el apartado de “Antenna Gain” introduciendo una ganancia de la antena de -8 dBi, al solamente poder utilizar enteros para la ganancia. Finalmente, se vuelve a activar y el sistema funciona correctamente.

Es necesario estudiar el rango de cobertura que tendría el presente nodo al utilizar la tecnología LoRa, para ello se realizó el mismo análisis que en la sección anterior, pero modificando los valores de los parámetros en función del dispositivo de *Dragino* (33), GW(48) y antena (49).

TRANSMISIÓN Y RECEPCIÓN DE LOS DATOS

Símbolo de la medida	Nombre	Valor	Unidades
F	Frecuencia	868	MHz
Λ	Longitud de onda	345,62	mm
P_t	Potencia transmitida	20	dBm
P_r	Sensibilidad del Receptor	-148	dBm
G_t	Ganancia de la antena Transmisora	-2,8	dBi
G_r	Ganancia de la antena Receptora	-8	dBi
ROE_t	Relación de onda estacionaria de la antena Transmisora	1,70	V/V
ROE_r	Relación de onda estacionaria de la antena Receptora	1,70	V/V
$\rho_t = \frac{ROE_t - 1}{ROE_t + 1}$	Coeficiente de reflexión de la antena Transmisora	0,26	V/V
$\rho_r = \frac{ROE_r - 1}{ROE_r + 1}$	Coeficiente de reflexión de la antena Receptora	0,26	V/V
$1 - \rho_t^2$	Pérdidas por desadaptación de la antena Transmisora	0,93	V ² /V ²
$1 - \rho_r^2$	Pérdidas por desadaptación de la antena Receptora	0,93	V ² /V ²
$ \hat{e}_t \hat{e}_r^* ^2$	Pérdidas por polarización	0,9	
L_{rg}	Pérdidas por Ruido Galáctico	0,90	dB/km
L_{rf}	Pérdidas por Ruido de Fondo	0,035	dB/km
L_{zb}	Pérdidas por Zona Boscosa	0,18	dB/km

Tabla 4.2.1 Datos de las antenas y módulos de comunicación para ver el balance de enlace del segundo nodo

Además, se modificaron los parámetros de las pérdidas para estos nuevos parámetros haciendo uso de las gráficas de las figuras 4.1.4 y 4.1.5. La nueva ecuación de Friis es la siguiente:

$$\begin{aligned}
 P_r(dBm) = & P_t(dBm) + D_t(dBi) + \eta_t(dB) + D_r(dBi) + \eta_r(dB) \\
 & + 20 \log_{10} \left(\frac{\lambda}{4\pi R} \right) (dB) + 10 \log_{10}(1 - |\Gamma_t|^2) (dB) \\
 & + 10 \log_{10}(1 - |\Gamma_r|^2) (dB) + 10 \log_{10}(|\hat{e}_t \hat{e}_r^*|^2) (dB) \\
 & - 0.9R \frac{dB}{km} - 0.035R \frac{dB}{km} - 0.18R \frac{dB}{km}
 \end{aligned}$$

Haciéndose uso del código de Matlab expuesto en el [apéndice C](#), se obtiene un rango de cobertura de 31,36 km, similar al obtenido en el primer nodo.

4.3. Tercer nodo

Este nodo se caracteriza por utilizar las funcionalidades de base que posee la estación meteorológica. Esta se puede conectar a través del Wifi a Internet, también posee la opción de ser configurada de tal forma que se consigan transmitir las medidas a diferentes BD. Por tanto, el nodo central asociado es un *router* y su rango de cobertura no será superior a los noventa metros.

La estación cuenta con la pantalla de la figura 4.3.1 donde se reflejan de forma periódica las medidas, esta será un complemento, pero no aporta prácticamente nada al proyecto. Para configurar correctamente la estación meteorológica es necesario descargar la aplicación móvil “WSView Plus”.

TRANSMISIÓN Y RECEPCIÓN DE LOS DATOS



Figura 4.3.1 Pantalla de la estación meteorológica WS3500

Una vez descargada, es necesario lograr la conexión entre la aplicación y la estación meteorológica. Para ello se ha de añadir una nueva estación a la aplicación, elegir el tipo de estación que se va a utilizar, pulsar los botones “Wind” y “Preassure” de la pantalla y, de esta manera, se logra establecer la conexión.

Para que se transmitan correctamente los datos, se ha de enlazar la estación meteorológica con las distintas BD disponibles como son “Ecowitt”, “Wunderground”, “Weathercloud” y “WeatherObservationsWebsite”. Para ello hay que crearse una cuenta en cada una de estas BD, estas proporcionarán un identificador y una clave que servirán para enlazar la estación meteorológica.

A parte de poder enviarlo a todas las BD expuestas anteriormente, también se puede utilizar la herramienta SW “Home Assistant” para poder almacenar los datos en la Raspberry y poderlos tratar. Para ello, en la aplicación “WSView Plus” se va a la pestaña *Customized* y se procede a añadir la IP de la Raspberry con la ruta para integrar el dispositivo indicada por “Home Assistant” junto con el puerto 8123 y el intervalo en el que se desea enviar los datos (el mínimo son 16 segundos). Todo esto se refleja en la siguiente figura:

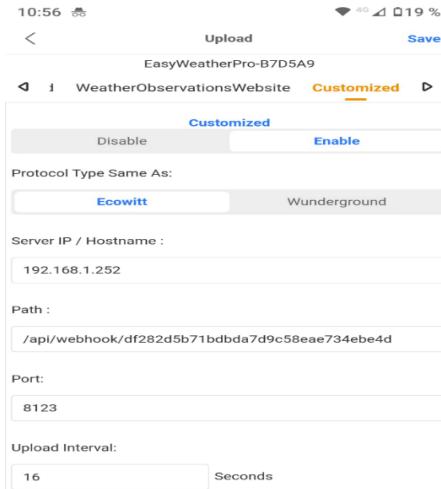


Figura 4.3.2 Configuración para el “Home Assistant”

Esto supondría la configuración de la estación meteorológica para lograr transmitir de manera adecuada los datos a las diferentes BD vistas y a la *Raspberry Pi 4*.

5. Procesado y almacenamiento de datos

En este apartado se va a explicar cómo una vez recogida la información por el nodo central, este la transmite a la BD para procesarla y almacenarla. Se van a observar los distintos sistemas implementados para este nodo, comprobando que la presente red está conformada por diversos modos de trabajo. Esto último se debe a que se quiere presentar una red con un amplio abanico de posibilidades mostrando la versatilidad, modularidad y complejidad que tienen los sistemas IoT.

5.1. Primer nodo

Una vez se reciben los datos del nodo central a través del protocolo implementado y visto en el [apartado 4.1](#), es necesario almacenarlos en alguna BD de secuencias temporales, de esta forma se tendrá un modo de observar la evolución del sistema en estudio a lo largo del tiempo.

Para este primer nodo se hará uso de la BD *InfluxDB* (52), desarrollada por *InfluxData*, la cual está diseñada para el procesamiento y almacenamiento de series temporales para IoT, utilizando un lenguaje similar a SQL.

Se ha de recordar que los datos recibidos por el nodo central se almacenan de forma temporal en el fichero “*rx_data.json*”. Este procedimiento se realiza principalmente para segmentar el sistema, teniendo un código en C que se encarga de la recepción de los datos medidos por el nodo sensor para almacenarlos temporalmente en el archivo JSON y un código en Python para leer este fichero y transmitir la información a la BD *InfluxDB*.

Para el desarrollo de este proyecto se hizo uso de *InfluxDB Cloud*, de esta forma se tienen los datos disponibles en la nube de *Amazon Web Services*. Una vez creada una cuenta, elegida la región y creada la organización, se ha de crear un *bucket* (*Control_forestal_v0*) donde almacenar los datos y enlazarlo con un *token* (*Token_TFM*) de escritura y lectura como se muestra en las figuras 5.1.1 y 5.1.2.

Para aclarar lo dicho previamente, en *InfluxDB* un *bucket* (53) es una ubicación perteneciente a una organización donde se recopilan los datos de series temporales durante un periodo de retención, esto último quiere decir que se eliminan los datos previos al tiempo establecido. De la misma manera, un *token* (54) sirve para enlazar los datos con los usuarios, además de establecer los permisos que tiene *InfluxDB* para con estos datos.

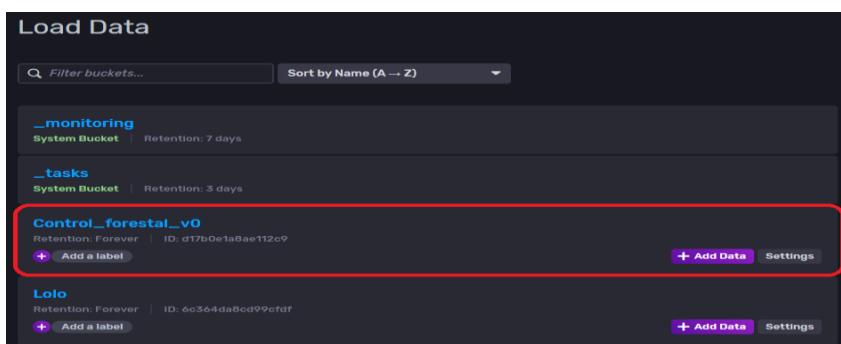


Figura 5.1.1 Bucket utilizado en la BD InfluxDB

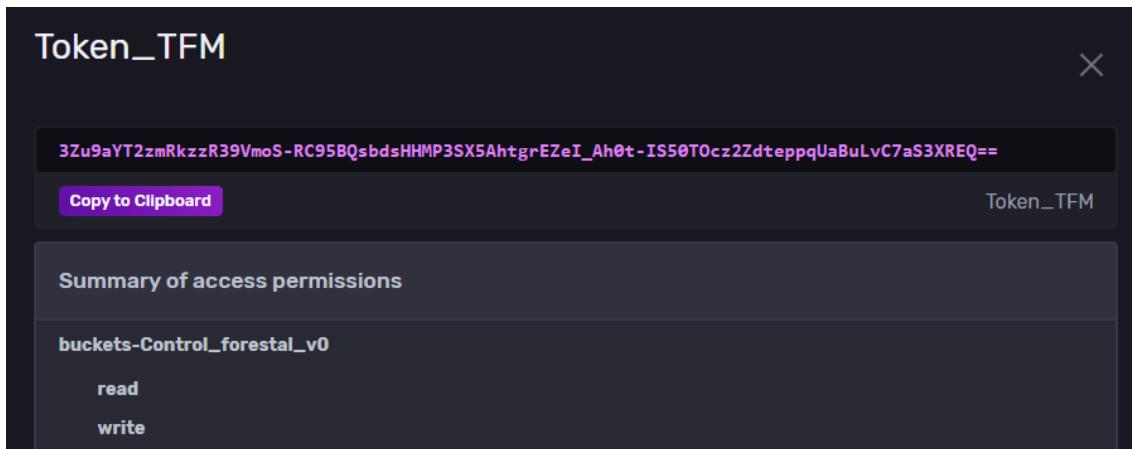


Figura 5.1.2 Token que enlaza el bucket junto con los permisos

Si hubiese cualquier inconveniente con el desarrollo en la nube de esta BD, se podría realizar de la misma manera una BD con *InfluxDB* de forma local para tener los datos almacenados en una estación física.

El código desarrollado en el nodo central para este propósito es el que se muestra en el [apéndice G](#), se procede a explicar las secciones de código para comprender los distintos pasos necesarios para adecuar la información y subirla a la nube:

- Primero se han de importar las librerías necesarias para llevar a cabo todo lo que se va a describir a continuación.

```
import pandas as pd
import requests
import time
import json
```

- Es necesario adecuar el fichero para que sea verdaderamente un *JSON* añadiéndole al principio y al final unos corchetes. Posteriormente, se leen estos datos como un *JSON* y se eliminan de la memoria, cuando pasen 20 minutos, al recibir nuevos datos, se repetirá el proceso.

```
# Adecua fichero y lo lee como JSON
with open('rx_data.json', 'r') as f:
    aux = f.read()
    aux = '[\n\r' + data.replace('\n', ',')[:-1] + '\n\r]\0' # Agregar
corchetes y eliminar la última coma
    data = json.load(aux)

# Se eliminan los datos
with open('rx_data.json', 'w') as json_empty:
    json_empty.write('')
```

- Se empieza a generar la cabecera y el cuerpo de la consulta a la BD para transmitir los datos. Para ello se solicitan distintos parámetros como el

PROCESADO Y ALMACENAMIENTO DE DATOS

identificador y el nombre de la organización, el nombre del *bucket*, la clave del *token*, y la URL.

```
# Se establece los valores necesarios para la conexión:  
INFLUX_TOKEN = '3Zu9aYT2zmRkzzR39VmoS-RC95BQsbdsHHMP3SX5AhtgrEZeI_Ah0t-  
IS50T0cz2ZdteppqUaBuLvC7aS3XREQ=='  
ORG ='Lolo'  
ORGID = 'f386e6a8f9501a3a'  
INFLUX_CLOUD_URL='https://europe-west1-1.gcp.cloud2.influxdata.com'  
BUCKET_NAME ='Control_forestal_v0'c  
  
# Establecemos la precisión en MILISEGUNDOS:  
QUERY_URI='{}/api/v2/write?org={}&orgID={}&bucket={}&precision=ms'.format  
(INFLUX_CLOUD_URL,ORG,ORGID,BUCKET_NAME)  
  
# Se establece la cabecera con el token correspondiente  
headers = {}  
headers['Authorization'] = 'Token {}'.format(INFLUX_TOKEN)  
measurement_name = 'nodo'
```

- Finalmente, se realiza un bucle para enviar múltiples consultas donde varían los datos de medición.

```
# Se leen todas las filas del CSV y se van enviando a InfluxDB  
ID_Nodo = 1  
i=0  
print (data)  
for row in data:  
    i=i+1  
    timestamp = str(int(row['timestamp']*1000))  
    Humidity = str(row['humidity'])  
    Leaf_Oxygen = str(row['oxygen'])  
    Leaf_CO2 = str(row['co2'])  
    PH_value = str(row['pH'])  
  
    dataE = []  
    dataE.append('{measurement},id_nodo={id_node}  
humedad_hoja={Hum},oxigeno_hoja={leaf_oxi},co2_hoja={leaf_co2},ph={ph_val}  
{ts}'.  
format(measurement=measurement_name,id_node=ID_Nodo,Hum=Humidity,leaf_ox  
i=Leaf_Oxygen,leaf_co2=Leaf_CO2,ph_val=PH_value,ts=timestamp))  
  
    dataEnvio = '\n'.join(dataE)  
    r = requests.post(QUERY_URI, data=dataEnvio, headers=headers)  
    time.sleep(0.0002)  
    print(i)
```

PROCESADO Y ALMACENAMIENTO DE DATOS

Una vez realizado el código, el sistema comienza la toma de datos del entorno en estudio. Se puede observar parte de la información de la humedad de hoja que se ha ido recopilando en el *bucket* en la figura 5.1.3.

false	false	true	true	true
dateTime:RFC3339	double	string	string	string
_time	_value	_field	_measurement	id_nodo
2023-08-14T16:49:51.450Z	19.783783783783782	humedad_hoja	nodo	0
2023-08-14T18:34:01.452Z	19.648604269293923	humedad_hoja	nodo	0
2023-08-14T20:18:11.454Z	22.545426170468186	humedad_hoja	nodo	0
2023-08-14T22:02:21.456Z	19.92798690671031	humedad_hoja	nodo	0
2023-08-14T23:46:31.458Z	14.727858293075684	humedad_hoja	nodo	0
2023-08-15T10:11:31.470Z	9.612850082372322	humedad_hoja	nodo	0
2023-08-15T11:55:41.472Z	10.731078904991948	humedad_hoja	nodo	0
2023-08-15T13:39:51.474Z	12.788687397708673	humedad_hoja	nodo	0
2023-08-15T15:24:01.476Z	18.74098360655738	humedad_hoja	nodo	0
2023-08-15T17:08:11.478Z	19.884552845528454	humedad_hoja	nodo	0
2023-08-15T18:52:21.480Z	9.0625	humedad_hoja	nodo	0
2023-08-15T20:36:31.482Z	19.085899513776337	humedad_hoja	nodo	0
2023-08-15T22:20:41.484Z	19.052459016393442	humedad_hoja	nodo	0
2023-08-16T00:04:51.486Z	17.914614121510474	humedad_hoja	nodo	0

Figura 5.1.3 Parte de la toma de datos que recoge InfluxDB Cloud

5.2. Segundo nodo

Se han de estudiar los datos recibidos para los dos dispositivos de *Dragino*, el NBSN95 que hace uso del protocolo de NB-IoT y la LSN50 que utiliza LoRa. Sin embargo, al transmitir las tramas con una estructura idéntica de datos en estos dos tipos de dispositivos de medición, el procesamiento y tratamiento de los datos será idéntico.

Para el correcto almacenamiento de los datos en el segundo nodo se emplea la BD de secuencias temporales *The Things Network* (55). Esta BD realmente es una iniciativa comunitaria abierta para realizar una red colaborativa de bajo consumo y de área amplia. Además, ofrecen herramientas para el diseño de aplicaciones.

Una vez enviado el dato por el nodo sensor al GW, este último lo transmitirá a la BD, en forma de una trama hexadecimal. En el interior de esta se alberga la información captada por los sensores en los bits, por ende, se han de procesar para adecuarlos a la presente BD. Para ello, se utiliza un código en *JavaScript* que transforma los bits correspondientes en los datos de las magnitudes medidas por los sensores.

Será necesario realizar distintos códigos decodificadores que adecúen de forma correcta los datos recibidos por el nodo sensor para cada uno de los modos implementados. A modo de ejemplo, se muestra en el [apéndice H](#) el código utilizado para el procesamiento de la trama en el modo 1 y 8.

Los datos procesados de la trama hexadecimal se recogen en las distintas magnitudes de medida. Dentro del apartado “Dispositivos” en “Última Telemetría” se muestran los datos de los sensores del modo 7, es decir, se pueden observar los datos de las diversas

PROCESADO Y ALMACENAMIENTO DE DATOS

magnitudes de la estación meteorológica, junto con los datos de latitud y longitud del GPS, todo esto en las figuras 5.2.1, 5.2.2 y 5.2.3. Además, se puede observar en la figura 5.2.2 la trama hexadecimal sin procesar.

The screenshot shows a monitoring application's interface. On the left is a sidebar with navigation links: Principal, Alarms, Panels, Solution templates (NEW), Entities, Dispositivos, Activos, Vistas de entidad, Perfiles, Clientes, Usuarios, Integrations center, Integraciones, Conversores de datos, Cadenas de Reglas, and Gestión de Edges. The main area has tabs for All and Groups. Under 'Dispositivos', there is a table with columns 'Fecha de creación' and 'Nombre'. One row is selected, showing '2023-08-31 13:22:16' and 'NCK867787052774209'. A modal window titled 'NCK867787052774209' displays 'Detalles del dispositivo' with tabs for Detalles, Atributos, Última telemetría, Alarmas, and Eventos. The 'Última telemetría' tab is active, showing a table of recent measurements. A red circle with a white edit icon is overlaid on the top right of the modal. At the bottom of the modal, there is a pagination control 'Ítems por página: 10' and a link '1 - 10 de 13'.

Figura 5.2.1 Datos recibidos de la estación meteorológica y GPS 1

This screenshot is similar to Figure 5.2.1 but for GPS 2. The sidebar and main navigation are identical. The 'Dispositivos' table shows a different set of device entries. The modal window for 'NCK867787052774209' is open, with the 'Última telemetría' tab selected. The telemetry table includes entries like 'rain' (0), 'rawData' (hex value), 'signal' (18), 'temperature' (37), and 'uv' (19715). A red circle with a white edit icon is overlaid on the top right of the modal. The pagination at the bottom of the modal is 'Ítems por página: 10' and '1 - 10 de 13'.

Figura 5.2.2 Datos recibidos de la estación meteorológica y GPS 2

This screenshot is similar to Figures 5.2.1 and 5.2.2 but for GPS 3. The sidebar and main navigation are identical. The 'Dispositivos' table shows a different set of device entries. The modal window for 'NCK867787052774209' is open, with the 'Última telemetría' tab selected. The telemetry table includes entries like 'wind_burst' (33), 'wind_dir' (86), and 'wind_speed' (0). A red circle with a white edit icon is overlaid on the top right of the modal. The pagination at the bottom of the modal is 'Ítems por página: 10' and '11 - 13 de 13'.

Figura 5.2.3 Datos recibidos de la estación meteorológica y GPS 3

5.3. Tercer nodo

Gracias a las funciones básicas que vienen implementadas en la estación meteorológica WS3500, se puede realizar una transmisión de forma sencilla y fiable de los datos recogidos por esta.

Para ello, como se dijo en el [apartado 4.3](#), se ha de instalar la aplicación *VSView Plus* en un dispositivo móvil. Posteriormente, se enlazará la estación meteorológica con la BD correspondiente haciendo uso del identificador y la clave respectiva.

De esta forma, se registró la estación meteorológica utilizada en este nodo, pero se podrían añadir varias estaciones meteorológicas, de distintas marcas y fabricantes. De manera que se podría escalar y añadir tecnología más avanzada.

En las siguientes figuras se pueden observar algunos de los datos y sus características en las BD *Ecowitt* (figura 5.3.1), *Weather Underground* (figura 5.3.2) y *Weathercloud* (figura 5.3.3).



Figura 5.3.1 Recepción de los datos en la BD Ecowitt

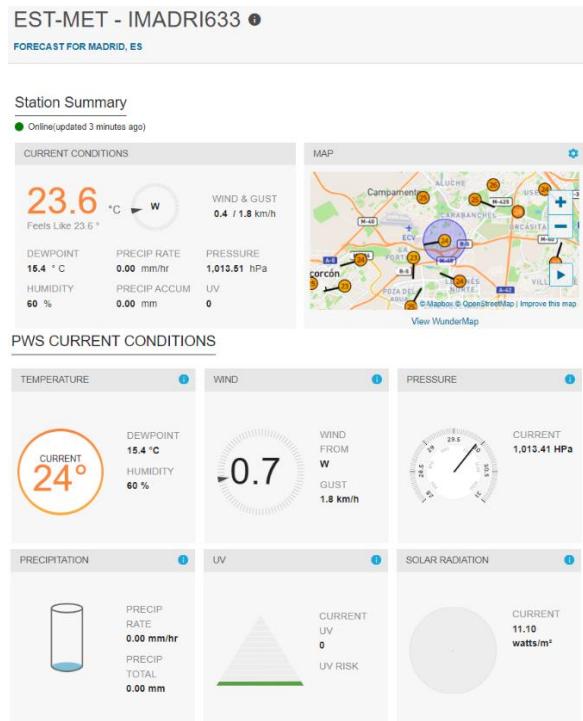


Figura 5.3.2 Recepción de los datos en la BD Weather Underground

PROCESADO Y ALMACENAMIENTO DE DATOS

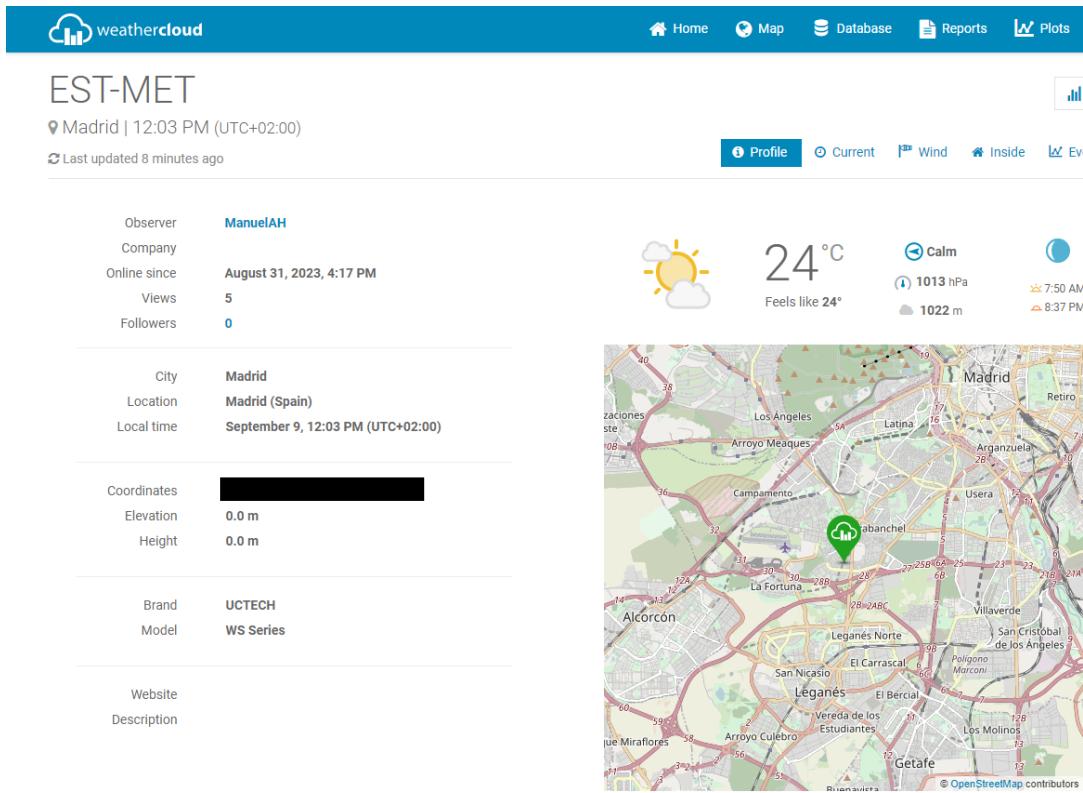


Figura 5.3.3 Recepción de los datos en la BD Weathercloud

A pesar de la simplicidad de este sistema, una de las formas más interesantes y apropiadas para el desarrollo de este proyecto es hacer uso de *Home Assistant* (56) para el recopilado y procesado de datos.

Es necesario explicar que *Home Assistant* es un SW gratuito y de código abierto diseñado para funcionar como un sistema de control para la automatización del hogar con dispositivos inteligentes.

Para la integración de la estación meteorológica WS3500 es preciso realizar una configuración previa. Para ello, es necesario ir a *Ajustes* y en la ventana *Integración* se busca la marca *Ecowitt* debido a que se van a enviar los datos en ese formato, esto se muestra en la figura 5.3.4. Posteriormente, en la figura 5.3.5 se puede observar la configuración que se debía introducir en la aplicación, como se aprecia en la figura 4.3.2.

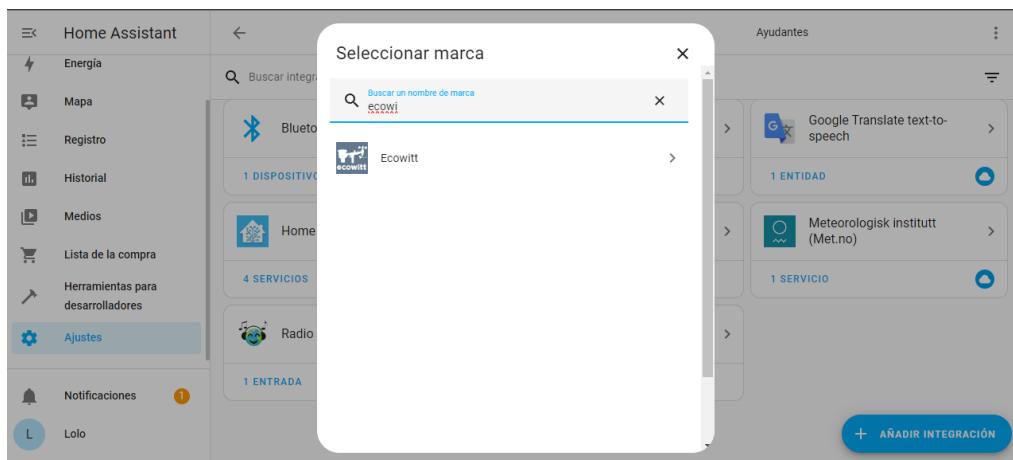


Figura 5.3.4 Integrar estación meteorológica en Home Assistant

PROCESADO Y ALMACENAMIENTO DE DATOS

Éxito!



Para terminar de configurar la integración, usa la aplicación Ecowitt (en tu teléfono) o accede a Ecowitt WebUI en un navegador en la dirección IP de la estación.

Elige tu estación -> Menú Otros -> Servidores de carga de bricolaje. Presiona siguiente y selecciona 'Personalizado'

- IP del servidor: 192.168.1.252
- Ruta: /api/webhook/df282d5b71bdbda7d9c58eae734ebe4d
- Puerto: 8123

Haz clic en 'Guardar'.

Configuración creada para Ecowitt.

[TERMINAR](#)

Figura 5.3.5 Datos a introducir en la aplicación para la integración

Los datos enviados a *Home Assistant* se recogen en una tabla, la cual muestra la última medida tomada por el tercer nodo. Todo lo mostrado anteriormente, demuestra la cantidad y versatilidad de opciones que existen para la toma de magnitudes medioambientales.

	WS2900_V2.01.18 Absolute Pressure	941,0 hPa
	WS2900_V2.01.18 Dewpoint	16,6 °C
	WS2900_V2.01.18 Feels like Temperature	22,8 °C
	WS2900_V2.01.18 Humidity	68%
	WS2900_V2.01.18 Indoor Dewpoint	17,0 °C
	WS2900_V2.01.18 Indoor Humidity	53%
	WS2900_V2.01.18 Indoor Temperature	27,4 °C
	WS2900_V2.01.18 Max Daily Gust	5,4 km/h
	WS2900_V2.01.18 Outdoor Temperature	22,8 °C
	WS2900_V2.01.18 Wind Gust	0,0 km/h
	WS2900_V2.01.18 Wind Speed	0,0 km/h

Figura 5.3.6 Recepción de los datos en la BD Home Assistant

6. Representación gráfica y estudio de las mediciones

Los datos recopilados durante los períodos de medición del entorno en estudio se han de representar para facilitar el posterior estudio de los especialistas. Asimismo, será de gran utilidad hacer uso de herramientas propias para el tratamiento de los datos y la generación de alarmas que indiquen los casos atípicos. Con esto último se podrán indicar los casos de malfuncionamiento del nodo, del sistema o una situación extrema del medio o del ejemplar en estudio. También, se incluye un somero análisis de los resultados inferidos del sistema.

6.1. Gráficas, tratamiento de la información y alertas

El sistema implementado cuenta con diversas BD, las cuales desempeñan un papel fundamental para el estudio del entorno forestal y el análisis del medioambiente que lo influye. La exploración de esta multitud de BD se realiza con el propósito de proporcionar completitud, enriqueciendo al sistema desarrollado y profundizando en las opciones que ofrecen cada una de ellas.

A continuación, se presentarán las principales características de las BD más relevantes para el presente estudio, profundizando en las funcionalidades más útiles para el proyecto.

6.1.1. InfluxDB

InfluxDB no es solo una BD, sino que también es una herramienta SW de gran utilidad para el almacenamiento, procesado y la representación gráfica de los datos obtenidos. Para asegurar el correcto almacenaje de los datos transmitidos, se puede hacer uso de la interfaz *Data Explorer* para buscar en el *bucket* la magnitud que se quiere representar. Todo esto se muestra en las figuras 6.1.1 y 6.1.2.

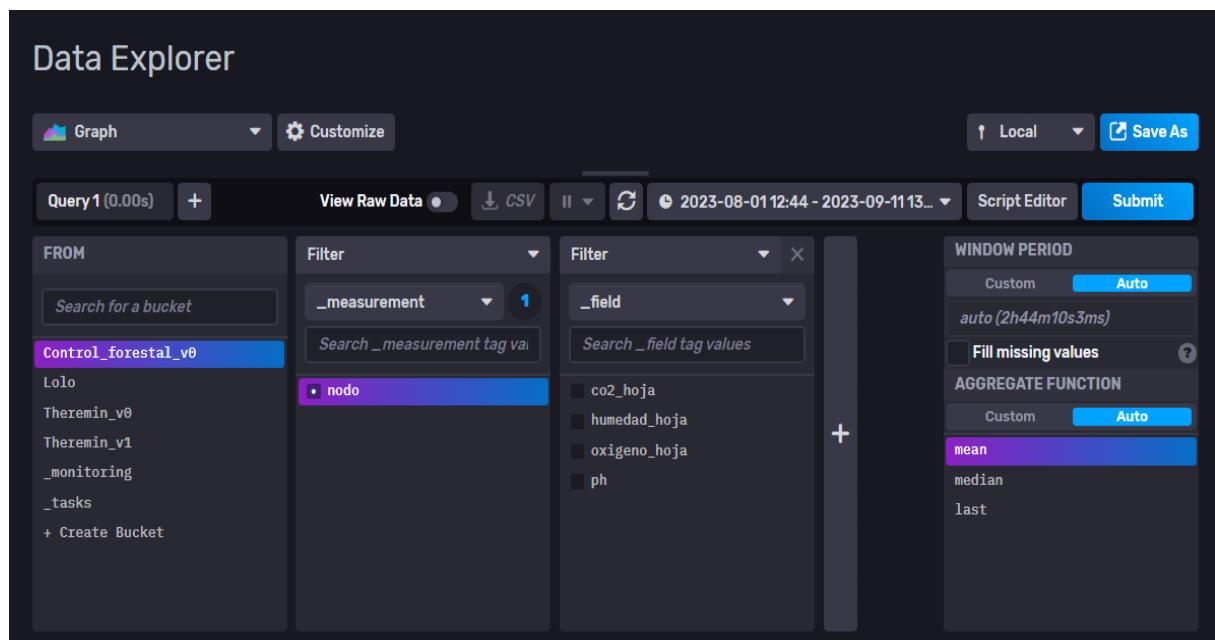


Figura 6.1.1 Explorador de datos

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

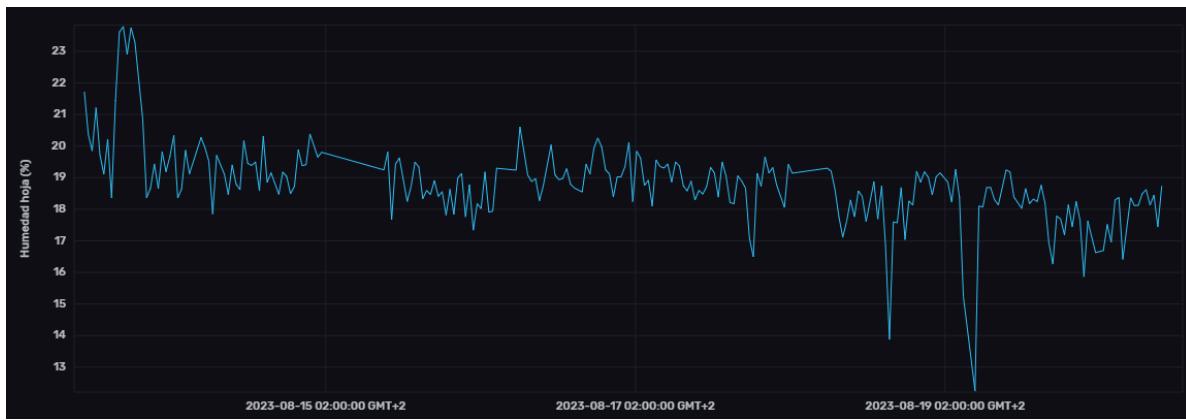


Figura 6.1.2 Datos de la humedad de las hojas

A parte de poder representar cada una de las gráficas, una herramienta fundamental que tiene *InfluxDB* son los *dashboards*. Con esta interfaz se puede observar de una manera clara, interactiva y personalizable los datos recibidos otorgando una facilidad de síntesis al poder realizar una comparación visual de toda la información.

Asimismo, se pueden realizar diversos *dashboards* en función de las necesidades de los especialistas que se encarguen del estudio de los datos. Como se observa en las figuras 6.1.3 y 6.1.4, se creó un panel para graficar las magnitudes que estudian el follaje del ejemplar junto con la media en todo el rango de medición y otro donde se muestran todas las magnitudes estudiadas por el primer nodo.



Figura 6.1.3 Dashboard de los sensores de hoja



Figura 6.1.4 Dashboard de todos los sensores

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

Para representar cada una de estas gráficas y/o datos técnicos se crearon códigos en *Flux* (57), este es el lenguaje funcional desarrollado para poder realizar consultas, análisis y ejecución de los datos y la información asociada y/o desarrollada a partir de estos. Permite a los usuarios interactuar con los datos de manera eficiente en una plataforma potente y versátil.

En la figura 6.1.5 se muestra el código utilizado para representar la humedad de la hoja en el *dashboard* donde se grafican los datos recopilados por los sensores de hoja. Es remarcable que esta es una potente herramienta que permite no solamente representar datos, sino que también sirve para generar nuevas gráficas con la media, mediana, entre otros. Incluso se pueden cargar modelos de predicción para generar gráficas que predigan el funcionamiento del sistema.

The screenshot shows the InfluxDB Flux UI interface. At the top, there's a title 'Humedad hoja (%)'. Below it, a toolbar with 'Graph' (selected), 'Customize', and other buttons like 'View Raw Data', 'CSV', and 'Past 1h'. A central code editor window displays the following Flux query:

```
1 from(bucket: "Control_forestal_v0")
2 //|> range(start: v.timeRangeStart, stop: v.timeRangeStop)
3 |> range(start: 2023-08-13T00:00:00Z, stop: 2023-08-21T00:00:00Z)
4 |> filter(fn: (r) => r["_measurement"] == "nodo")
5 |> filter(fn: (r) => r["_field"] == "humedad_hoja")
6 |> filter(fn: (r) => r["id_nodo"] == "0") // Eliminar para ver todos
7 |> yield(name: "mean")
```

To the right of the code editor, there's a sidebar with sections for 'Transformations' and 'Functions'. The 'Transformations' section contains 'aggregate.rate'. The 'Functions' section lists several functions: 'changeMomentumOscillator', 'columns', 'cov', 'covariance', and 'cumulativeSum'. On the far right, there are tabs for 'Variables' and 'Functions'.

Figura 6.1.5 Script creado para mostrar la humedad de hoja

Es una buena praxis garantizar la integridad del sistema desarrollado. Por lo cual, se han implementado una serie de alertas que, de forma continua, supervisan el funcionamiento. Las alertas serán de utilidad para detectar cualquier anomalía al enviar notificaciones que revelarían un amplio abanico de situaciones, las cuales incluyen desde condiciones anómalas, deterioro del sistema, hasta situaciones extremas como incendios.

En *InfluxDB*, las alertas se gestionan a partir de *checks*, los cuales evalúan los datos y asignan un estado o nivel a cada punto en función de las condiciones. Los estados pueden clasificarse en *OK*, *INFO*, *WARN* y *CRIT*.

Para avalar la correcta operación del sistema, se han utilizado solamente los *checks* del tipo *WARN* y *CRIT*. Los primeros para alertar de situaciones anómalas, pero que no son de gran importancia. Los segundos sirven para proporcionar información sobre casos climáticos extremos, condiciones críticas del ejemplar o malfuncionamiento del sistema, esto se muestra en la figura 6.1.6.

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES



Figura 6.1.6 Check de la magnitud temperatura ambiente con un *WARN* y un *CRIT*

Una funcionalidad de gran utilidad y relevancia es la capacidad de poder transmitir estas alertas a diversas aplicaciones, es fundamental para transmitir sin gran dilación y de forma efectiva los diferentes estados que amenazan al sistema y al entorno en estudio. A través de un “*Notification Endpoint*” se puede realizar una comunicación multicanal con distintos servicios de terceros como *Slack*, *PagerDuty* o, incluso, en un servidor *HTTP* propio.

Para controlar este sistema se va a utilizar la aplicación de *Slack* para enviar la información a los especialistas y técnicos. Al no ser tan simple el enlace entre *InfluxDB* y el espacio de trabajo de *Slack*, se procede a realizar una somera explicación sobre como realizar este enlace. Primero hay que dirigirse a la web y pulsar el botón “*Create an App*” como se muestra en la figura 6.1.7. Posteriormente, se creará una aplicación haciendo uso de Scratch como se muestra en la figura 6.1.8.

Figura 6.1.7 Creación una aplicación para *Slack*

Create an app

X

Choose how you'd like to configure your app's scopes and settings.

From scratch

Use our configuration UI to manually add basic info, scopes, settings, & features to your app.



From an app manifest BETA

Use a manifest file to add your app's basic info, scopes, settings & features to your app.



Need help? Check our [documentation](#), or [see an example](#)

Figura 6.1.8 Elección de opción "From Scratch" para crear la aplicación

Una vez creada la aplicación se ha de enlazar con el *workspace* (*Efficiency Boys*) como se muestra en la figura 6.1.9. Se crea el *webhook* (figura 6.1.10) y el canal (figura 6.1.11).

Name app & choose workspace

X

App Name

Control_forestal_v0

16

Don't worry - you'll be able to change this later.

Pick a workspace to develop your app in:



Efficiency Boys



Keep in mind that you can't change this app's workspace later. If you leave the workspace, you won't be able to manage any apps you've built for it. The workspace will control the app even if you leave the workspace.

[Sign into a different workspace](#)

By creating a Web API Application, you agree to the [Slack API Terms of Service](#).

Cancel

Create App

Figura 6.1.9 Elección de nombre a la aplicación y asignación de workspace

Add features and functionality

Choose and configure the tools you'll need to create your app (or review all our documentation).

Building an internal app locally or behind a firewall?

To receive your app's payloads over a WebSockets connection, enable [Socket Mode](#) for your app.

Incoming Webhooks

Post messages from external sources into Slack.

Interactive Components

Add components like buttons and select menus to your app's interface, and create an interactive experience for users.

Figura 6.1.10 Creación de un webhooks

Crear un canal

X

Los canales son el lugar donde se comunica tu equipo. Funcionan mejor cuando se organizan en torno a un tema, por ejemplo #marketing.

Nombre

control_forestal_iot_v0

57

Descripción (optativa)

Canal que te avisa de Warnings y Crits de InfluxDB

¿De qué trata este canal?

Establecer como canal cerrado

Cuando un canal se convierte en "cerrado", solo pueden verlo o unirse a él quienes reciban una invitación.



Compartir fuera de Efficiency ... ⓘ

PREMIUM

Crear

Figura 6.1.11 Creación de un canal

Posteriormente, se enlaza el *webhook* con el *workspace* y con el canal, como se muestra en las figuras 6.1.12 y 6.1.13.

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

The screenshot shows the 'Incoming Webhooks' section of the Slack app configuration. On the left, a sidebar lists various app settings like Basic Information, Collaborators, and Features. The 'Incoming Webhooks' option is selected and highlighted in blue. The main content area has a title 'Activate Incoming Webhooks' with a green 'on' toggle switch. Below it, a text block explains what incoming webhooks are and how they work. It also notes that adding a bot user is required if none exists. A note states that each app installation generates a new webhook URL. Another note says that deactivating the feature will stop generating URLs for new installations. A 'Webhook URLs for Your Workspace' section follows, with a curl command example for posting messages. A table below shows no webhook URLs added yet, with a button to 'Add New Webhook to Workspace'. The bottom of the page includes links for Slack, Help, Contact, Policies, and Our Blog.

Figura 6.1.12 Asignación del webhook al workspace

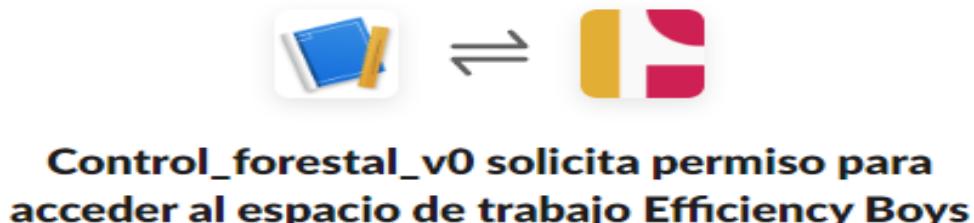


Figura 6.1.13 Asignación del canal al canal de Slack

Para finalizar este procedimiento, se le añade el *Notification Endpoint* para enlazar el canal a la BD *InfluxDB*. Esto último se muestra en las siguientes figuras:

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

Webhook URLs for Your Workspace

To dispatch messages with your webhook URL, send your message in JSON as the body of an `application/json` POST request.

Add this webhook to your workspace below to activate this curl example.

Sample curl request to post to a channel:

```
curl -X POST -H 'Content-type: application/json' --data '{"text":"Hello, World!"}' https://hooks.slack.com/services/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
```

[Copy](#)

Webhook URL	Channel	Added By
https://hooks.slack.com/services/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	#control_forestal_iot_v0	MANUEL ALVAREZ HERRERA Mar 27, 2023

[Add New Webhook to Workspace](#)

Figura 6.1.14 Con el webhook creado se copia el enlace

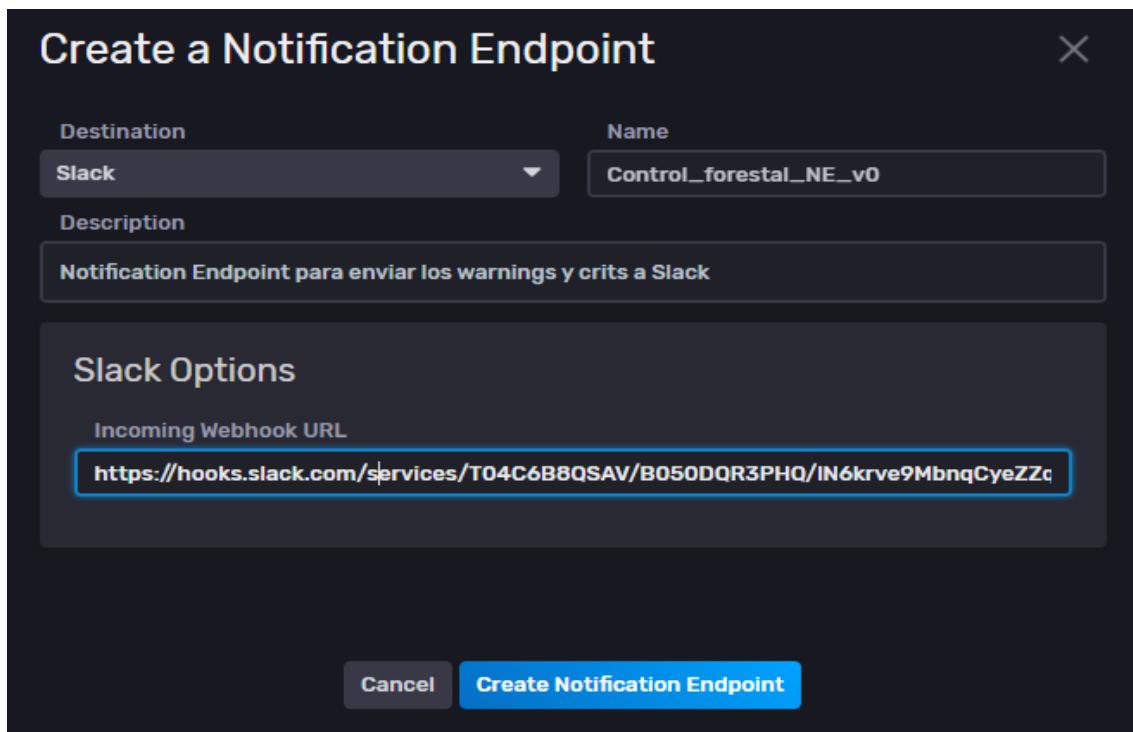


Figura 6.1.15 Creación del Notification Endpoint

Recopilando todo lo visto, se han expuesto las diferentes herramientas que son de utilidad para el estudio, desarrollo y análisis del proyecto, explicando cómo se pueden crear *dashboards* para facilitar la visualización de los datos, mostrando la forma en la que se pueden modificar haciendo uso del lenguaje Flux y desarrollando algunas de las múltiples alertas esenciales para tener una rápida respuesta ante situaciones extremas.

6.1.2. The Things Board

The Things Board es una poderosa herramienta SW que tendrá un papel fundamental para el almacenamiento, procesamiento y representación gráfica de los datos recopilados en una variedad de aplicaciones y sectores.

Al igual que ocurría con *InfluxDB*, esta BD está diseñada específicamente para operar con datos de series temporales de una forma escalable y eficiente. Por tanto, podrá almacenar una gran cantidad de datos útiles para campos como *Big Data*.

Como se muestran en las figuras 6.1.16 y 6.1.17, se pueden realizar gráficas donde, a parte de representar el dato, se genere un pequeño informe que resuma lo observado, facilitando el análisis de los especialistas.

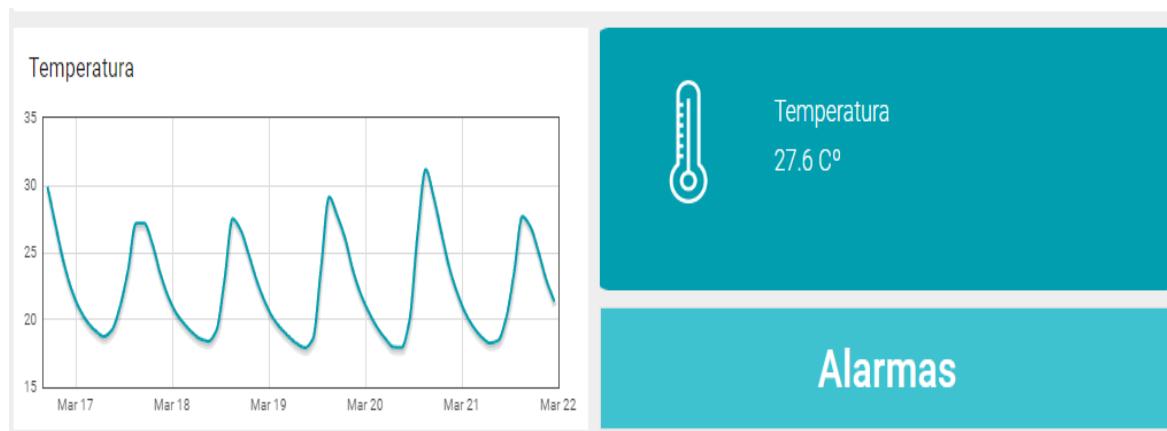


Figura 6.1.16 Datos de la temperatura ambiental recopilada por el segundo nodo

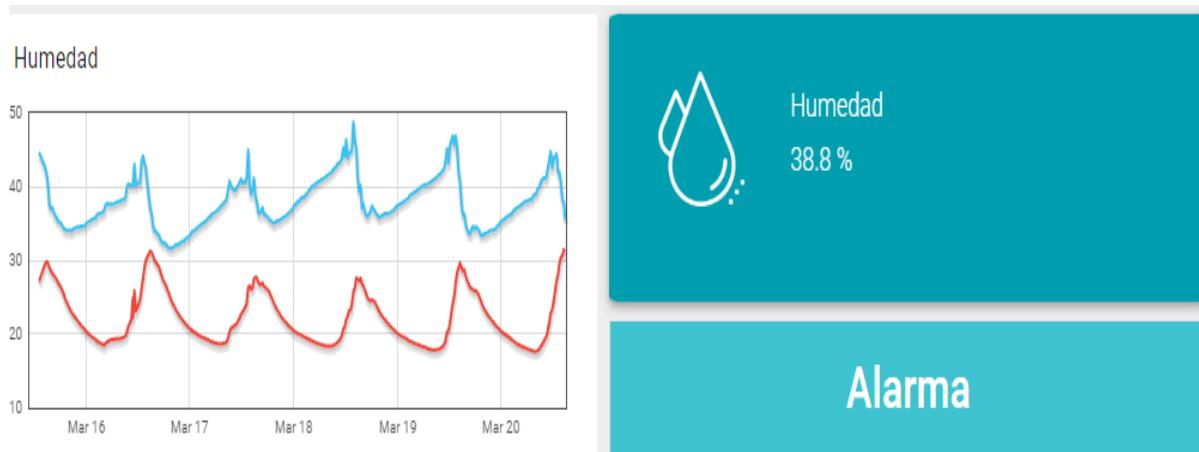


Figura 6.1.17 Datos de la humedad ambiental recopilada por el segundo nodo

Como se han observado en las figuras anteriores, la interfaz desarrollada es interactiva, debido a que proporciona la función programar alarmas que avisarán directamente al dispositivo móvil del usuario. Esto último se muestra en la figura 6.1.18.

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

Ajustes

X

Teléfono

Introduce el número de teléfono

Número de teléfono

GUARDAR

Temperatura límite

Introduce el número límite de temperatura

Límite de alarma ✓

GUARDAR

Figura 6.1.18 Panel para generar las alertas para la temperatura para el segundo nodo

Otra de las herramientas más importantes y potentes que tiene esta BD es la capacidad de exportar de forma cómoda y sencilla todos los datos recopilados, como se muestra en la figura 6.1.19. Esto último es fundamental para el procesamiento y análisis de los datos, pudiendo utilizar estos datos para entrenar modelos de aprendizaje máquina o de inteligencia artificial para poder realizar predicciones. (58)

Historial de datos

Timestamp ↓	Temperatura	Humedad	Batería
2023-08-26 14:18:27	30.71 °C	43.17 %	3.29 V
2023-08-26 02:18:27	31.40 °C	40.46 %	3.29 V
2023-08-25 14:18:27	31.36 °C	39.71 %	3.29 V
2023-08-25 02:18:27	31.26 °C	38.68 %	3.29 V
2023-08-24 14:18:27	31.39 °C	39.79 %	3.29 V
2023-08-24 02:18:27	31.28 °C	39.84 %	3.29 V
2023-08-23 14:18:27	31.17 °C	41.60 %	3.30 V
2023-08-23 02:18:27	31.14 °C	41.51 %	3.30 V
2023-08-22 14:18:27	31.24 °C	40.92 %	3.30 V
2023-08-22 02:18:27	31.06 °C	41.87 %	3.30 V
2023-08-21 14:18:27	31.15 °C	44.80 %	3.30 V

Figura 6.1.19 Algunos de los datos exportados de la BD The Things Boards

A parte de todo lo mostrado, esta herramienta tiene diversas funcionalidades que pueden ser de grandísima utilidad como *dashboards*, una zona de trabajo para realizar pruebas con los datos recogidos, etc., pero, por no saturar la memoria, no se van a desarrollar.

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

6.1.3. Bases de datos meteorológicas

A parte de todas las BD vistas, se van a estudiar el funcionamiento de las BD que están diseñadas expresamente para el estudio de las condiciones ambientales expuestas previamente. Este estudio también es importante para el análisis de la evolución del cambio climático y para comprobar el desarrollo del bosque de coníferas.

Ecowitt

Al igual que ocurre en *InfluxDB*, se tiene un *dashboard* donde se puede elegir el rango de tiempos que se quiere graficar. El inconveniente es que no es tan interactivo ni ajustable como *InfluxDB*, pero igualmente esta BD es de gran utilidad para el análisis.

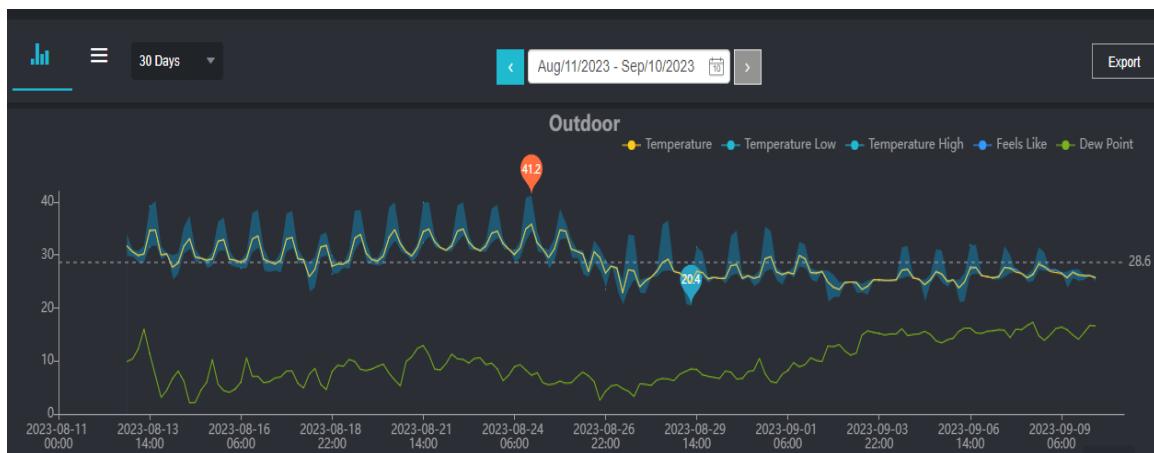


Figura 6.1.20 Gráfica de la temperatura en Ecowitt

También, se pueden crear alertas que notifiquen los casos extremos. En la figura 6.1.21 se muestra, a modo de ejemplo, alertas cuando la temperatura sea inferior a 30 °C. Los datos que hacen saltar estas alarmas se enviarán al correo, facilitando y agilizando el análisis y protección del medio en estudio.

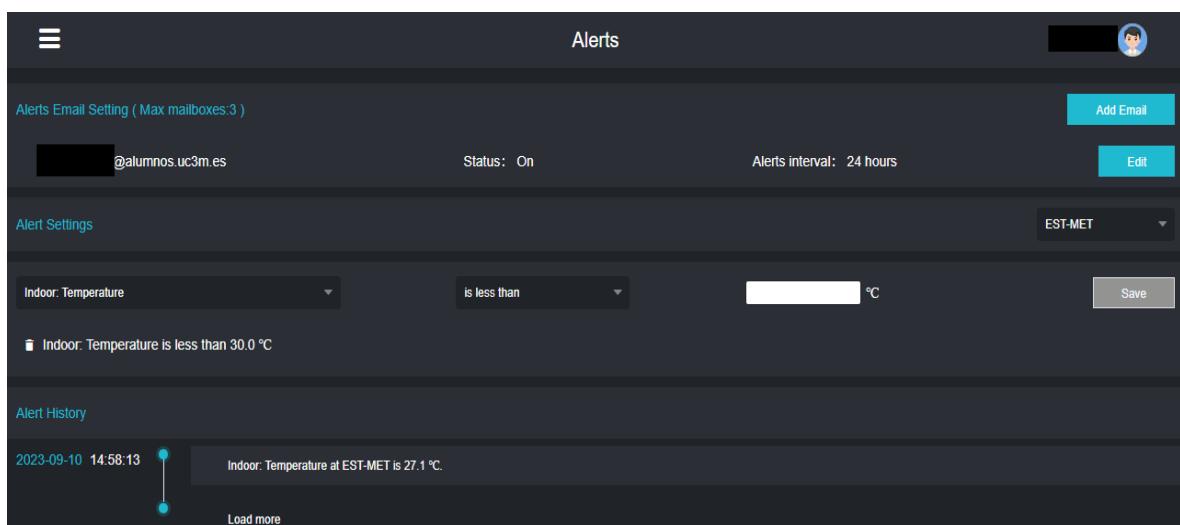


Figura 6.1.21 Creación de la alerta en Ecowitt

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

17:14

4G 36 %



Alert - Indoor: Temperature

at EST-MET is 27.1 °C.

Externo



Recibidos



Ecowitt Weather 14:58

para mí ▾



Traducir al español



Sun, 10 Sep 2:58 PM Indoor: Temperature at
EST-MET is 27.1 °C.



Please do not reply to this message. This e-mail was sent from a notification-only address that cannot accept incoming e-mail.

[Service](#) | [Shopping](#) | [Forum](#) | [Contact Us](#)

Copyright © 2023 Ecowitt Weather . All Rights Reserved.

Responder

Responder a
todos

Reenviar

No add-ons available for this email



Figura 6.1.22 Notificación de Ecowitt al saltar la alerta

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

Además, posee un mapa para ver la evolución de la temperatura de forma local, esto podría ser de gran utilidad para comprobar la evolución en las grandes áreas del bosque o colindantes a este.

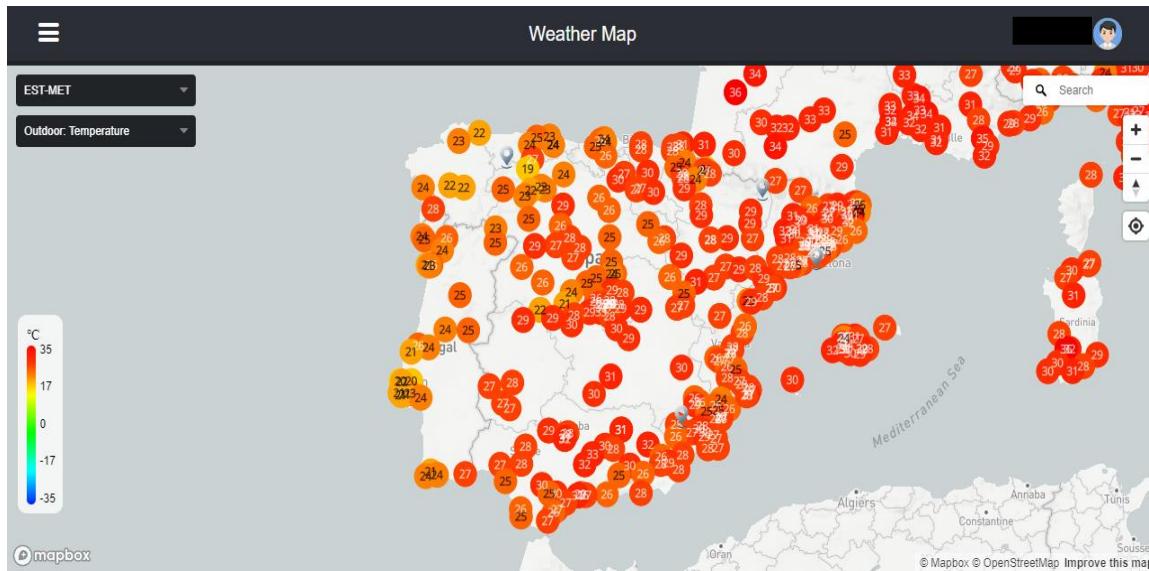


Figura 6.1.23 Mapa de temperatura en España con Ecowitz

Weather Underground

Esta BD tiene la capacidad de generar informes temporales que condensan la evolución del entorno estudiado a lo largo de un periodo definido de tiempo. Esta herramienta presenta una funcionalidad valiosa para los especialistas, debido a que les proporciona una síntesis concisa de lo ocurrido. Un ejemplo de ello se muestra en la figura 6.1.24.

Weather History for IMADRI633

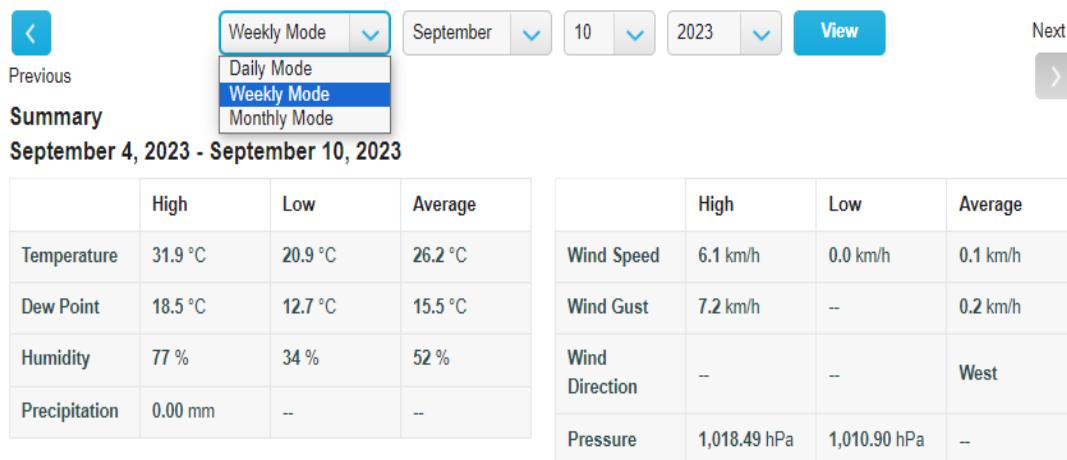


Figura 6.1.24 Informe semanal del tercer nodo en Weather Underground

Además, posee un *dashboard* donde se exponen los datos recopilados durante el tiempo de medida. Sin embargo, no es tan ajustable como *InfluxDB*. En la figura 6.1.25 muestra una de las gráficas del panel donde se observan la temperatura y el punto de rocío durante una semana.

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

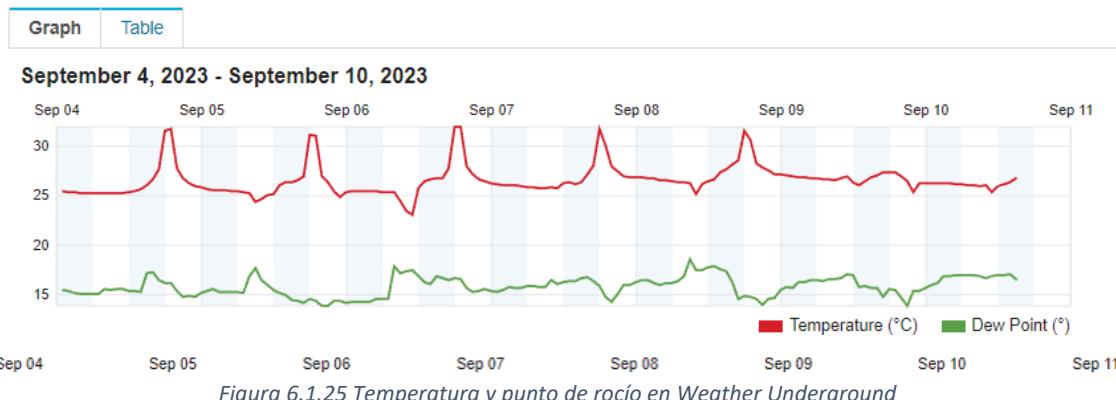


Figura 6.1.25 Temperatura y punto de rocío en Weather Underground

Como ocurría con Ecowitt, también se tiene un mapa donde se pueden estudiar la evolución del clima en la región elegida. Puede ser de gran utilidad para este proyecto al comprobar la evolución de un área tan extensa como puede ser un bosque de coníferas.



Figura 6.1.26 Mapa de temperatura en España con Weather Underground

Weathercloud

La última BD que proporciona funcionalidades útiles para el estudio del sistema de monitorización es *Weathercloud*. Esta BD proporciona la capacidad de descargar como fichero CSV todos los datos recopilados durante un periodo de tiempo, como se muestra en la figura 6.1.27. Esto es de gran interés para el procesamiento de los datos con algoritmos de regresión para generar un sistema de predicción meteorológica.

Date (Europe/Madrid)	Inside temperature (°C)	Temperature (°C)	Wind chill (°C)	Dew point (°C)	Heat index (°C)
01/09/2023 0:00	28.1	27	27	5.3	27
01/09/2023 0:10	28.1	27.1	27.1	5.4	27.1
01/09/2023 0:20	28.2	27.1	27.1	6	27.1
01/09/2023 0:30	28.1	27	27	6.5	27
01/09/2023 0:40	28	27.1	27.1	6.5	27.1
01/09/2023 0:50	28	26.5	26.5	7.6	26.5
01/09/2023 1:00	28	25.8	25.8	7.4	25.8
01/09/2023 1:10	27.9	25.6	25.6	7.7	25.6
01/09/2023 1:20	27.9	25.4	25.4	6.6	25.4
01/09/2023 1:30	28	25.9	25.9	6.5	25.9
01/09/2023 1:40	28	25.8	25.8	6	25.8
01/09/2023 1:50	27.9	25.2	25.2	5.4	25.2
01/09/2023 2:00	27.9	25.1	25.1	6.3	25.1

Figura 6.1.27 Archivo CSV generado del estudio en Weathercloud

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

Como se ve en la figura 6.1.28, esta BD también tiene la capacidad de generar informes útiles para el análisis de los datos meteorológicos en torno a los ejemplares de medición.

EST-MET September 2023										
Sep 2023	26.1 ^32.5 v21.0	48.6 ^76.0 v24.0	1013.8 ^1018.3 v1009.5	0.1 Fri 1 2.7	0.1 Fri 1 3.6	WSW Calm 0%	0.0 Fri 1 0.0	7.3 346.6	0.0 3.0	
Atmospheric										
Date	Temperature (°C)	Humidity (%)	pressure (hPa)	Wind speed (m/s)	Gust of wind (m/s)	Wind direction	Rain (mm)	Rain rate (mm/h)	Solar radiation (W/m²)	Evapotranspiration UV index
Fri 1	27.6	30.7	1014.4	0.3	3.6	WSW	0.0	0.0	10.1	- 0.0
Sat 2	25.2	43.4	1012.1	0.0	0.5	WSW	0.0	0.0	5.3	- 0.0
Sun 3	24.6	51.1	1012.3	0.0	0.0	WSW	0.0	0.0	2.4	- 0.0
Mon 4	26.0	51.6	1013.5	0.0	0.0	WSW	0.0	0.0	5.7	- 0.0
Tue 5	25.6	51.1	1016.2	0.0	1.0	WSW	0.0	0.0	10.7	- 0.0
Wed 6	25.8	52.9	1016.5	0.0	0.5	W	0.0	0.0	5.8	- 0.0
Thu 7	26.5	51.1	1013.7	0.0	1.0	W	0.0	0.0	9.0	- 0.0
Fri 8	26.9	50.6	1012.9	0.0	1.0	W	0.0	0.0	9.1	- 0.0
Sat 9	26.4	51.0	1012.5	0.2	2.0	W	0.0	0.0	9.0	- 0.0

Figura 6.1.28 Informe semanal del tercer nodo en Weathercloud

Esta BD carece de algo al que se le pueda denominar *dashboard*, en cambio, tiene un sistema para poder representar una o varias magnitudes enfrentadas, en un rango temporal indicado (figura 6.1.29). Incluso, la propia gráfica es capaz de generar un pequeño informe que resume lo representado (figura 6.1.30). Convirtiéndose en una herramienta muy potente y ajustable que puede ser de gran utilidad para los analistas.

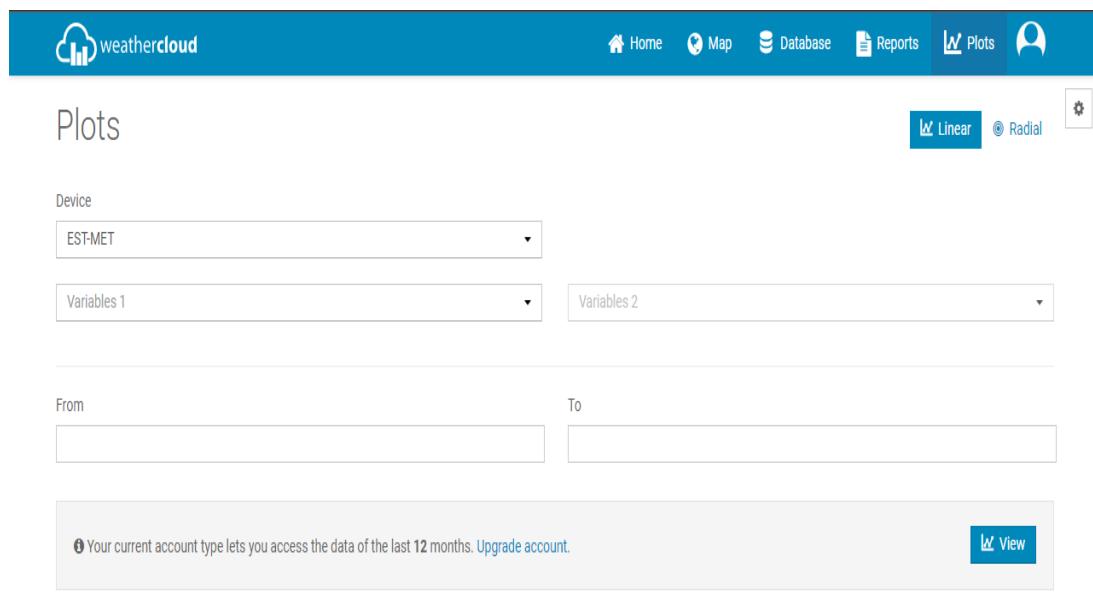
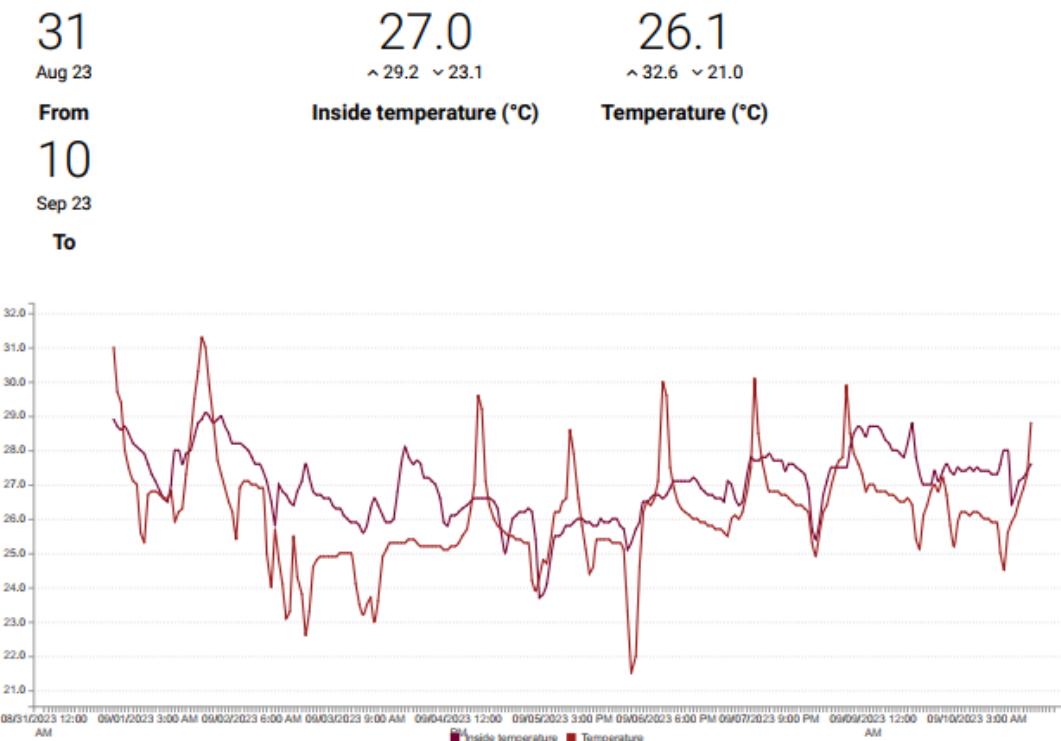
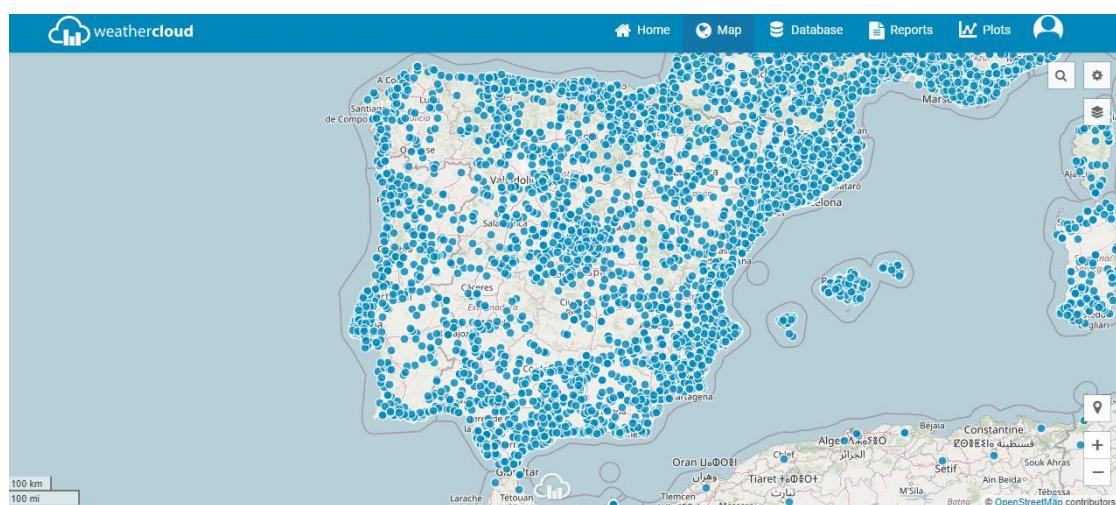


Figura 6.1.29 Sistema de elección de magnitudes a graficar proporcionada por Weathercloud

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES



Al igual que ocurría en BD meteorológicas previas, tiene una funcionalidad de mapa para estudiar la evolución climática del entorno en estudio (figura 6.1.31).



6.2. Resultados

Con todos los datos recopilados, se procede a realizar un análisis de los datos tratados. Es remarcable decir que este TFM de IoT se enfoca en la creación de la red, por ende, este análisis tendrá una naturaleza más superficial.

El propósito fundamental de este proyecto es crear un sistema que sea funcional y útil, con el objetivo de brindar herramientas que facilite y agilice a expertos el análisis de la evolución del cambio climático y de sus efectos sobre el entorno forestal.

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

Es imprescindible explicar, para el correcto entendimiento de los resultados obtenidos, que las mediciones se realizaron con un ejemplar de pino joven confinado en una maceta en un entorno urbano y cerrado. Por tanto, los datos que se van a obtener diferirán de lo previsto, tanto a nivel ejemplar como medioambiental, debido a la idiosincrasia del entorno estudiado.

6.2.1. Magnitudes propias del árbol

En la presente sección se desarrollará el análisis de los datos adquiridos del ejemplar en estudio. Se ha de comentar que se ha hecho uso de un ejemplar de pino joven para el presente análisis. Estos datos se representarán con diferentes gráficas para facilitar el estudio del ejemplar.

Se concedió al sistema dos períodos de estudios diferentes para lograr atisbar levemente la evolución de las magnitudes de medida, esto permitirá comprobar como el cambio de clima afecta al desarrollo del ejemplar.

Es necesario destacar que el sistema está diseñado para obtener los datos a intervalos amplios de tiempo, desde varios minutos hasta horas. Sin embargo, a causa de restricciones en cuestión de tiempo. Se disminuyó el periodo para la toma de medidas a diez segundos, de esta forma se puede obtener una mayor cantidad de datos y así poder realizar un análisis más detallado.

Lamentablemente, por falta de tiempo no se va a poder realizar el análisis de las magnitudes propias del árbol con el segundo tipo de nodo implementado.

Para llevar a cabo este estudio se van a comparar los datos graficados con estudios previos para las magnitudes de medida. Es importante recalcar que los datos de referencia no serán para la especie en estudio, pero al no querer realizar un análisis exhaustivo, esto último no conllevará un grave perjuicio.

- **Humedad de la hoja**

Se comienza analizando la humedad de la hoja. Los estudios de referencia se llevaron a cabo en ejemplares ubicados en un entorno natural al aire libre. De esta forma, se concluye que por las noches aumenta la humedad foliar debido al rocío, en cambio, por el día se mantiene más seco. Es importante destacar que el estudio de referencia de Nathan Emery (59) podría ser de utilidad como indicador potencial de enfermedades para este tipo de ejemplares.

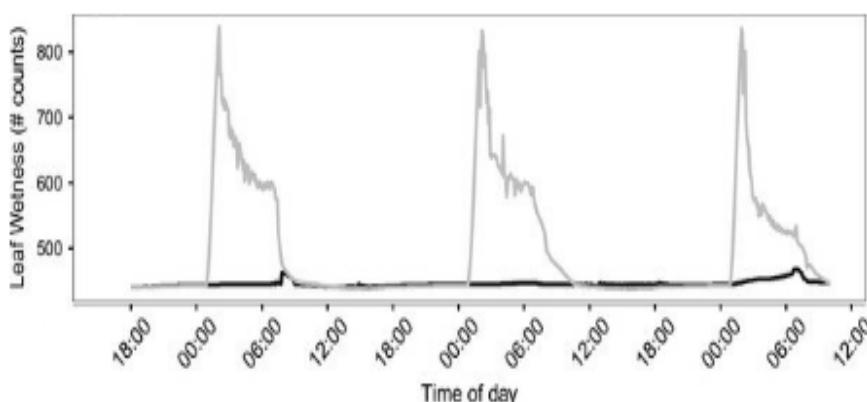
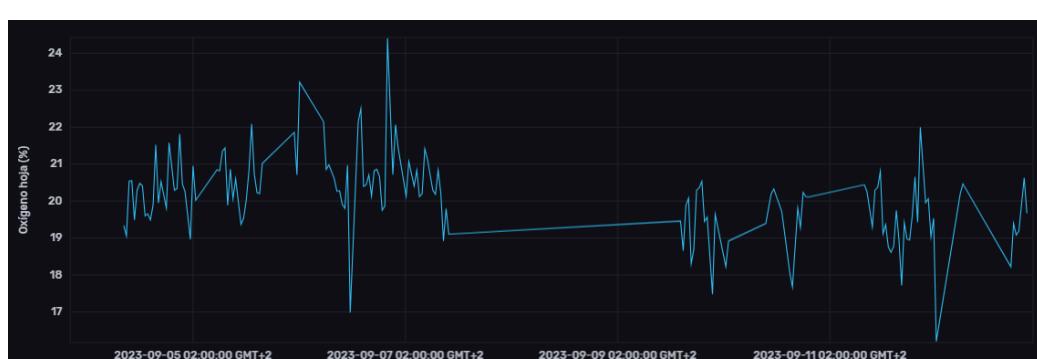


Figura 6.2.1 Evolución de la humedad de las hojas (59)

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

Todo lo expuesto anteriormente, se ve perfectamente indicado en la figura 6.2.1, la humedad aumenta de forma drástica a medianoche y va descendiendo hasta que amanece. A continuación, se representan las gráficas obtenidas en los dos períodos de estudio:

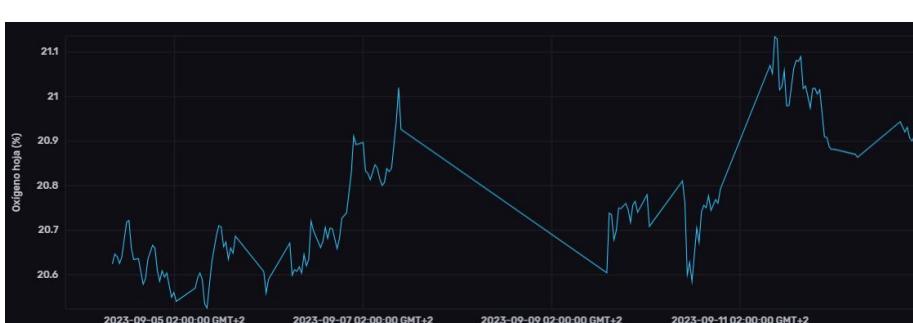
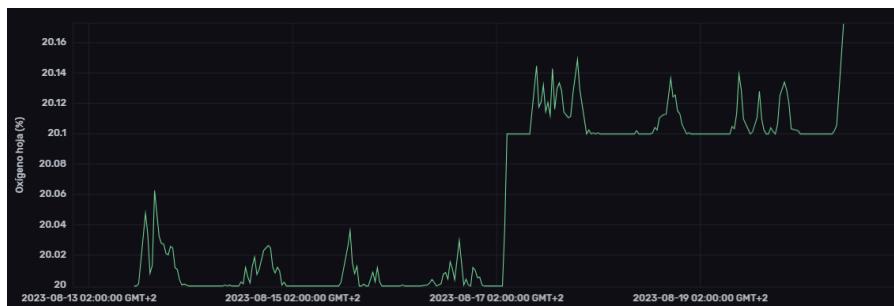


REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

Como se pueden observar en los datos de las figuras 6.2.2, 6.2.3, 6.2.4 y 6.2.5, el sistema es muy ruidoso, oscilando siempre en un rango estable entre el 16 % y el 28 %. Además, se observa una tendencia a un aumento de humedad por la noche, siendo el día ligeramente más seco.

• Oxígeno de la hoja

El sensor utilizado para la monitorización de esta magnitud no está diseñado para la medición foliar, por tanto, se ha de situar el sensor próximo al follaje para recoger datos significativos. Desafortunadamente, no se pudo encontrar trabajos que hiciesen estudios rigurosos sobre el O_2 generado por las hojas a lo largo del tiempo.



REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

Se comprueba en las figuras 6.2.6, 6.2.7, 6.2.8 y 6.2.9 que el sensor es robusto, por tanto, a pesar de que haya “picos” en las gráficas, se puede observar que son en rangos muy precisos de esta magnitud. Se puede ver que durante el primer periodo de medición el ejemplar producía menos cantidad de oxígeno, esto puede ser debido a las diferencias de temperaturas existentes.

- **Dióxido de carbono de la hoja**

Esta magnitud sufre una variación significativa a lo largo del día debido al proceso fotosintético del árbol. Los árboles respiran por las noches produciendo, por tanto, mayor cantidad de CO_2 por la noche y al realizar el proceso fotosintético por el día se limitará la producción de esta magnitud. Sin embargo, es fundamental destacar que, al tomar los datos en un entorno urbano, vehículos y otras máquinas pueden contrarrestar este efecto.

En la figura 6.2.10 la evolución típica del CO_2 a lo largo del día en un entorno natural y en uno urbano.

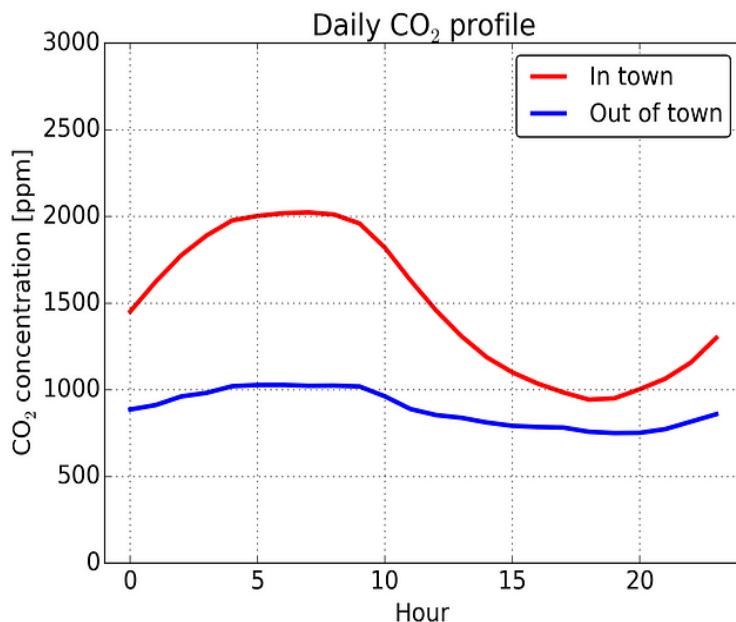


Figura 6.2.10 Evolución del CO_2 del aire en un día en una ciudad y en una zona rural (60)



Figura 6.2.11 Evolución del CO_2 del pino durante todo el primer periodo de medición

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES



En las figuras 6.2.11 y 6.2.12 se comprueba que, a pesar de lo esperado por la fotosíntesis, el CO_2 incrementa por el día y se reduce por la noche. Seguramente esto se deba que se han tomado las medidas en un barrio y no en un entorno forestal, por ende, esta magnitud se ve influenciada por la actividad humana.

Además, se puede observar en las figuras 6.2.13 y 6.2.14 que el CO_2 aumenta hasta que satura el sistema, esto último seguramente se deba que el sensor ha dejado de funcionar de la forma adecuada.

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

• pH del suelo

El rango óptimo del pH para el desarrollo adecuado de los pinos varía entre 6,5 y 7,0. Esto indica que se desarrollan correctamente entre entornos ligeramente ácidos y neutros. En la figura 6.2.15 se expone la evolución de dos zonas de un bosque, una quemada y la otra de control. (61)

Se verifica en la siguiente figura que el suelo se vuelve más básico al sufrir un incendio. Sin embargo, lo que interesa es que este no sufre una gran evolución a lo largo del tiempo si se mantiene inalterado.

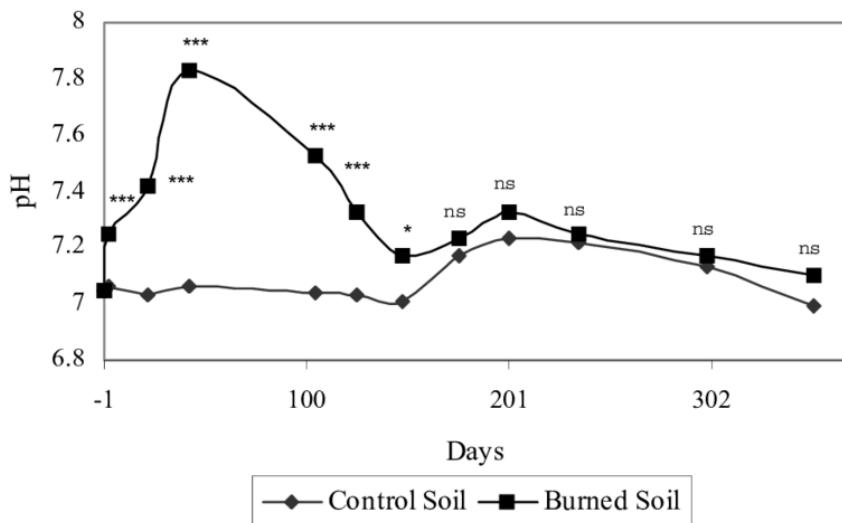


Figura 6.2.15 Evolución del pH del suelo en una zona controlada y en otra quemada (62)



Figura 6.2.16 Evolución del pH del pino durante todo el primer periodo de medición



Figura 6.2.17 Evolución del pH del pino durante un día del primer periodo

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

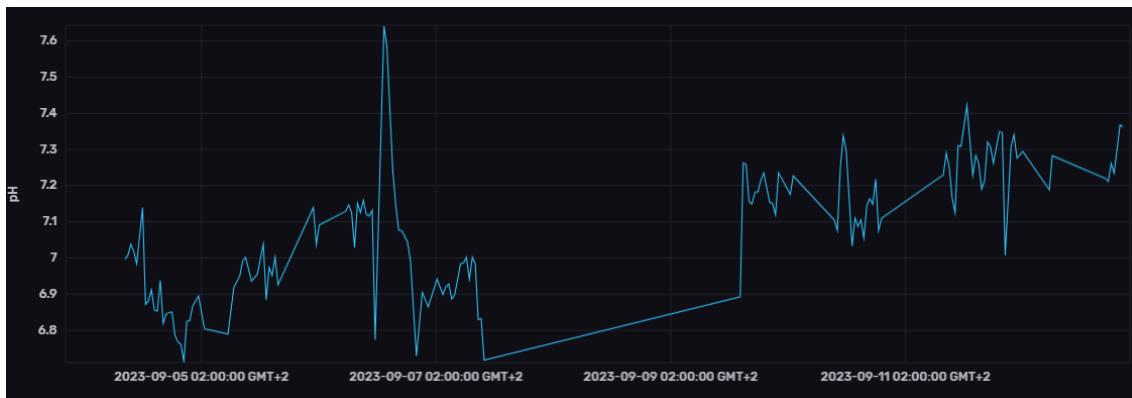


Figura 6.2.18 Evolución del pH del pino durante todo el segundo periodo de medición



Figura 6.2.19 Evolución del pH del pino durante un día del segundo periodo

En las figuras 6.2.16, 6.2.17, 6.2.18 y 6.2.19 se observa la evolución del pH del pino en los dos períodos de medida. Se comprueba que el pH del suelo es ligeramente inferior al óptimo durante el primer periodo de medición, pero en el segundo periodo, después de abonarlo, se aprecia un aumento significativo del pH, siendo ligeramente superior al óptimo.

6.2.2. Magnitudes meteorológicas

Para finalizar este análisis, se van a estudiar los resultados que se han obtenido de la estación meteorológica del tercer nodo. Por proximidad, en este estudio se van a comparar los datos obtenidos con los recopilados en el barrio de Carabanchel de la BD meteorológica *Meteosierra*. (63)

Se recuerda que la estación meteorológica se ha situado en el interior de un recinto, pero con las ventanas abiertas. Por lo cual, estos datos no serán del todo verosímiles, sería conveniente para futuros proyectos realizar el análisis en el exterior de una zona boscosa y no en el interior de un local en una urbe.

• Temperatura ambiental

Se observa en la figura 6.2.20 la evolución de diversos parámetros de temperatura como el bochorno, la sensación climática o el punto de rocío junto con la humedad. Se procede a compararlo con los datos de la estación meteorológica del tercer tipo de nodo, representados en las distintas BD utilizadas en las figuras 6.2.21, 6.2.22, 6.2.23 y 6.2.24.

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

Se observa que en la BD *Meteosierra* varía la temperatura de una forma razonable, comprobando que la temperatura aumenta por la mañana y disminuye por la noche, quitando el nueve de septiembre. Esto último no es por la situación meteorológica de ese día, sino que es debido a, como se puede comprobar en gráficas posteriores, que durante varias horas de ese día la BD dejó de tomar datos.

Sin embargo, se observa que los datos obtenidos por el tercer tipo de nodo respecto a la temperatura se mantienen estables gran parte del día disminuyendo de forma considerable por la noche y aumentando de forma abrupta a media tarde, esto se debe a la localización del dispositivo. La estación meteorológica se sitúa en una sala orientada al oeste donde se deja la ventana abierta, por ese mismo motivo sale lo reflejado.

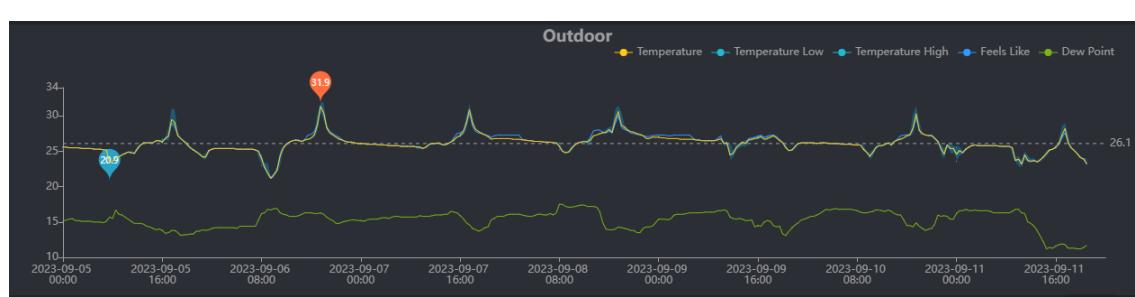
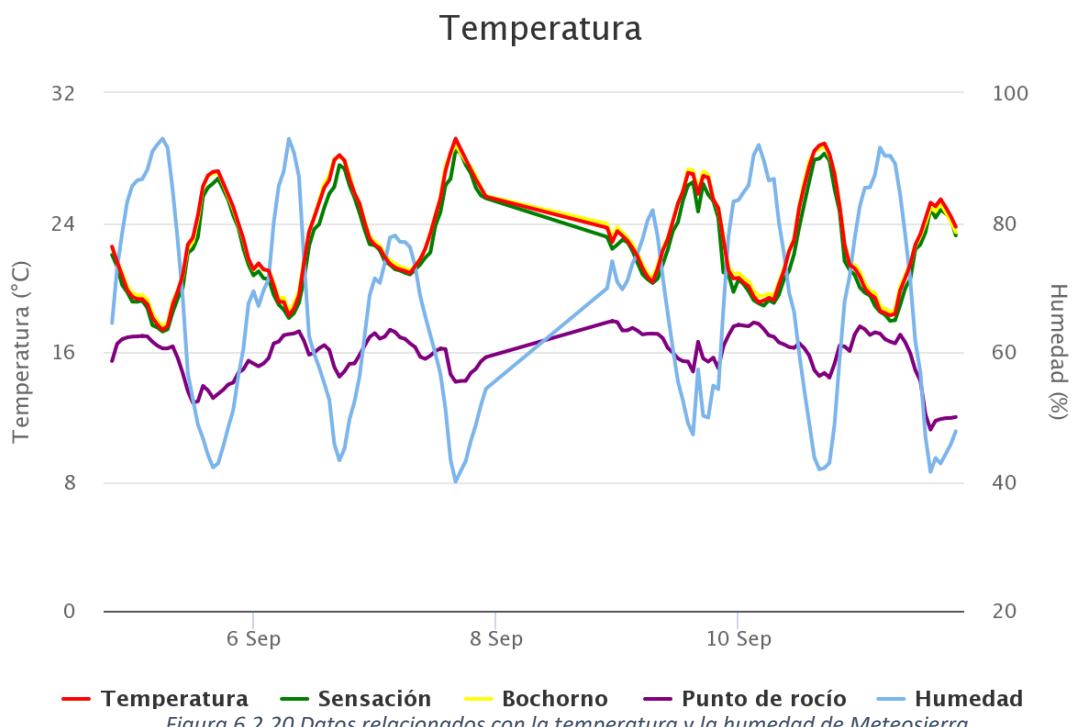


Figura 6.2.21 Datos relacionados con la temperatura exterior del tercer tipo de nodo representados en Ecowitt

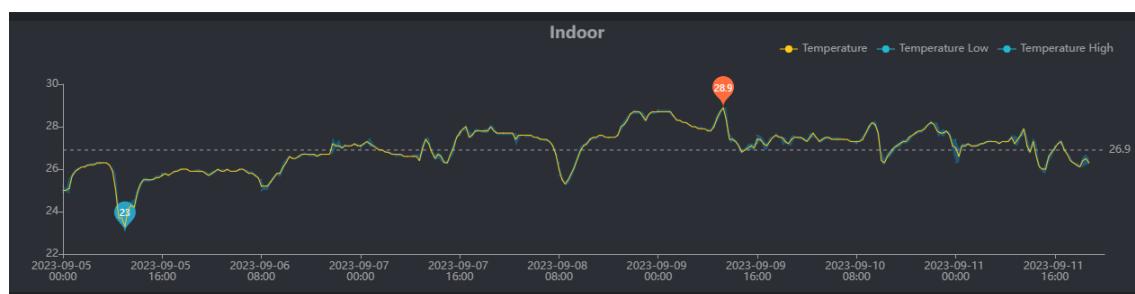


Figura 6.2.22 Datos relacionados con la temperatura interior del tercer tipo de nodo representados en Ecowitt

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

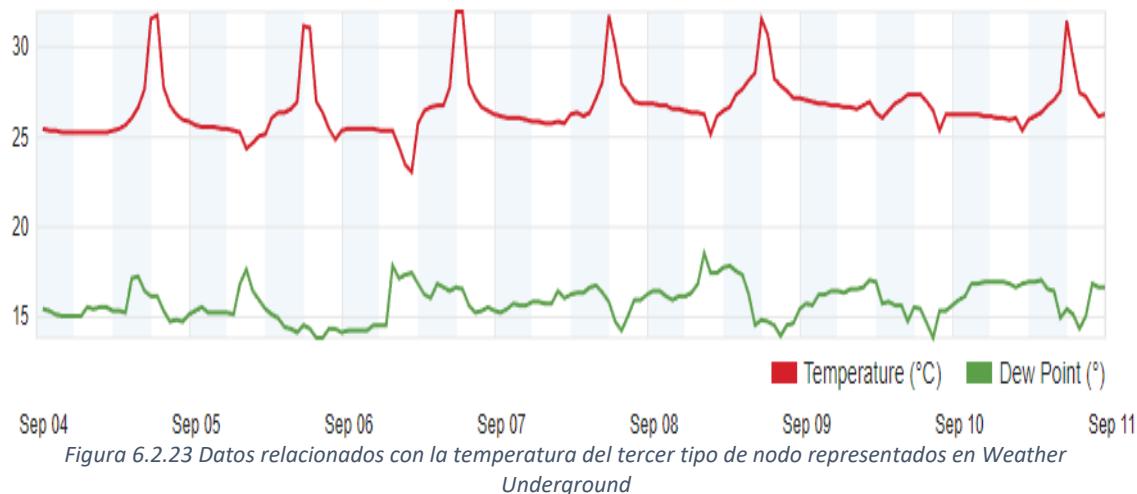


Figura 6.2.23 Datos relacionados con la temperatura del tercer tipo de nodo representados en Weather Underground

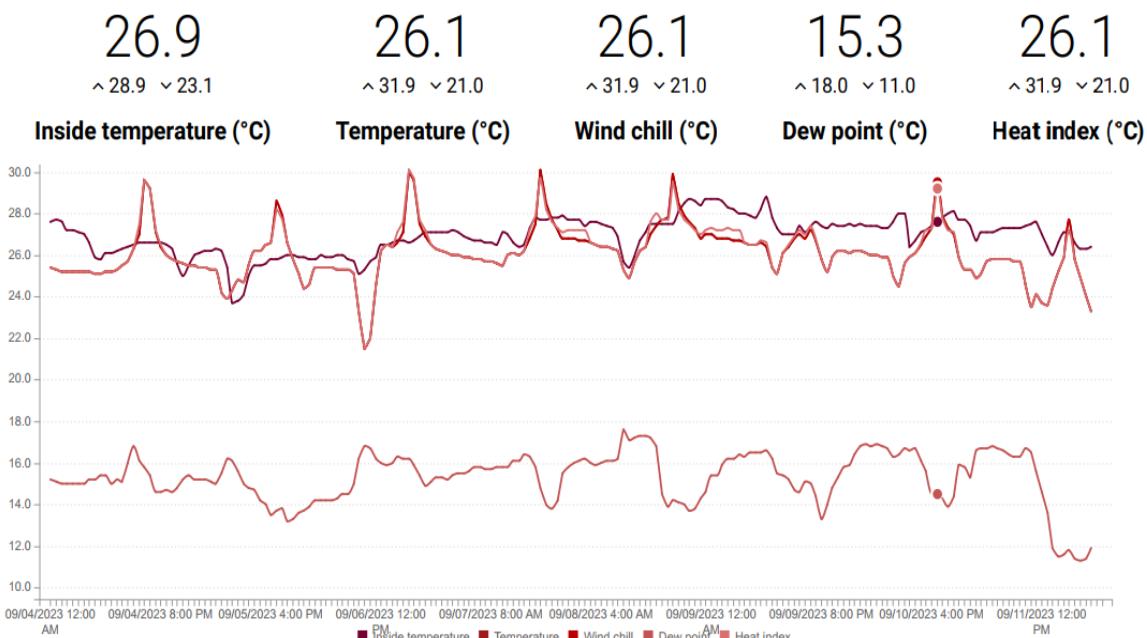


Figura 6.2.24 Datos relacionados con la temperatura interior del tercer tipo de nodo representados en Weathercloud

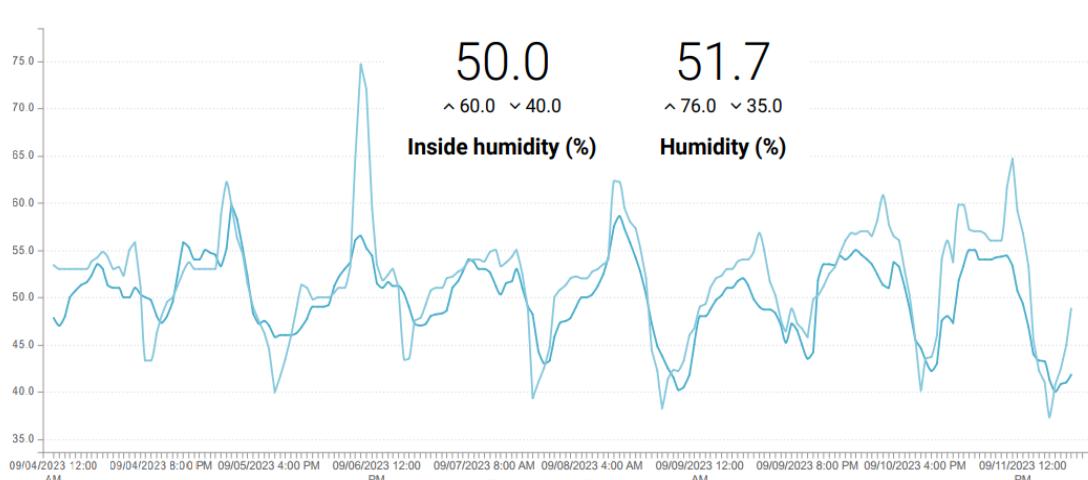
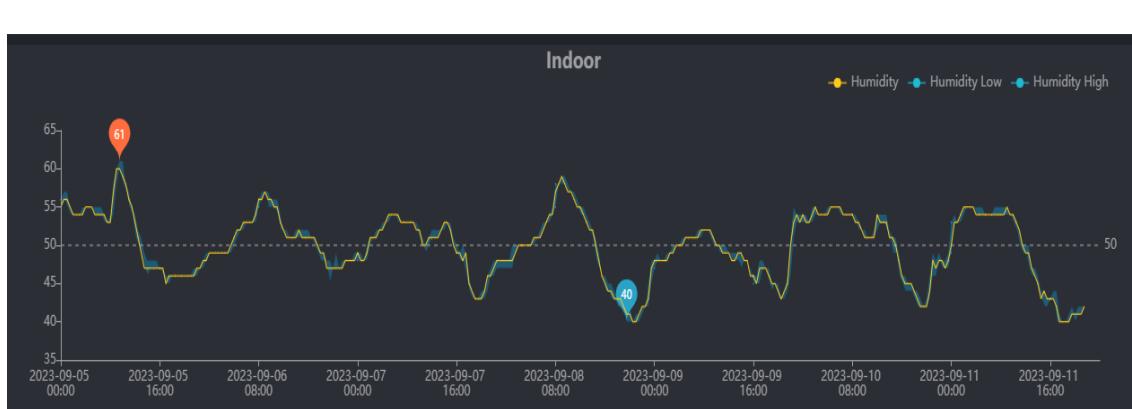
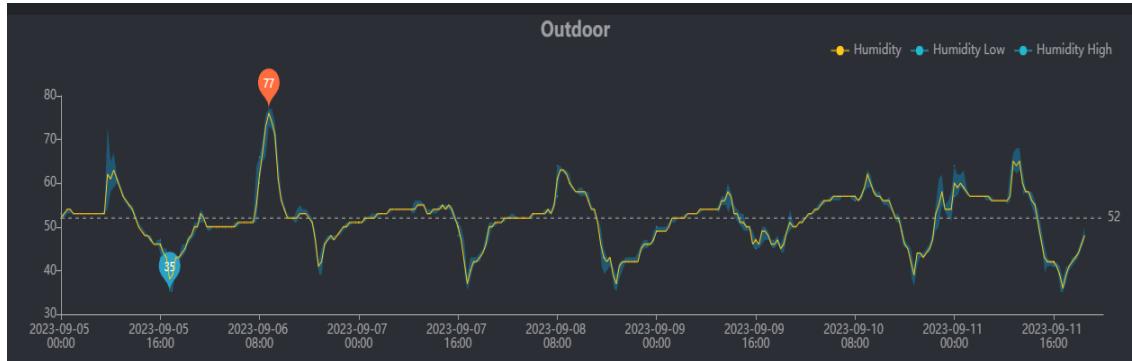
• Humedad ambiental

En la figura 6.2.20 se refleja la evolución de la humedad a lo largo de la semana del cuatro de septiembre. Estos datos se confrontarán con los datos de la estación meteorológica del nodo que se encarga de medir solamente parámetros medioambientales, estos datos se recogen en las figuras 6.2.25, 6.2.26 y 6.2.27 sobre las diversas BD utilizadas. Lamentablemente, la BD *Weather Underground* no tiene la capacidad de representar la humedad.

Ambas gráficas representan una evolución no idéntica, pero sí similar. Se comprueba que la humedad aumenta por la noche, pero disminuye de forma considerable a lo largo del día.

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

Se puede observar en los datos recopilados por los nodos que la humedad es alta por haber sufrido unas épocas de lluvias, pero a las horas del atardecer se experimenta una gran disminución debido al lugar donde se sitúa la estación meteorológica.



• Dirección y velocidad del viento

En la figura 6.2.28 se observa la evolución de la dirección y velocidad del viento a lo largo de la semana según las medidas tomadas por la BD *Meteosierra*. Si se comparan con los datos del nodo sensor, mostrados en las figuras 6.2.29, 6.2.30, 6.2.31, 6.2.32, 6.2.33 y 6.2.34, se puede comprobar que la dirección se mantiene estática apuntando prácticamente al oeste y la velocidad del viento no corresponde del todo con lo mostrado en la figura 6.2.28.

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

Esto último se debe a que, como se dijo anteriormente, la localización del dispositivo en una vivienda, por tanto, solamente recibirá ráfagas de vientos que vayan del este al oeste o viceversa. Aunque los datos difieran completamente, sí que se puede inferir de la figura 6.2.29 como en aquellos momentos que hay ráfagas de viento con la misma dirección indicada, hay rachas de viento similares en el nodo de medición.

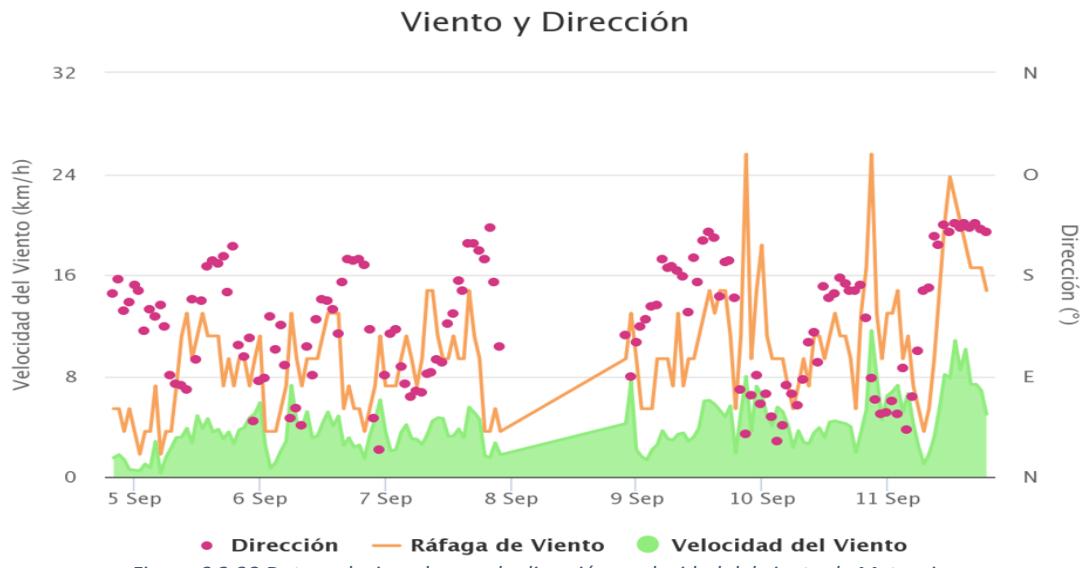


Figura 6.2.28 Datos relacionados con la dirección y velocidad del viento de Meteoserra

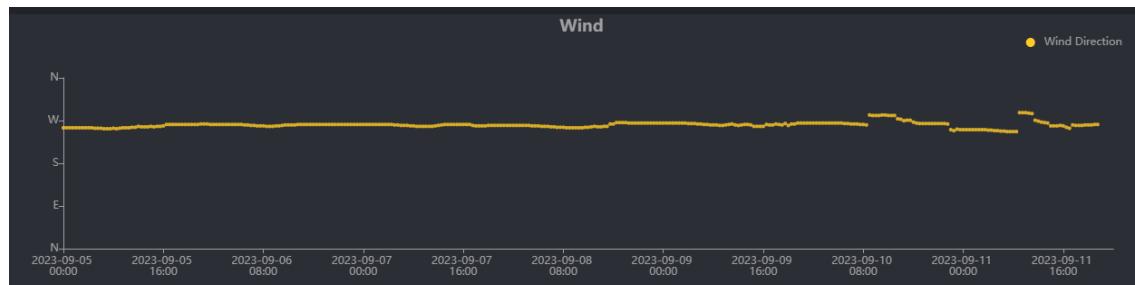


Figura 6.2.29 Datos relacionados con la dirección del viento del tercer tipo de nodo representados en Ecowitz

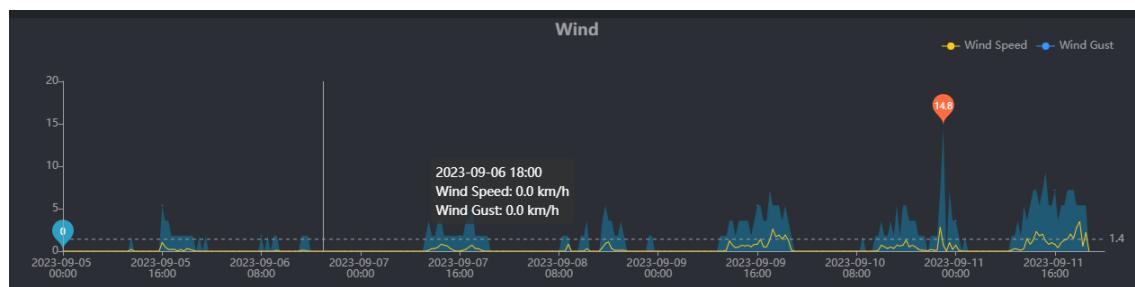


Figura 6.2.30 Datos relacionados con la velocidad del viento del tercer tipo de nodo representados en Ecowitz



Figura 6.2.31 Datos relacionados con la dirección del viento del tercer tipo de nodo representados en Weather Underground

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

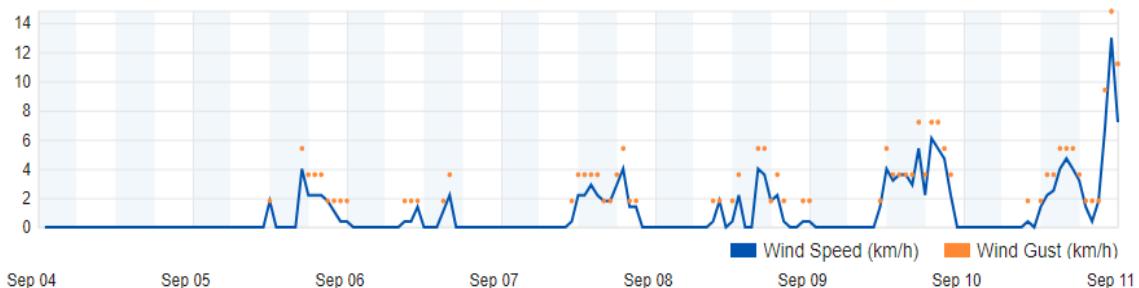


Figura 6.2.32 Datos relacionados con la velocidad del viento del tercer tipo de nodo representados en Weather Underground

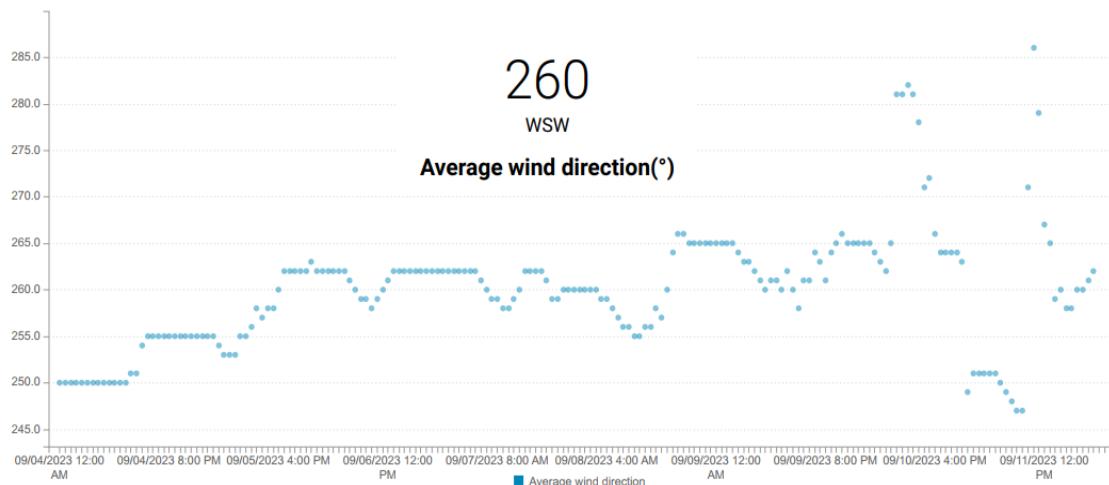


Figura 6.2.33 Datos relacionados con la dirección del viento del tercer tipo de nodo representados en Weathercloud

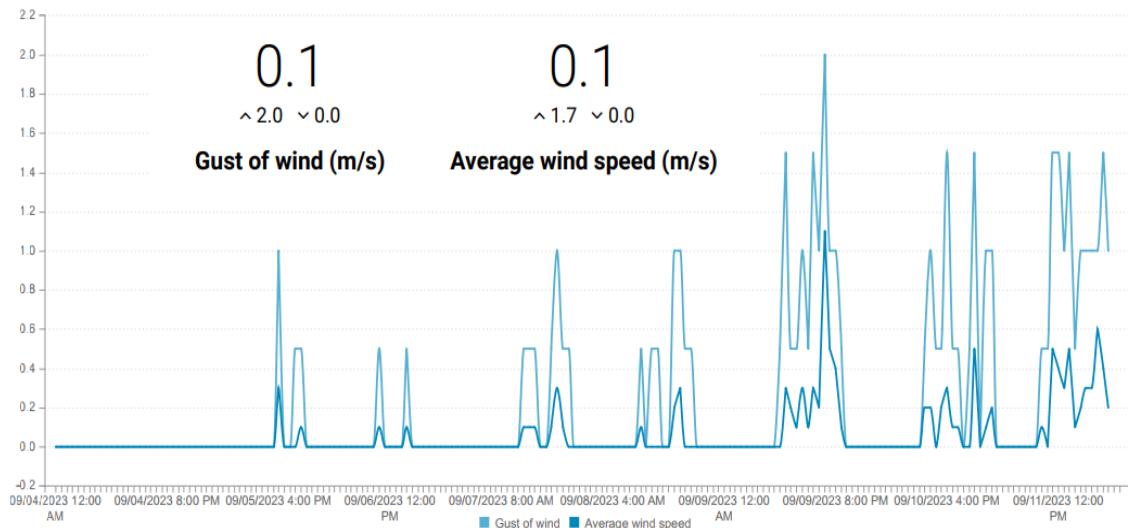


Figura 6.2.34 Datos relacionados con la velocidad del viento del tercer tipo de nodo representados en Weathercloud

• Radiación solar

Los datos de radiación solar e índice ultravioleta dado por la BD de Meteosierra, mostrados en la figura 6.2.35, concuerdan con lo esperado. Se observa a simple vista que por la mañana aumentan progresivamente estos valores hasta aproximadamente las dos de la tarde y, posteriormente, va disminuyendo hasta que anocche y se vuelve nulo.

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

Sin embargo, los datos obtenidos por el nodo, mostrados en las figuras 6.2.36, 6.2.37, 6.2.38 y 6.2.39, difieren al incrementar levemente a lo largo del día hasta a las cinco de la tarde que aumentan de forma abrupta.

Esto último se debe a, como se ha visto en otras ocasiones, la disposición de la estación meteorológica en el local, al situarse al oeste, esto va a ocasionar que los valores reales se recojan en una pequeña ventana de tiempo al estar a resguardo el resto del día.

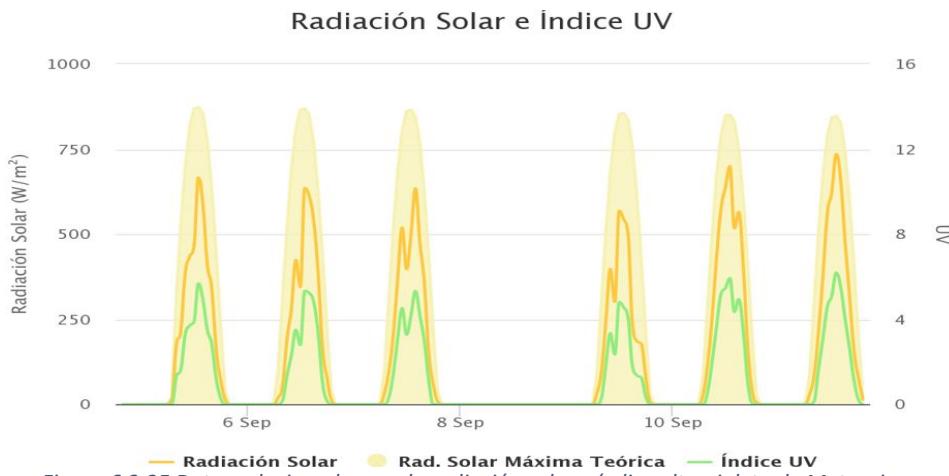


Figura 6.2.35 Datos relacionados con la radiación solar e índice ultravioleta de Meteosierra

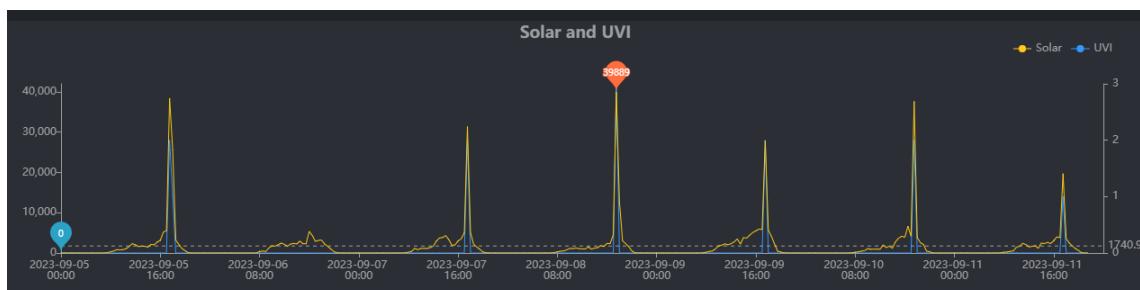


Figura 6.2.36 Datos relacionados con la radiación solar e índice ultravioleta del tercer tipo de nodo representados en Ecowitt

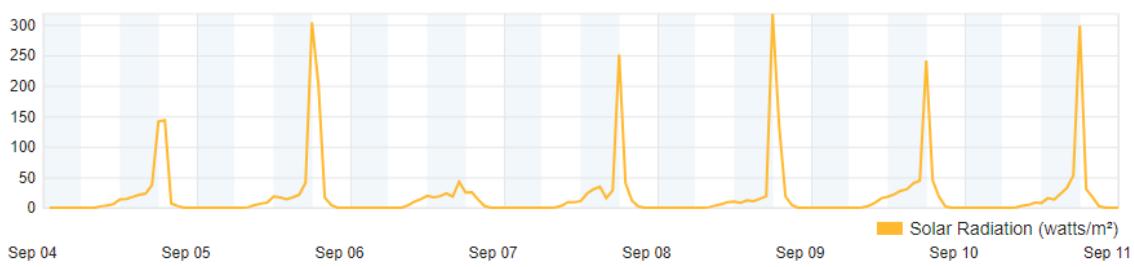


Figura 6.2.37 Datos relacionados con la radiación solar del tercer tipo de nodo representados en Weather Underground

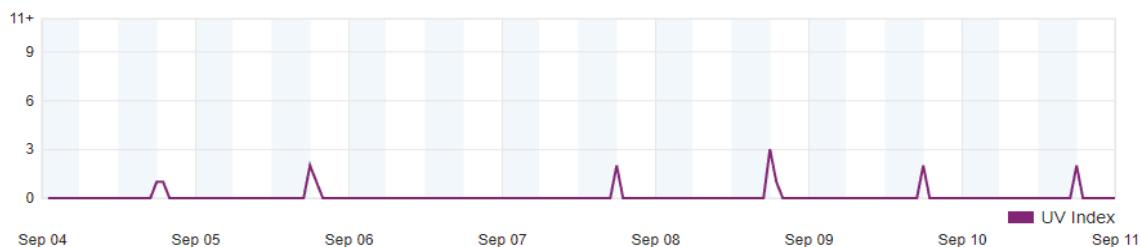
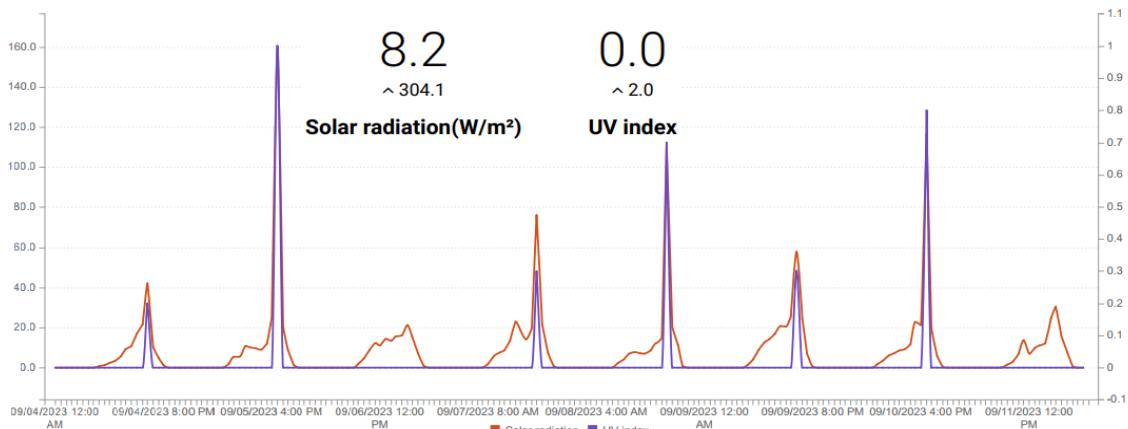


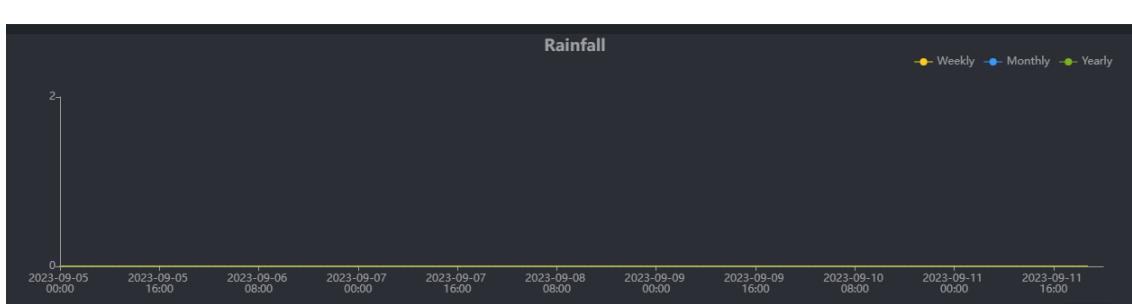
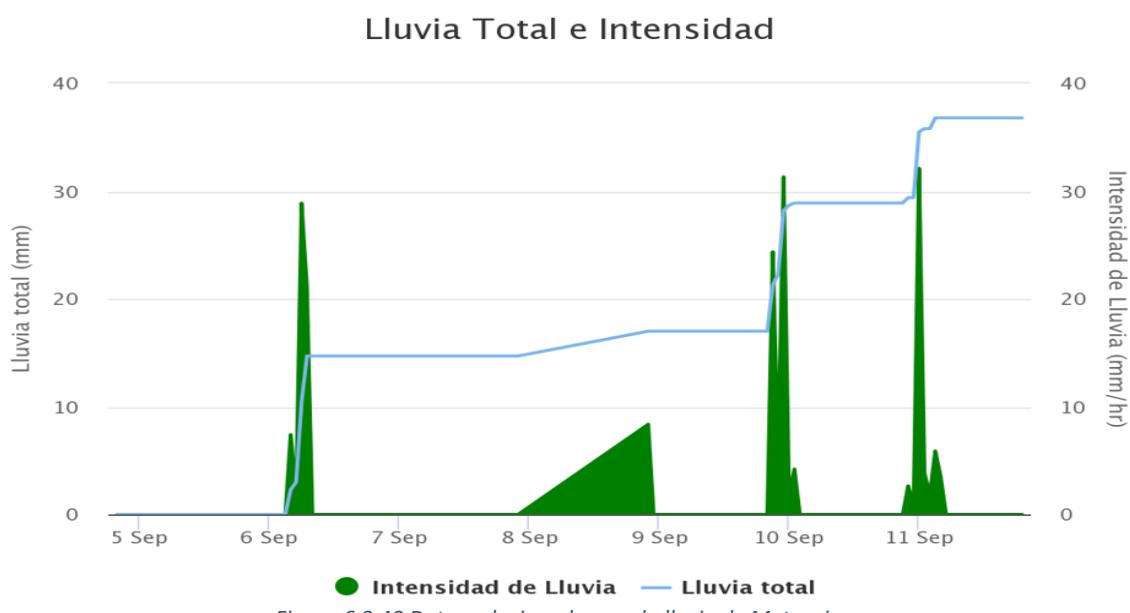
Figura 6.2.38 Datos relacionados con el índice ultravioleta del tercer tipo de nodo representados en Weather Underground

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

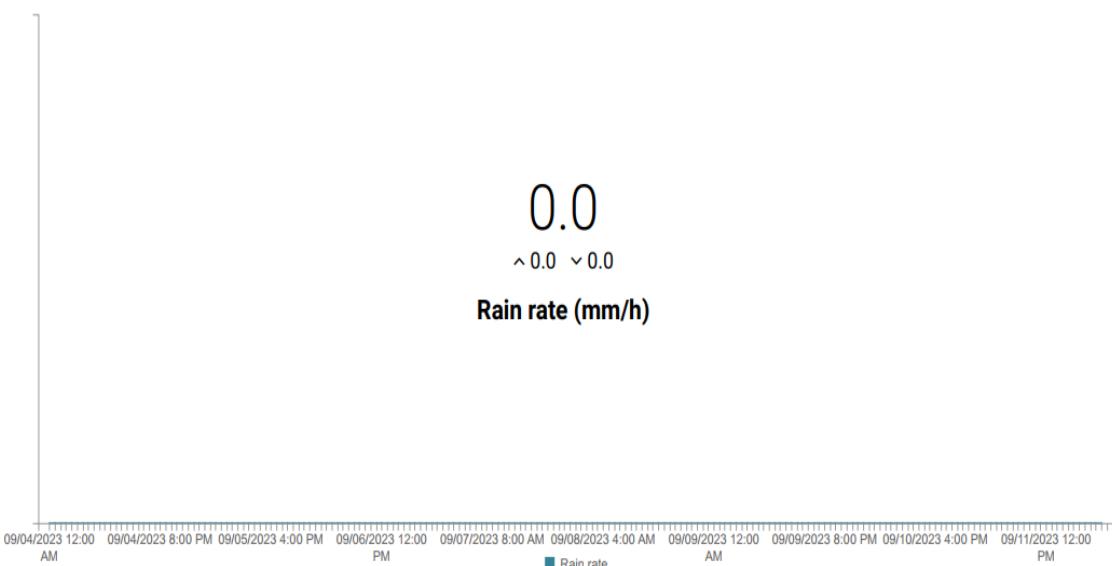


• Precipitaciones

Se observa en la figura 6.2.40 la evolución de las precipitaciones según lo medido por las estaciones meteorológicas de *Meteosierra* y en las figuras 6.2.41, 6.2.42 y 6.2.43 se muestran los datos recopilados por el tercer tipo de nodo. Sin embargo, al estar el nodo a resguardo en el interior de una vivienda, no hay registro de precipitaciones.



REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES



• Presión atmosférica

La BD *Meteosierra* no recoge los datos relacionados con la presión atmosférica, por tanto, en este análisis no se podrá realizar una comparativa entre los datos de referencia y los obtenidos. Todos los datos de medidas en las diferentes BD utilizadas se exponen en las figuras 6.2.44, 6.2.45 y 6.2.46.

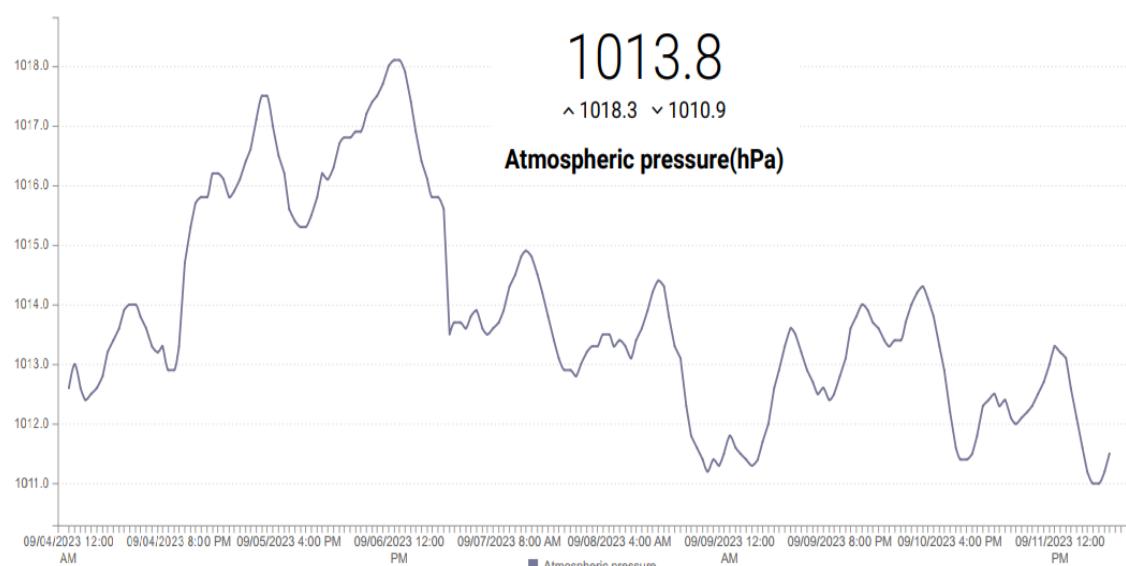
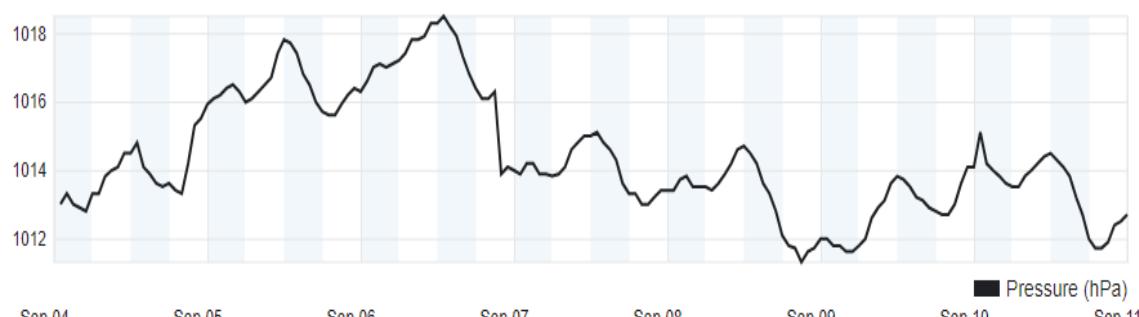
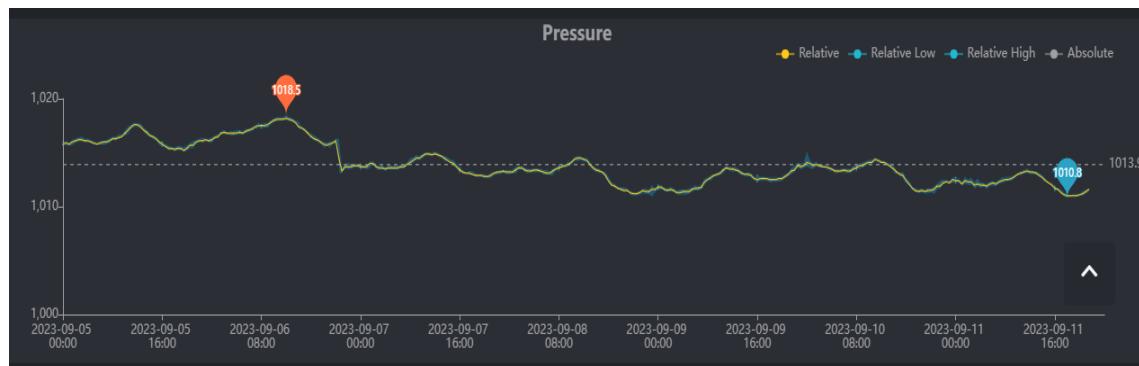
Antes de comenzar el análisis, hay que exponer unas nociones básicas sobre la presión atmosférica. Esta es influenciada tanto por la temperatura como por las precipitaciones:

- **Temperatura:** cuando hace más calor el aire cerca de la superficie aumenta su temperatura expandiéndose, por tanto, disminuye la densidad de aire lo que conlleva una disminución de la presión atmosférica. Lo contrario ocurre cuando hace frío al aumentar la densidad y, por ende, la presión atmosférica. Normalmente, por el día hará menos presión que por la noche.
- **Precipitaciones:** en zonas de baja presión atmosférica el aire sube enfriándose, de esta forma, se asocian como sistemas climáticos más activos, es decir, suele haber mayor precipitación. Lo contrario ocurre en zonas de alta presión al descender el aire y aumentar la densidad.

Lo expuesto con anterioridad se relaciona con lo reflejado en las gráficas, cuando llegaron las lluvias disminuyó la presión atmosférica. Además, también se observa que

REPRESENTACIÓN GRÁFICA Y ESTUDIO DE LAS MEDICIONES

la presión atmosférica empieza a aumentar según llega la noche, continúa hasta bien entrada la mañana y por la tarde comienza a disminuir.



7. Conclusiones y trabajo futuro

Una vez se ha realizado la implementación y el análisis del sistema, en este apartado se expondrán las conclusiones que se derivan del desarrollo del trabajo, junto con posibles mejoras y propuestas de futuro. Además, se puede observar el coste del proyecto realizado en el [apéndice I](#).

7.1. Conclusiones

La finalidad del presente trabajo reside en realizar una red IoT para la monitorización de un bosque de coníferas junto con las magnitudes medioambientales que pueden influir en el desarrollo del mismo. Para ello se ha implementado un sistema a pequeña escala donde se monitoriza un ejemplar de pino. En una monitorización a largo plazo y mayor escala se podrán observar los efectos mutuos del tejido de bosques y el cambio climático.

Este trabajo es complementario a otro donde se han desarrollado nodos sensores para la monitorización de dos ejemplares de encinas. En este proyecto se ha ampliado el anterior trabajo, creando un nuevo nodo sensor, junto con nuevos nodos centrales que se encargan de comunicarse con los nodos anteriormente creados para tratar los datos y enviarlos a las BD.

Se ha conseguido un sistema funcional, aunque modesto. Si bien da la posibilidad de extenderlo ampliamente por su escalabilidad, el reto de monitorizar entornos forestales excede a las posibilidades de este trabajo. Por otro lado, el sistema desarrollado proporciona diversas alarmas para avisar de situaciones anómalas ante las cuales se debe actuar.

Se puede comprobar que el objetivo principal del presente TFM se ha conseguido, es decir, se ha logrado realizar una red funcional IoT para la monitorización de ejemplares de coníferas y del entorno que les influye. Para ello se han diseñado e implementado diversos nodos tanto para la monitorización como para la gestión y transporte de las medidas tomadas por los diversos sensores. Sin embargo, no se ha podido llevar a cabo sobre un entorno forestal abierto, esto último, debido a su complejidad influyendo en los resultados tomados.

Se ha conseguido realizar un sistema de monitorización para entornos aislados, por tanto, se han utilizado distintas técnicas de comunicación a largas distancias y de bajo consumo. Además, se han realizado distintas técnicas para estudiar el rango de cobertura de la red implementada.

También se había planteado utilizar sistemas para la recopilación de los datos obtenidos. Para este propósito se han utilizado diversas BD de secuencias temporales para poder almacenar y representar todos los datos obtenidos por el sistema de monitorización. Asimismo, con algunas de las BD utilizadas se han realizado un procesamiento posterior de los datos, junto con sistemas de alarmas para conservar y proteger el bienestar del entorno en estudio.

Bajo mi punto de vista, este TFM, desde un primer momento, me ha hecho ver la complejidad de realizar un sistema de este calibre al sentir la necesidad de tener un equipo multidisciplinar. Ha sido una ardua tarea obtener unos conocimientos mínimos para poder englobar ramas del conocimiento tan dispares como son la electrónica, botánica, programación, ciencias medioambientales, IoT... Sin embargo, a mi parecer, se ha logrado un resultado satisfactorio.

7.2. Trabajo futuro

Para finalizar, se van a exponer algunas posibles mejoras, alternativas y desarrollos posteriores que pueden ampliar y continuar el presente trabajo.

Existen otras maneras de abordar el objetivo general de este proyecto. La opción elegida en este TFM ha sido monitorizar muestras del bosque (árboles) y del entorno. Una alternativa más extendida es la utilización de imágenes aéreas de amplias parcelas del bosque (inicialmente desde satélite, actualmente también imágenes hiperespectrales y vídeos, y medios aéreos como aviones y drones).

El trabajo realizado podría complementar la información proporcionada por estos sistemas de monitorización, a través de muestras representativas. Sería deseable hacer uso de algoritmos de *machine learning* o de inteligencia artificial que encuentren patrones relevantes a la hora de comparar resultados y comprender el estado del bosque en estudio.

Otro posible trabajo futuro sería llevar a cabo un estudio más exhaustivo y extendido a un mayor número de ejemplares, así como determinar el número de ejemplares necesarios para que la información que aporten sobre el estado del bosque sea relevante. Asimismo, sería muy conveniente aumentar el periodo de estudio para obtener medidas a lo largo de grandes periodos de tiempo, esto llevaría a observar la evolución de los ejemplares en todas las estaciones y en diferentes situaciones, además de ver cómo afectan estos parámetros al crecimiento del ejemplar.

Siguiendo con la idea expuesta con anterioridad, se podría realizar un sistema similar al implementado en este trabajo, pero específicamente guiado a observar la evolución de entornos donde se ha sufrido una catástrofe natural. De esta forma se puede observar la evolución ante distintas situaciones extremas y con esto obtener conclusiones que ayudasen a prevenir y conseguir la mejora de los terrenos una vez sucedida la desgracia. Técnicamente, sería interesante estudiar las protecciones que deben incorporarse a los distintos elementos que componen el sistema: por una parte, que puedan adaptarse al crecimiento y al deterioro del árbol sin alterar sus mediciones, y por otra, que resista las inclemencias meteorológicas y el ataque de la vegetación o de la fauna.

Otro posible trabajo sería analizar una forma de alimentación más eficiente y reducción del consumo, esenciales cuando el entorno es difícilmente accesible: desde un estudio más completo acerca de las baterías existentes que pudiesen alimentar al sistema completo durante el tiempo requerido, hasta el diseño de un sistema combinado de batería y fuente de energía renovable que proporcione una autonomía total, incluso el

CONCLUSIONES Y TRABAJO FUTURO

uso de sensores de muy bajo consumo capaces de alimentarse con la energía que produce el propio árbol. (64)

Por último, el sistema desarrollado es una primera aproximación que requiere ser optimizado para poder abordar un despliegue amplio y práctico. Esta tarea puede realizarse de manera compartimentada, mejorando cada una de sus partes:

- Estudio de la mejor forma de realizar los nodos centrales y la comunicación a largas distancias.
- Análisis de las posibles BD alternativas, sus capacidades y costes.
- Uso de algoritmos de *machine learning* o inteligencia artificial complejos para inferir nuevos resultados a partir de la cantidad de datos que se pueden obtener con este sistema.
- Implementación de nuevas alarmas que avisen de diversos escenarios para facilitar el trabajo a los expertos.

Bibliografía

1. Broecker, W. S. Climatic Change: Are We on the Brink of a Pronounced Global Warming? *Science* 1975, 189, 460-463.
2. Kalmus, P. *Being the Change: Live Well and Spark a Climate Revolution*; New Society Publishers: 2017.
3. Álvarez Herrera, M. Sistema de monitorización de la salud de los bosques de quercíneas, Universidad Carlos III de Madrid, 2023.
4. Bonells, J. E. EL MUNDO DE LAS CONIFERAS. 2017.
5. Klorane, F. The main characteristics of conifers.
<https://www.kloranebotanical.foundation/en/main-characteristics-conifers>.
6. Quiroz, J. Matusalén es árbol más viejo del mundo: te impresionará cuántas cosas ha visto pasar. <https://www.univision.com/explora/matusalen-es-arbol-mas-viejo-del-mundo-te-impresionara-cuantas-cosas-ha-visto-pasar> (accessed Sep 13, 2022).
7. Carracedo, C. Así es Hyperión, el árbol más grande del mundo.
<https://www.expansion.com/fueradeserie/viajes/2022/05/09/6270eed2e5fdea4e6d8b4660.html> (accessed Sep 13, 2022).
8. Nichols, M. Tall Trees. <https://education.nationalgeographic.org/resource/tall-trees> (accessed Sep 13, 2023).
9. Pérez Porto, J.; Gardey, A. Definición de coníferas - Definicion.de.
<https://definicion.de/coniferas/> (accessed Sep 13, 2022).
10. Illanes Jacinto, J. Los cultivos de Pinos en España. Interés económico e impacto ambiental. Universidad de Sevilla, 2022.
11. Fernández Muerza, A. Muerte de los bosques | Consumer. 2007.
12. Tyndall, J. On the Absorption and Radiation of Heat by Gases and Vapours, and on the Physical Connexion of Radiation, Absorption, and Conduction. *Philosophical Transactions of the Royal Society of London* 1861, 151, 1-36.
13. Arrhenius, S. On the Influence of Carbonic Acid in the Air upon the Temperature of the Ground. *Philosophical Magazine and Journal of Science* 1896, 41, 237-276.
14. Broecker, W. S. Are We on the Brink of a Pronounced Global Warming? *Science* 1975, 189, 460-463.
15. Martínez, M. ¿Sabes cuál fue el primer sensor conectado de la historia?
<https://www.nobbot.com/primer-sensor-conectado/> (accessed Sep 3, 2023).

BIBLIOGRAFÍA

16. Úbeda Miñarro, B. Apuntes de: Sistemas embebidos.
<https://www.um.es/documents/4874468/19345367/ssee-t01.pdf/4ea71f56-2950-4c3f-acbe-e7699e490f4e> (accessed Aug 30, 2023).
17. AnonymousSistemas empotrados > Información, Biografia, Archivo, Historia.
<https://es.wikidat.com> (accessed Aug 30, 2023).
18. seabrookewindows La Historia de los Sistemas Embebidos.
<https://www.seabrookewindows.com/JM7p4x2Ww/>.
19. AnonymousTecnología Inalámbrica.
<https://www.areatecnologia.com/informatica/tecnologia-inalambrica.html>
(accessed Sep 13, 2023).
20. Mekki, K.; Bajic, E.; Chaxel, F.; Meyer, F. A comparative study of LPWAN technologies for large-scale IoT deployment. 2019, 5, 1-7.
21. da Cruz, F. Hollerith 1890 Census Tabulator.
<http://www.columbia.edu/cu/computinghistory/census-tabulator.html> (accessed Sep 14, 2023).
22. Ministerio de Agricultura, Alimentación y Medio Ambiente Criterios e Indicadores de gestión forestal sostenible en los bosques España. *Ministerio de agricultura alimentación y medio ambiente* 2012, 79.
23. Ministerio para la transición ecológica y el reto demográfico ESTRATEGIA FORESTAL ESPAÑOLA HORIZONTE 2050. *Ministerio para la transición ecológica y el reto demográfico* 2022, 213.
24. Comisión Europea El Pacto Verde Europeo. *Comisión Europea* 2019, 28.
25. Comisión Europea Anexo de El Pacto Verde Europeo. *Comisión Europea* 2019, 5.
26. Naciones Unidas Acuerdo de París. *Naciones Unidas* 2016, 18.
27. Gobierno de España Agenda 2030. *Gobierno de España* 2015, 168.
28. Naciones Unidas CONVENIO SOBRE LA DIVERSIDAD BIOLÓGICA. *Naciones Unidas* 1992, 32.
29. Naciones Unidas PROTOCOLO DE KYOTO DE LA CONVENCIÓN MARCO DE LAS NACIONES UNIDAS SOBRE EL CAMBIO CLIMÁTICO. *Naciones Unidas* 1998, 25.
30. Zamora, R.; Pérez Luque, A. J.; Guerrero Alonso, P. D.; Moreno Llorca, R. A.; Merino Ceballos, M.; Ros Candeira, A. Uniendo macro y microclima en paisajes de montaña: una aproximación conceptual e instrumental. *Ecosistemas: Revista científica y técnica de ecología y medio ambiente* 2021, 30, 3.
31. González Briones, A.; Castellanos-Garzón, J.; Martín, Y.; Prieto, J.; Corchado Rodríguez, J. A Framework for Knowledge Discovery from Wireless Sensor

BIBLIOGRAFÍA

Networks in Rural Environments: A Crop Irrigation Systems Case Study. *Wireless Communications and Mobile Computing* 2018, 2018, 1-14.

32. Xiaoling NBSN95_NBSN95A NB-IoT Sensor Node User Manual - DRAGINO.
http://wiki.dragino.com/xwiki/bin/view/Main/User%20Manual%20for%20LoRaWA_N%20End%20Nodes/NBSN95_NBSN95A%20NB-IoT%20Sensor%20Node%20User%20Manual/ (accessed Aug 30, 2023).
33. Xiaoling LSN50 & LSN50-V2 - LoRaWAN Sensor Node User Manual - DRAGINO.
http://wiki.dragino.com/xwiki/bin/view/Main/User%20Manual%20for%20LoRaWA_N%20End%20Nodes/LSN50%20%26%20LSN50-V2%20-%20LoRaWAN%20Sensor%20Node%20User%20Manual/ (accessed Aug 30, 2023).
34. Anonymous Proyecto de Orden ETD/----/2021, de - - de - - - -, por la que se aprueba el CUADRO NACIONAL DE ATRIBUCIÓN DE FRECUENCIAS. *Ministerio de Asunto Económicos y Transformación Digital* 2014.
35. Anonymous MODULO TRANSCEPTOR LORA SX1278 RA-02 433 MHZ.
<https://www.vistronica.com/comunicaciones/wifi/modulo-transceptor-lora-sx1278-ra-02-433-mhz--detail.html> (accessed Sep 4, 2023).
36. Anonymous 433MHz Wireless 3dBi IPEX WiFi Spring Antenna for Ra-02 Module Lora OLED Board. <https://www.ebay.com/itm/183269788494> (accessed Sep 4, 2023).
37. Ruslan, Y. sx1278-LoRa-RaspberryPi. 2022.
38. Anonymous Ra-02 LoRa Module. https://docs.ai-thinker.com/_media/lora/docs/c048ps01a1_ra-02_product_specification_v1.1.pdf (accessed Sep 7, 2023).
39. Segovia-Vargas, D. INTRODUCTION TO ANTENNAS SRA 6-1. 2016.
40. Anonymous Recomendación UIT-R P.833-8 (09/2013) Atenuación debida a la vegetación. <https://www.itu.int> 2014.
41. Shelby, Z.; ARM; Universitaet Bremen, T.; Hartke, K.; Bormann, C. RFC 7252 (Jun 2014, Proposed STD, 112 pages): 1 of 6, p. 1 to 15. 2014.
42. Postel, J.; ISI RFC 768: User Datagram Protocol (Aug 1980, Internet Standard: 6, 3 pages). 1980.
43. OASIS MQTT Version 5.0. 2019.
44. Anonymous ISO/IEC 20922:2016(en), Information technology — Message Queuing Telemetry Transport (MQTT) v3.1.1. ISO/IEC 2016, 1.
45. W.Eddy, E.; MTI Systems RFC 9293 (Aug 2022, Informational, 98 pages). 2022.

46. Huang, D. NBSN95. <https://docs.influxdata.com/influxdb/cloud/query-data/get-started/> (accessed Sep 15, 2023).
47. Von Holck, F. NB-IoT roll out underway in Denmark. <https://beacontower.io/nb-iot-roll-out-underway-in-denmark/> (accessed Sep 2, 2023).
48. Mikrotik User Manual KNOT LR8 kit.
<https://help.mikrotik.com/docs/pages/viewpage.action?pageId=41680915>
 (accessed Sep 02, 2023).
49. Anonymous2JW0315-868-C675B. <https://www.2j-antennas.com/media/original/datasheets/2jw0315-868-c675b.pdf> (accessed Sep 7, 2023).
50. AnonymousIot gateway lora 8 mikrotik knot lr8 kit rb924ir-2nd-bt5&bg77&r11e-...
https://qrinet.com/redes-y-comunicaciones-routers-puntos-acceso-antenas-antenas-wifi-cables-antenas/145279-iot-gateway-lora-8-mikrotik-knot-lr8-kit-rb924ir-2nd-bt5bg77r11e-lr8-mikrotik-rb924ir-2nd-bt5bg77r11e-lr8.html?gclid=Cj0KCQjwusunBhCYARIaAFBsUP_XhFpDSM1HJclHx8UpA-gbUgcWhCiZsptRN5PCcj_F_nY0LNp0Kc0aAqHzEALw_wcB (accessed Sep 2, 2023).
51. Anonymous2JW0315-868-C675B - 868 MHz ISM Narrowband Antenna | 2J Antennas. <https://www.2j-antennas.com/antennas/single-external-antennas/2jw0315-868-c675b-868-mhz-ism-narrowband-antenna/428> (accessed Sep 7, 2023).
52. InfluxData InfluxDB.
 It's About Time. <https://www.influxdata.com> (accessed Sep 09, 2023).
53. AnonymousManage buckets in InfluxDB | InfluxDB OSS 2.0 Documentation.
<https://docs.influxdata.com/influxdb/v2.0/organizations/buckets/> (accessed Sep 9, 2023).
54. AnonymousCreate an API token in InfluxDB | InfluxDB Cloud (TSM) Documentation. <https://docs.influxdata.com/influxdb/cloud/security/tokens/create-token/> (accessed Sep 9, 2023).
55. AnonymousThe Things Network. <https://www.thethingsnetwork.org> (accessed Sep 08, 2023).
56. Assistant, H. Home Assistant. <https://www.home-assistant.io/> (accessed Sep 9, 2023).
57. InfluxData Get started with Flux | InfluxDB Cloud (TSM) Documentation.
<https://docs.influxdata.com/influxdb/cloud/query-data/get-started/> (accessed Sep 11, 2023).
58. Postolache, O. A. In *Smart Sensing Systems and AI for Precision Agriculture in Climate Changes Era*; 859 0364 6756;
https://us06web.zoom.us/w/85903646756?tk=WYVYHWYLGcUsI_1jVPUqWfZYHtur7NGsfTct3R3KacY.DQMAAAUAEGgJBZMTUhDclB1bVNCdW13bWZ

BIBLIOGRAFÍA

https://zoom.us/j/1bjhtWHRnAAAAAAAAAAAAAAA&pwd=dFdPdDJSaFhQOXIvSnNnOEIRY3dQZz09&uuid=WN_uoVZHPvZScewe9F3jgIhw: zoom.us, Jul 26, 2023.

59. Emery, N. Foliar uptake of fog in coastal California shrub species. *Oecologia* 2016, 182.
60. Jean, J. I'm living in a carbon bubble. Literally. | by Joel Jean | Medium. 2016.
61. Masats, J. El pH del suelo para el cultivo de las plantas. <https://www.botanical-online.com/cultivo/suelo-tipos-ph> (accessed Aug 30, 2023).
62. Mataix-Solera, J.; Navarro-Pedreño, J.; Guerrero, C.; Gómez Lucas, I.; Marco, B.; Mataix, J. In *Effects of an experimental fire on soil microbial populations in a Mediterranean environment; Man and Soil at the Third Millennium*; Geoforma Ediciones: 2002; pp 1607-1614.
63. Meteos Sierra Gráficos interactivos | Carabanchel. <https://meteos Sierra.com/estaciones/carabanchel/> (accessed Sep 11, 2023).
64. Love, C. J.; Zhang, S.; Mershin, A. Source of Sustained Voltage Difference between the Xylem of a Potted Ficus benjamina Tree and Its Soil. *PLOS ONE* 2008, 3, e2963.

Apéndice A

En este apéndice inicial, se exponen múltiples de los distintos sensores y estaciones meteorológicas que han sido objeto de estudio para el desarrollo de los TFM realizados. Se recopila información importante para la elección de estos componentes como el precio, el tiempo de entrega, enlace web...

Temperatura estomática

Temperatura estoma

Sensor	Precio	Tiempo	Observaciones	Contacto	Enlace
LT-1M	275 €	3-6 semanas		info@phyto-sensor.com	http://phyto-sensor.com/LT-1M#:~:text=The%20LT-1M%20sensor%20about%20millimeter%20in%20diameter
LT-2M				solfranc@solfra...	http://www.solfranc.com/productos/wp-content/uploads/2014/11/lt_2m_sensor_de_temperatura_de_hoja_solfranc.es.pdf
LT-IRM				https://edaphic.com.au/contact-us/	https://edaphic.com.au/temperature/infrared-temperature-sensor/
LS-31	295 €		DESCATALOGADO		https://www.alphamega-electronics.com/en/agricultura/3967-ls-31-broad-leaf-temperature-sensor-for-broad-leaves-weight-2-gr.html
Ls-40	330 €				https://www.alphamega-electronics.com/en/agricultura/3967-ls-31-broad-leaf-temperature-sensor-for-broad-leaves-weight-2-gr.html
LAT-B3	225 €	1-2 semanas	Se puede utilizar un datalogger DL18 precio 188€	info@ecomatik.de	https://ecomatik.de/en/products/leaf-sensors/leaf-temperature/

Humedad de la hoja

Humedad emitida por la hoja higrómetro					
Sensor	Precio	Tiempo	Observaciones	Contacto	Enlace
HM102	230€ + IVA		s-40		http://www.ejisiberica.com/Hig%F3metros%20especiales/HM102.html
Higrómetro de punto de rocío WP4C	DESCATALOGADO		DESCATALOGADO		https://www.lab-ferrer.com/higrometro-de-punto-de-rocio-wp4c/
LWS			CONTACTO PROFESOR		https://www.evvos.com/product/leaf-wetness-sensor/
RK300-04	5 uds =421,20€	7-9dias laborables	Mínimo 5 unidades 340\$+ 78\$ gastos envío =421,20€	https://www.rikasensor.com/inquire/cart	https://www.rikasensor.com/rk300-04-leaf-wetness-sensor-humidity-sensor.html?gclid=EAIAjQobChMrJLmx_WC-QIV6gyLCh0w2z79EAAAYAiAAEgKkBvD_BwE
Leaf Wetness Sensor	140 €				https://www.darrera.com/wp/en/product/6420-leaf-wetness-sensor/
PESSL	339,12 €		sugieren el NMETOS200 con sensores temperatura, humedad, pluviómetro precio 943,79€	sales@metos.at	https://metos.at/es/portfolio/leaf-wetness/
NMETOS200	943,78 €		sugerencia empresa	sales@metos.at	https://metos.at/es/portfolio/leaf-wetness/
LWS-L	192 €			https://www.campbellsci.cc/request-quote	https://www.campbellsci.cc/lws
HD3901	250 €	10-12 semanas		https://www.deltaohm.com/product/hd3901-leaf-wetness-sensor/	https://www.deltaohm.com/product/hd3901-leaf-wetness-sensor/
PYTHOS31	180,00 €		Lo lleva LabFerrer en España	https://www.metergroup.com/en/meter-environment/products/phytos-31-leaf-wetness-sensor	https://www.metergroup.com/en/meter-environment/products/phytos-31-leaf-wetness-sensor

Diámetro del tallo, peciolo y tronco

Diámetro de tallo, peciolo y tronco

Sensor	Precio	Observaciones	Contacto	Enlace
Dendrómetros DS25	308 €			http://www.gisiberica.com/dend%F3metro/manual%20ds21.htm
Dendrómetro DS26	529 €			http://www.gisiberica.com/dend%F3metro/MANUAL%20DS26.htm
Dendrómetro MMM-Tech	DESCATALOGADO	DESCATALOGADO		https://www.mmm-tech.de/de/downloads?task=download_send&id=88&catid=3&m=0
DD-S	350 €			https://www.mmm-tech.de/de/downloads?task=download_send&id=88&catid=3&m=0
DF	365 €			https://www.mmm-tech.de/de/downloads?task=download_send&id=88&catid=3&m=0
DC-2	375 €			https://www.mmm-tech.de/de/downloads?task=download_send&id=88&catid=3&m=0
DL-18	425 €			https://www.mmm-tech.de/de/downloads?task=download_send&id=88&catid=3&m=0
Dendrómetro SD-5(6,10)T			https://edaphic.com.au/contact-us/	https://www.alphaomega-electronics.com/es/sensores-estaciones/1568-dendrometro-para-frutas-y-verduras-diametro-0-11-cm-df-11.html
Dendrómetro SDI12			https://edaphic.com.au/contact-us/	https://www.implexx.io/wp-content/uploads/2021/05/Impléxx-SD-Stem-Dendrometer-Manual.pdf
LVDT Solartron Metrology, recorrido ±25mm	492,37 €			https://es.rs-online.com/web/p/sensores-de-control-de-movimiento/7271329
LVDT Solartron Metrology, recorrido ±100mm	1.058 €			https://es.rs-online.com/web/p/sensores-de-control-de-movimiento/7271357
LVDT Analog Devices, recorrido ±2.5 → ±50.8mm	147,31 €			https://es.rs-online.com/web/p/sensores-de-control-de-movimiento/8031591

Tamaño de fruto

Tamaño de fruto		
Sensor	Precio	Enlace
Dendrómetro DS50	497 €	http://www.gisiberica.com/dend%F3metro/DS50.html
Dendrómetro DS50A	481 €	http://gisiberica.com/CODIGOS/DS50.HTM

Sensores de Humedad

Clorofila				
Sensor	Precio	Observaciones	Contacto	Enlace
Cyclops-7 de Turner Designs	3.000€-5.000€		https://www.ott.com/es-es/productos/sensores-179/sensor-de-clorofila-a-347/	https://www.ott.com/es-es/productos/sensores-179/sensor-de-clorofila-a-347/
MC-100	2.385 €			https://www.alphaomega-electronics.com/es/concentracion-clorofila/1365-mc-100-medidor-de-concentracion-de-clorofila-mol-m-2-con-gps-interno.html
Micro PAM	4 cabezales 15.675€	4 cabezales sin sistema de Adquisición	https://www.walz.com/support/contact_company.html	https://www.walz.com/products/chl_p700/micro-pam/introduction.html
Micro PAM	3cabazales 21.340€ 3cabazales 17830€ 4 cabezales 21665€	versión en línea 3cabazales 17830€ 4 cabezales 21665€	https://www.walz.com/support/contact_company.html	https://www.walz.com/products/chl_p700/micro-pam/introduction.html

Flujo de savia

Flujo de savia					
Sensor	Precio	Tiempo	Observaciones	Contacto	Enlace
Sensor de Flujo de Savia SFM1				https://www.lab-ferrer.com/contactar/	http://www.lapacacr.com/images/ICT/Sensor-Flujo-Savia-SFM-Espanol.pdf
Sensor de Flujo de Savia HRM				https://www.lab-ferrer.com/contactar/	https://www.lab-ferrer.com/sensor-de-flujo-de-savia-sfm1-y-hrm/#:~:text=El%20sensor%20SFM1%20es%20capaz,calor%20atendiendo%20al%20m%C3%A9todo%20HRM
SF-G Sensor Flujo de Savia Ecomatik (2 agujas)	339 €				https://www.alphomega-electronics.com/es/saviaflujo-de-savia/815-sf-g-sensor-flujo-de-savia-ecomatik-2-agujas.html
East30 Sensors				https://www.lab-ferrer.com/contactar/	https://www.east30sensors.com
SGEX-16	436 €		CABLES DE EXTENSION 15 m precio 200€ + batería y placa solar sin precio	https://www.idelsur.com/?page_id=1889 (gema@lab-ferrer.com)	https://www.idelsur.com/?cat=269
SGEX-19	442 €		CABLES DE EXTENSION 22 m precio 250€ + batería y placa solar sin precio	https://www.idelsur.com/?page_id=1889 (gema@lab-ferrer.com)	https://www.idelsur.com/?cat=269
SGEX-25	455 €		CABLES DE EXTENSION 28 m precio 300€ + batería y placa solar sin precio	https://www.idelsur.com/?page_id=1889 (gema@lab-ferrer.com)	https://www.idelsur.com/?cat=269
Sensores de Flujo de Savia EXO-Skin FLOW32A-1K	9.800 €		TODO INCLUIDO	https://www.idelsur.com/?page_id=1889 (gema@lab-ferrer.com)	https://www.idelsur.com/?cat=269
SF-L	580 €				https://www.alphomega-electronics.com/es/sensores-y-sondas/816-sf-l-sensor-de-flujo-de-savia-ecomatik-4-agujas.html
FRUITION	ERROR	ERROR	ERROR Envío solicitud	https://fruitionsciences.com/es/sap-flow-irrigation-sensors	https://fruitionsciences.com/es/sap-flow-irrigation-sensors
HFD (HFD8-50)	0 €		imagino precio erróneo consultar	https://www.cotecno.cl/solicitar-cotizacion/	https://www.cotecno.cl/sensor-de-flujo-de-savia-hfd-hfd8-100-hfd8-50/
HFD (HFD8-100)	0 €		imagino precio erróneo consultar	https://www.cotecno.cl/solicitar-cotizacion/	https://www.cotecno.cl/sensor-de-flujo-de-savia-hfd-hfd8-100-hfd8-50/

Oquedad del árbol

Oquedad del árbol (A través de sonido percutido o resistencia eléctrica de la madera)					
Sensor	Precio	Tiempo	Observaciones	Contacto	Enlace
PICUS Sonic	ERROR	ERROR	ERROR Envío solicitud	info@argus-electronic.de	https://www.agriexpo.online/es/prod/argus-electronic-gmbh/product-182221-61045.html
ARBOTOM	8.000 €		mínimo 8 unidades a 1.000€/und	info@rinntech.com	https://www.agriexpo.online/es/prod/rinntech/product-182572-64702.html
ArborSonic 3D				office@fakopp.com	https://www.agriexpo.online/es/prod/fakopp-bt/product-182571-64678.html
Tomografía sónica	650\$- 654,39€				https://www.newdayarborist.com/tree-diagnostic-services/sonic-tomography/#:~:text=We%20are%20proud%20to%20be,per%20tree%20plus%20travel%20time.

Presión del jugo celular sobre la hoja

Presión del jugo celular sobre la hoja			
Sensor	Precio	Contacto	Enlace
R16 SENSOR DE TURGENCIA DE HOJA	6 und 6.200 €	https://docs.google.com/forms/d/e/1FAIpQLSfTXqnFGh1kltckYkK0mGog6HLopfEy5X_jnkyc_LWhQeTCxg/viewform	https://apgefert.greenfield.farm/wp-content/uploads/2020/09/R16.pdf
Sensor de Agua Yara		marcelodip@seedmech.com	https://www.seedmech.com/producto/sensor-de-agua-yara/

Oxígeno en el suelo

Oxígeno en suelo

Sensor	Precio	Tiempo	Observaciones	Contacto	Enlace
Sensor de oxígeno gas en el aire y medios porosos Modelo SO de Apogee Instruments				https://www.lab-ferrer.com/contactar/	https://www.lab-ferrer.com/sensor-de-oxigeno-gas-en-el-aire-y-medios-porosos-so/
SOM1 Soil Oxygen Meter	1650\$-1657,19€			https://ictinternational.com/products/som1/som1-soil-oxygen-meter/	https://ictinternational.com/products/som1/som1-soil-oxygen-meter/
MIIJ-03	960\$-964.18€(ICT)			https://edaphic.com.au/contact-us/	https://edaphic.com.au/oxygen/es-o2-soil-oxygen-sensor/
RK520-02				https://www.rikasensor.com/inquire/cart	https://www.rikasensor.com/professional-soil-oxygen-sensor-industry-for-soil-monitoring.html
Soil oxygen content analysis system	Instalado 5.000€	< 1 mes	Instalación fija	https://www.rovaleijkelkamp.com/our-company/contact/TLF:648312373	https://www.rovaleijkelkamp.com/products/field-measuring-equipment/soil-oxygen/analysis/soil-oxygen-content-analysis-system/
ICTO2	ERROR	ERROR	ERROR Envío solicitud	https://www.htsperu.com.pe/ict-international/sensor-de-oxigeno-del-suelo-ict2-ict-international?page=5&limit=75	https://geomor.com.pl/wp-content/uploads/2020/04/ICTO2_Soil_Oxygen_Sensor.pdf
CS511-L	NO VTA EUROPA		No se distribuye en Europa	https://www.campbellsci.es/questions	https://www.campbellsci.com/cs511-l

Oxígeno en la hoja

Oxígeno en hoja/aire:

Sensores	Precio	Tiempo	Observaciones	Enlace
Gravity: I2C Oxygen Sensor (Raspberry Pi)	63,99 €			https://www.pi-shop.ch/gravity-i2c-oxygen-sensor
Seeed Grove - Oxígeno/Oxygen Sensor (MIX8410)	39,93 €			https://www.seeedstudio.com/Grove-Oxygen-Sensor-MIX8410-p-4697.html
EZO-O2™ Embedded Oxygen Sensor	100,99\$-101,47€		No valido, útil solo industria	https://atlas-scientific.com/probes/oxygen-sensor/#
o Sensor de oxígeno ME2-O2	35 €	1-5 días		https://es.aliexpress.com/item/32336030733.html?UTABTest=aliabtest300081_422161&_randl_currency=EUR&_randl_shipto=ES&src=google
MIX8410-O2	21\$-21,10€	5 días		https://spanish.alibaba.com/product-detail/MIX8410-O2-O2-oxygen-sensor-oxygen-1600221990224.html

Potencial hidráulico

Potencial hidráulico

Sensor	Precio	Contacto	Enlace
PSY1 Psicrómetro	3390\$-3404,76€	https://ictinternational.com/products/som1/som1-soil-oxygen-meter/	https://ictinternational.com/products/psy1/psy1-stem-psychrometer/

Dióxido de carbono en la hoja

Dióxido de carbono en la hoja/aire

Sensor	Precio	Observaciones	Contacto	Enlace
SEN0159	56,00 €			https://es.farnell.com/dfrobot/sen0159/sensor-anal-gico-gas-co2-placa/dp/3517879?cjevent=3f2e73c50b3e11ed814e03550a18050f&cidata=MXxZfDB8WXww&CMP=AFC-CJ-ES-1765328&gross_price=true&source=CJ#anchorTechnicalDOCS
CO2 Gas Sensor	377\$-378,78€			https://www.vernier.com/product/co2-gas-sensor/
				https://pubs.acs.org/doi/10.1021/acs.jchemed.8b00473
EZO-CO2	166,99\$-167,78€			https://atlas-scientific.com/probes/co2-sensor/
RK300-03B	276.78€	Minimo 2 unidades 138\$ unidas	https://www.rikasensor.com/inquire/cart	https://www.rikasensor.com/rk300-03-co2-concentration-sensor.html?gclid=Cj0KCQjw2_OWBhDqARIhAAUNTTFUhG2x3CsUkHPF9gru_EoQsYQYddnUTTXmtSFVA0u-HKst9SvAz5saAnPIEALw_wcB
Wireless Carbon Dioxide CO2 Sensor	257,71 €			https://heliot.ca/wireless-carbon-dioxide-co2-sensor/
MH-Z19C NDIR CO2 Sensor	45 €			https://www.winsen-sensor.com/sensors/co2-sensor/mh-z19c.html

Dióxido de carbono en el suelo

Dióxido de carbono en el suelo			
Sensor	Precio	Contacto	Enlace
SRS1000T		https://www.lab-ferrer.com/contactar/	https://www.lab-ferrer.com/sistema-para-medir-la-respiracion-del-suelo-srs/
SRS2000T		https://www.lab-ferrer.com/contactar/	https://www.lab-ferrer.com/sistema-para-medir-la-respiracion-del-suelo-srs/
eosGP CO2 sensor		https://edaphic.com.au/contact-us/	https://edaphic.com.au/products/soils/forerunner-gp-sensor/
eosFD	3.750 €	https://eosense.com/products/eosfd-soil-co2-sensor/#	https://eosense.com/products/eosfd-soil-co2-sensor/
SRC-2		https://coltein.com/producto/camara-de-respiracion-del-suelo-src-2/	https://coltein.com/producto/camara-de-respiracion-del-suelo-src-2/
LI-COR (CO2, CH4, and N2O)		https://www.licor.com/env/products/soil_flux/quote	https://www.licor.com/env/products/soil_flux/
LI-COR: 8250, 8250-01, 8200-104, 8200-104C		https://www.licor.com/env/products/soil_flux/quote	https://www.licor.com/env/products/soil_flux/long-term?clid=EA1a1QobChM12KS86fmI-QIVaoODBxlu7Qi7EAAYASAAEeLpEvD_BwE
CFLUX-1		https://ppsystems.com/quote-request-pp-systems/	http://www.hanatech-instruments.com/product/cflux-1/
MH-Z16		https://www.winsen-sensor.com/sensors/co2-sensor/mh-z16.html	https://www.winsen-sensor.com/sensors/co2-sensor/mh-z16.html

NPK

NPK (para saber fertilidad del suelo)				
Sensor	Precio	Tiempo	Contacto	Enlace
RS485 npk tester	169.78\$-170,58€			https://www.amazon.com/-/es/Taidacent-Sensores-Detector-Nitrógeno-nitrógeno/dp/B08MXXSP59
Soil NPK Sensor	52,3\$-52,55€			https://www.renkeer.com/product/soil-npk-sensor/
Replacement Sensor for Horiba LAQUAtwin NO3 Meters	160\$-160,76€			https://www.agriculturesolutions.com/replacement-sensor-for-horiba-laqua-no3-meters
Teralytic	1.200\$-1.205,10€	PEDIDO 2023		https://teralytic.com/index.html
EVTSCAN Sensor NPK	129,99 €			www.amazon.es/EVTSCAN-Fertilizante-Inteligente-nutrientes-precisión/dp/B0957874WZ/ref=sr_1_5
Soil NPK Sensor Revalcon			info@revalcon.com	https://revalcon.com/soil-npk-sensor/
All in One Soil Sensor NPK pH Ec Sensor for Agriculture Monitoring	190\$-190,90€			https://rainbowsensor.en.made-in-china.com/product/CFiApvTzhRYe/China-All-in-One-Soil-Sensor-NPK-pH-Ec-Sensor-for-Agriculture-Monitoring.html

pH

pH del suelo				
Sensor	Precio	Tiempo	Observaciones	Enlace
SEN-10972	DESCATALOGADO		DESCATALOGADO	https://www.digikey.es/es/products/detail/sparkfun-electronics/SEN-10972/5766911
SEN0161	34,82 €	DESCARTADO	MIDE PH SOLO AGUA	es.farnell.com/dfrobot/sen0161/kit-sensor-medidor-anal-gico-ph/dp/2946120
3 en 1, medidor luz solar, humedad del suelo, Detector de acidez, Monitor de PH	7,09 €			es.aliexpress.com/item/1005004063776178.html
Sonda de pH de alta precisión	26,27 €			www.amazon.es/adquisici%C3%B3n-Suministros-industrial-profesional-hidrop%C3%B3nicas/dp/B08S3N4212/ref=asc_df_B08S3N4212/
Soil detector	4,45 €			https://es.aliexpress.com/item/1005004300424170.html
SEN0249	116,85 €			es.farnell.com/dfrobot/sen0249/kit-sensor-ph-punta-lanza-anal/dp/3517936
DFRobot	84,75 €		DESCARTADO	es.rs-online.com/web/p/placas-y-kits-compatibles-con-arduino/2163779
SEN0169	67,17 €		DESCARTADO	https://es.farnell.com/dfrobot/sen0169/kit-sensor-anal-gico-ph-medidor/dp/3517876?gross_price=true
Sonda de pH ProMinent 150702	281,71 €		DESCARTADO	es.rs-online.com/web/p/electrodos-de-ph/0531292

Humedad del suelo

Humedad del suelo			
Sensor	Precio	Observaciones	Enlace
HPP809A033	143,32 €	DESCARTADO	https://www.digikey.es/es/products/detail/te-connectivity-measurement-specialties/HPP809A033/223-1589-ND/5277325
101990792	220,10 €		https://www.digikey.es/es/products/detail/seeed-technology-co-ltd/101990792/14672153
314990620	79,74 €		www.digikey.es/es/products/detail/seeed-technology-co-ltd/314990620/16570933
PIM520	13,97 €		https://www.digikey.es/es/products/detail/pimoroni-ltd/PIM520/13537122
28092	5,37€		https://www.digikey.es/es/products/detail/parallax-inc/28092/10453796

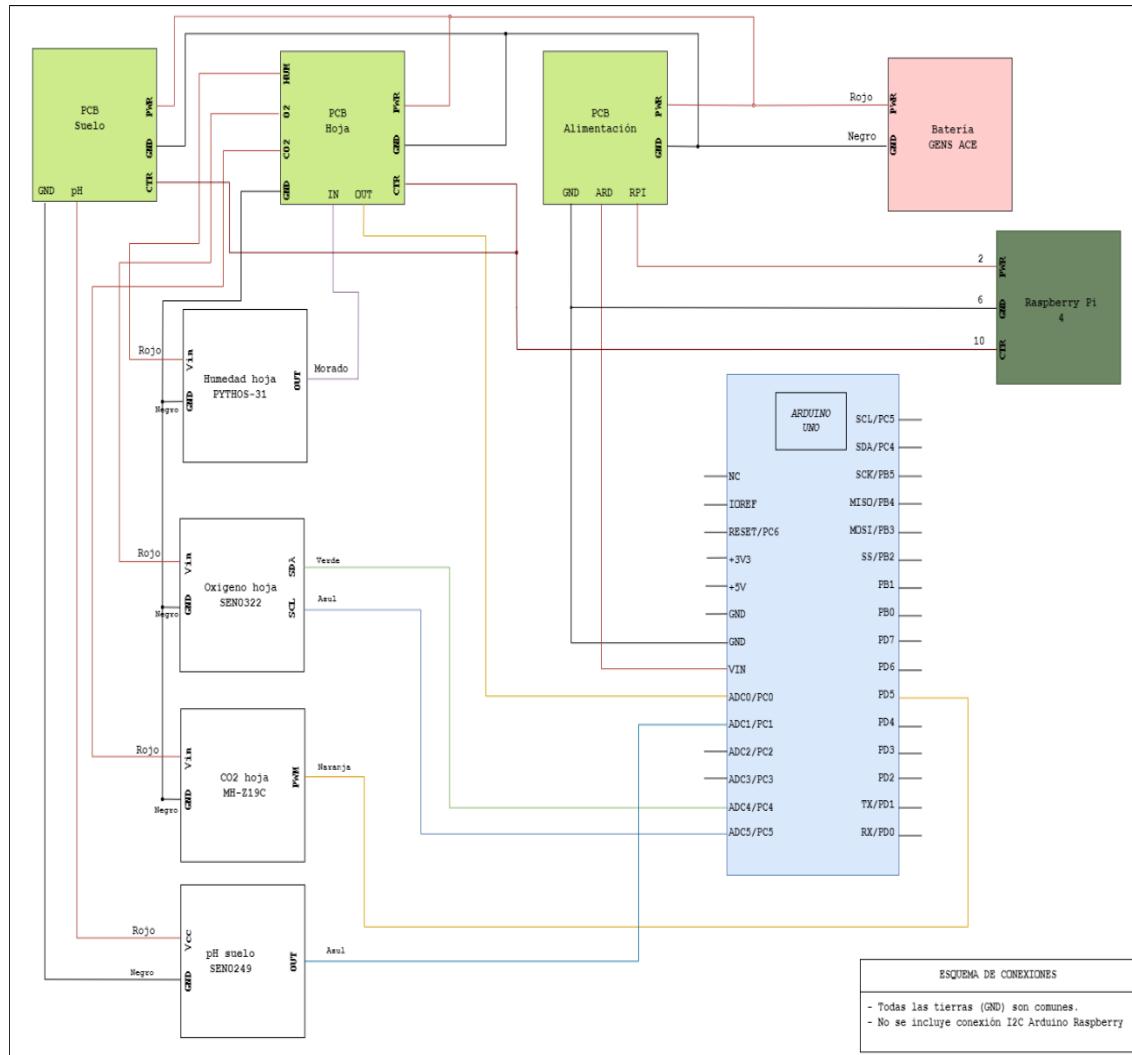
Estación meteorológica

Estación meteorológica		
Sensor	Precio	Enlace
WLAN Sainlogic	169 €	https://www.amazon.es/sainlogic-WS3500-Estaci%C3%B3n%20meteol%C3%B3gica-Inal%C3%A1mbrica/dp/B081F1VTYT/ref=asc_df_B081F1VTYT/
Bresser Wetterstation Funk mit Außensensor Wetter Center 5-in-	49 €	https://www.amazon.es/Bresser-Centro-meteorol%C3%B3gico-Estaci%C3%B3n%20meteol%C3%B3gica/dp/B07DW23KWF/ref=asc_df_B07DW23KWF/
Estación meteorológica Sainlogic profi WLAN, estación meteorológica inteligente de Internet WiFi	206,11 €	https://www.amazon.es/sainlogic-FT0300-Estaci%C3%B3n%20meteol%C3%B3gica-Negro/dp/B0836FVVYZ/ref=asc_df_B0836FVVYZ/
Sainlogic WS 3500 Plus. Estación Meteorológica Inalámbrica Profesional 10	185 €	https://estacionmeteorologicaproyecto.com/sainlogic/sainlogic-ws3500-10-en-1/
froggit Estación meteorológica inalámbrica WH6000 Pro con conexión por Internet, incluye sensor exterior combinado 7 en 1	260 €	https://www.amazon.es/dp/B08T1571GL
Estación Meteorológica WLAN Sainlogic Profesional, Estación Meteorológica WiFi con Sensor Para Exteriores Alimentada Por Energía Solar, Medidor de Lluvia, Pronóstico del Tiempo	185 €	https://www.amazon.es/dp/B07TVMQ2CK
Froggit WH3600 Funk Estación meteorológica WiFi WLAN	150 €	https://www.amazon.es/dp/B08H8H1MHQ
Froggit Estación meteorológica WH3000 SE (Edition 2018) a Internet Wi-Fi, App	149,99 €	https://www.amazon.es/dp/B06X6JPJ6Q
Sainlogic Estación meteorológica inalámbrica con 3 sensores exteriores, previsión meteorológica, pantalla a color, termómetro inalámbrico con sensor exterior.	65,37 €	https://www.amazon.es/dp/B07V6FYTKL

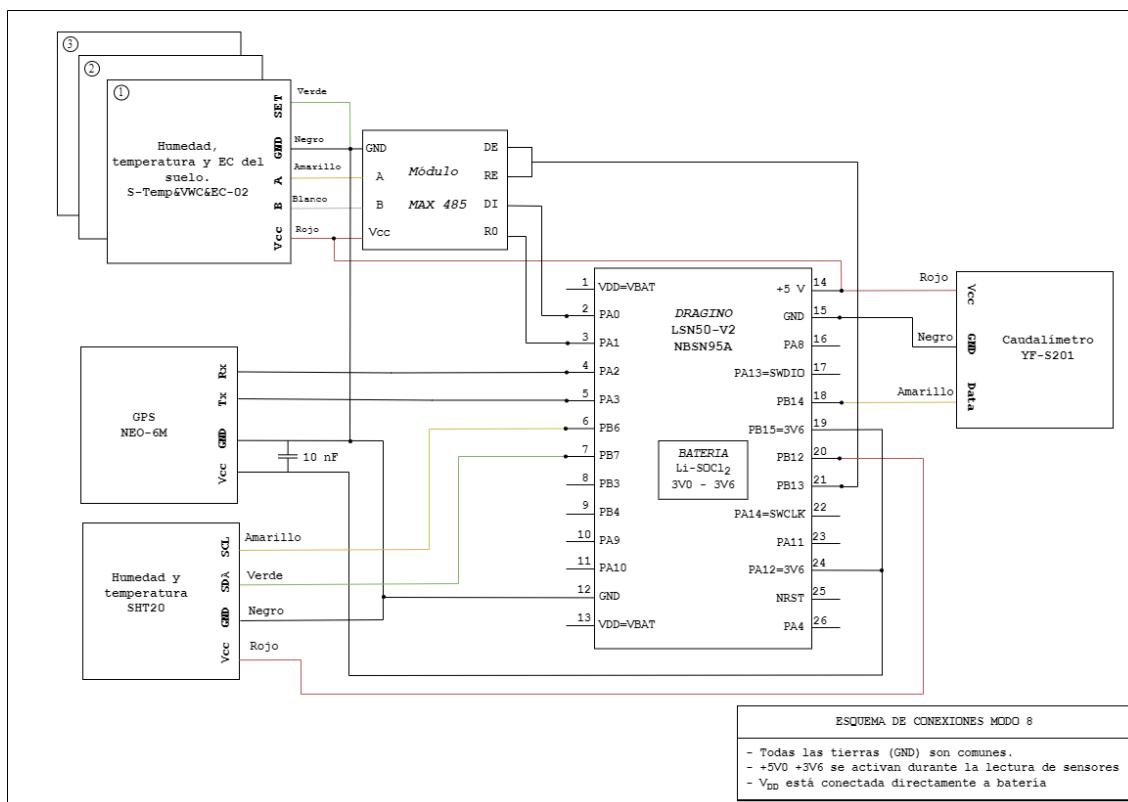
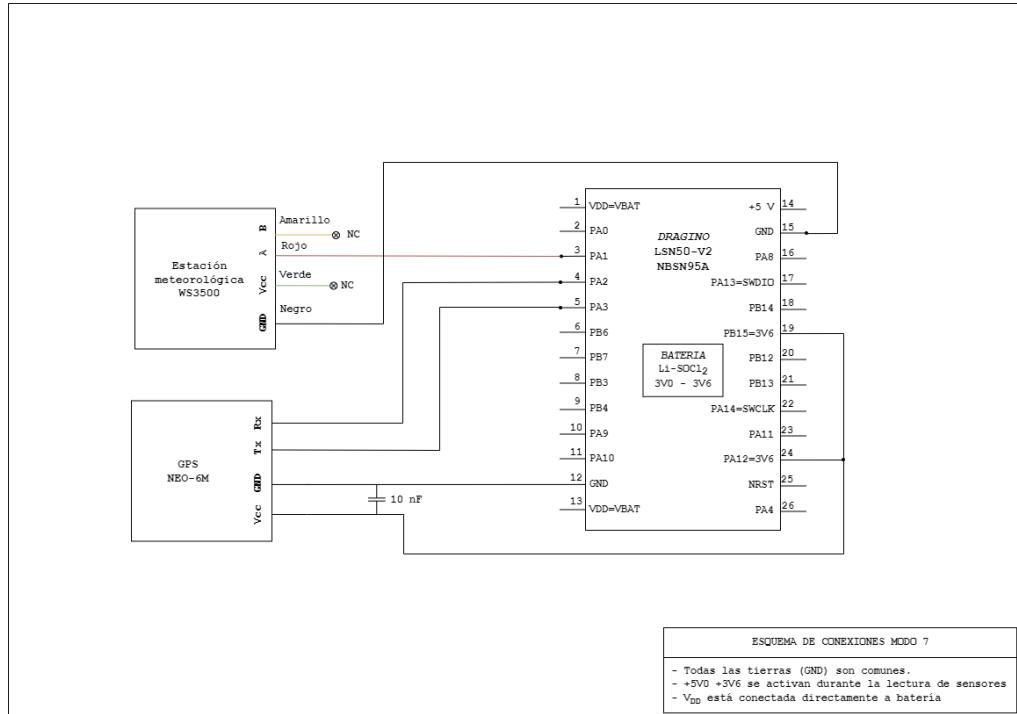
Apéndice B

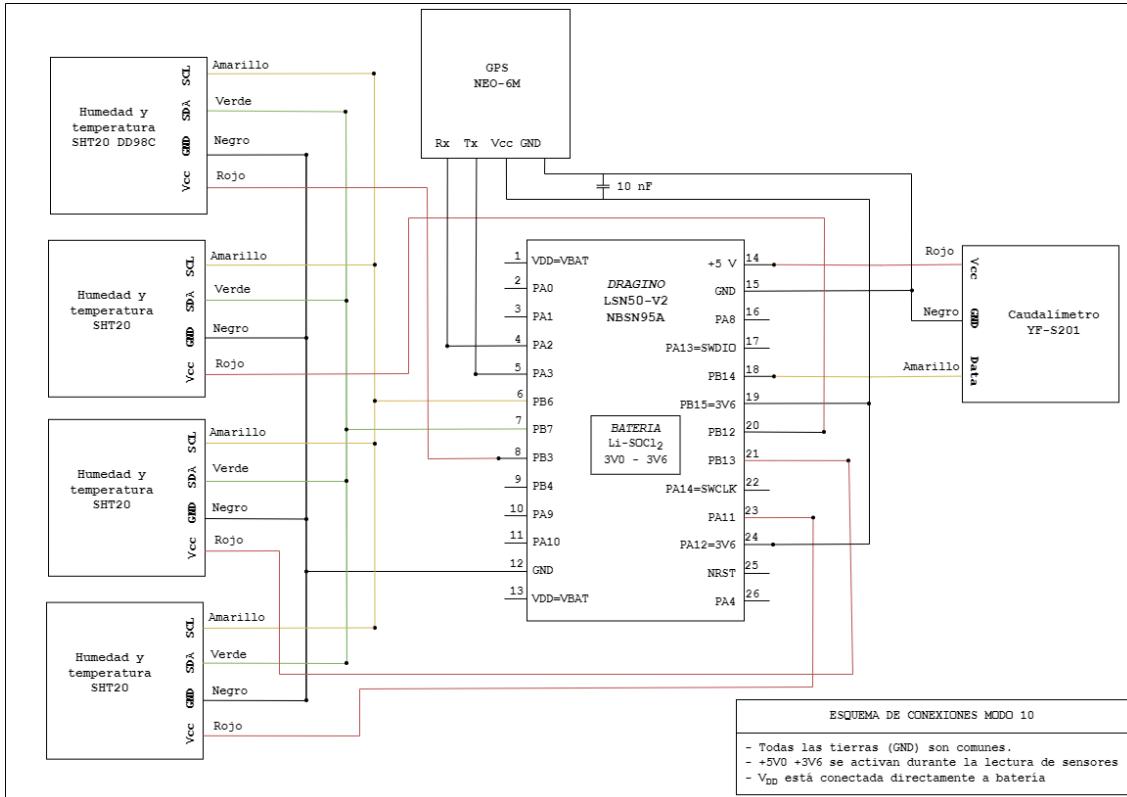
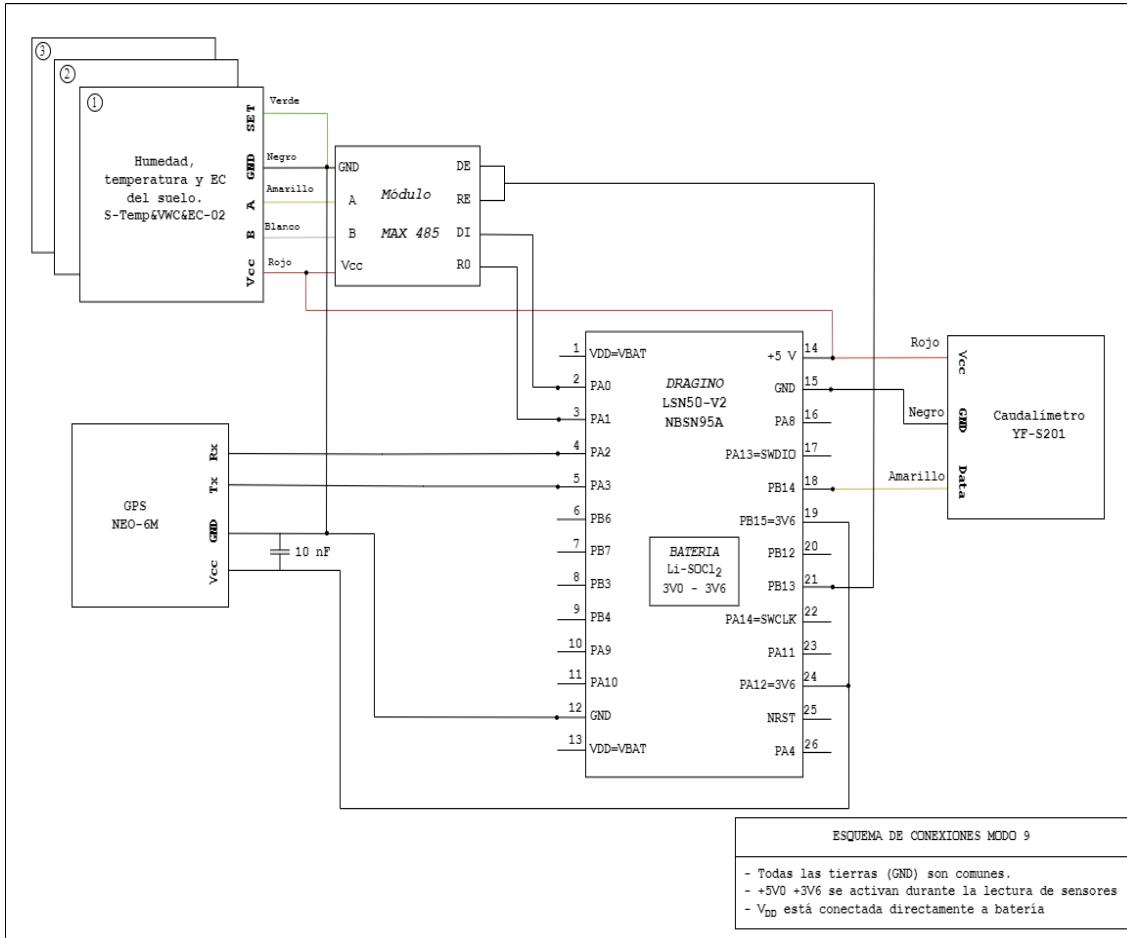
En este apartado se exponen los esquemas electrónicos que se han utilizado para conectar de forma adecuada los dos primeros tipos de nodos de la red IoT implementada.

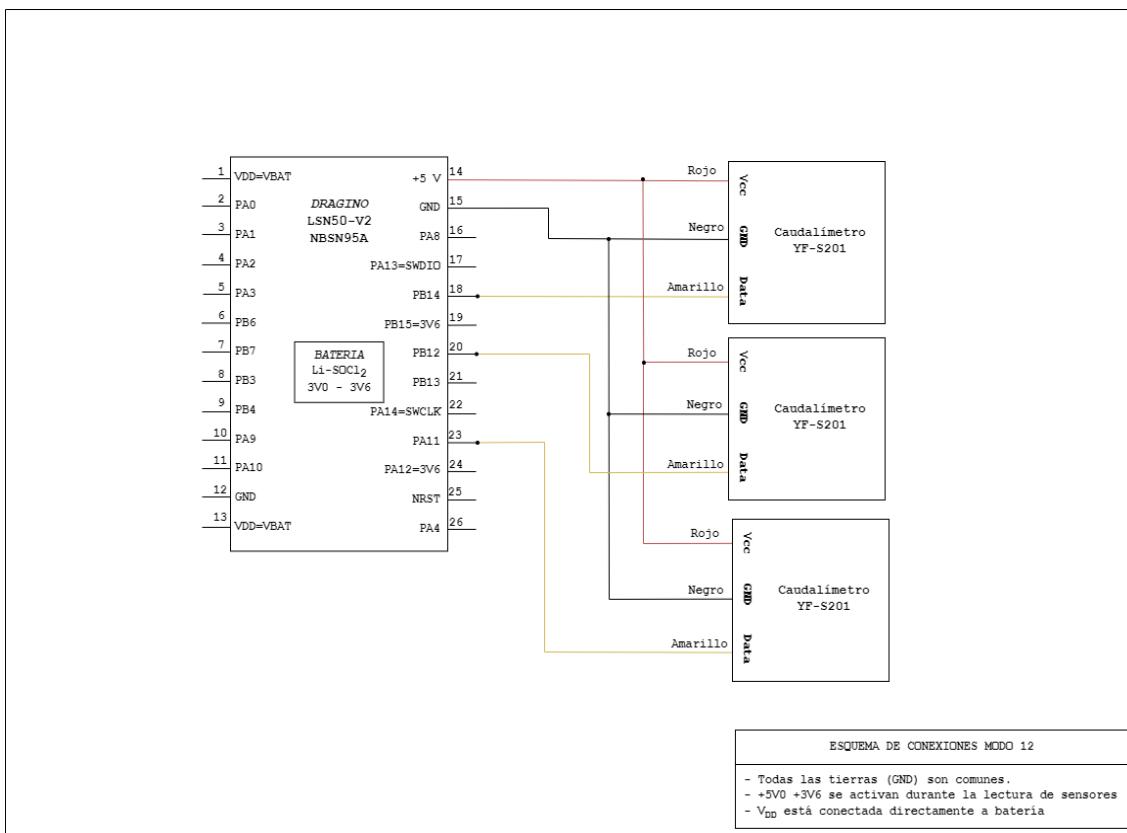
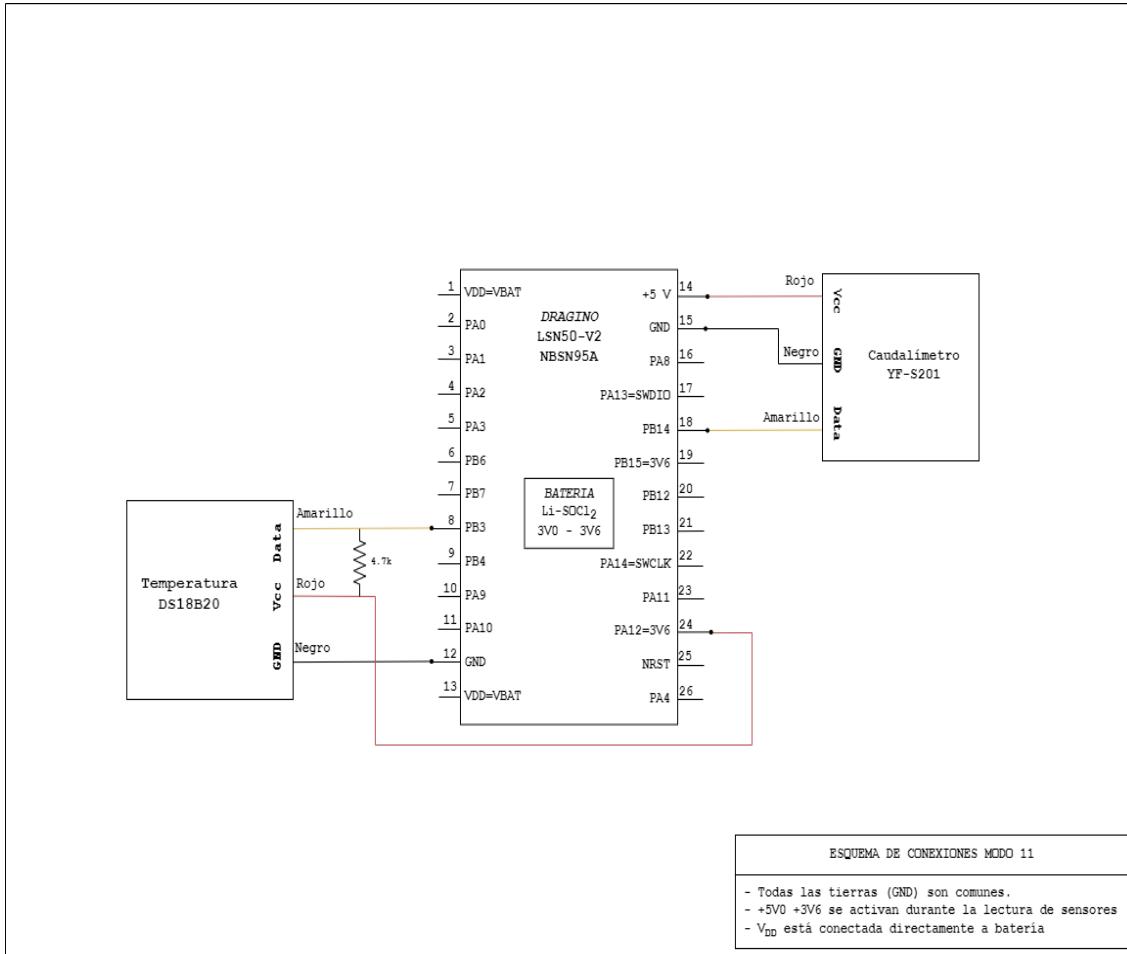
Esquema electrónico de las conexiones del primer tipo de nodo implementado

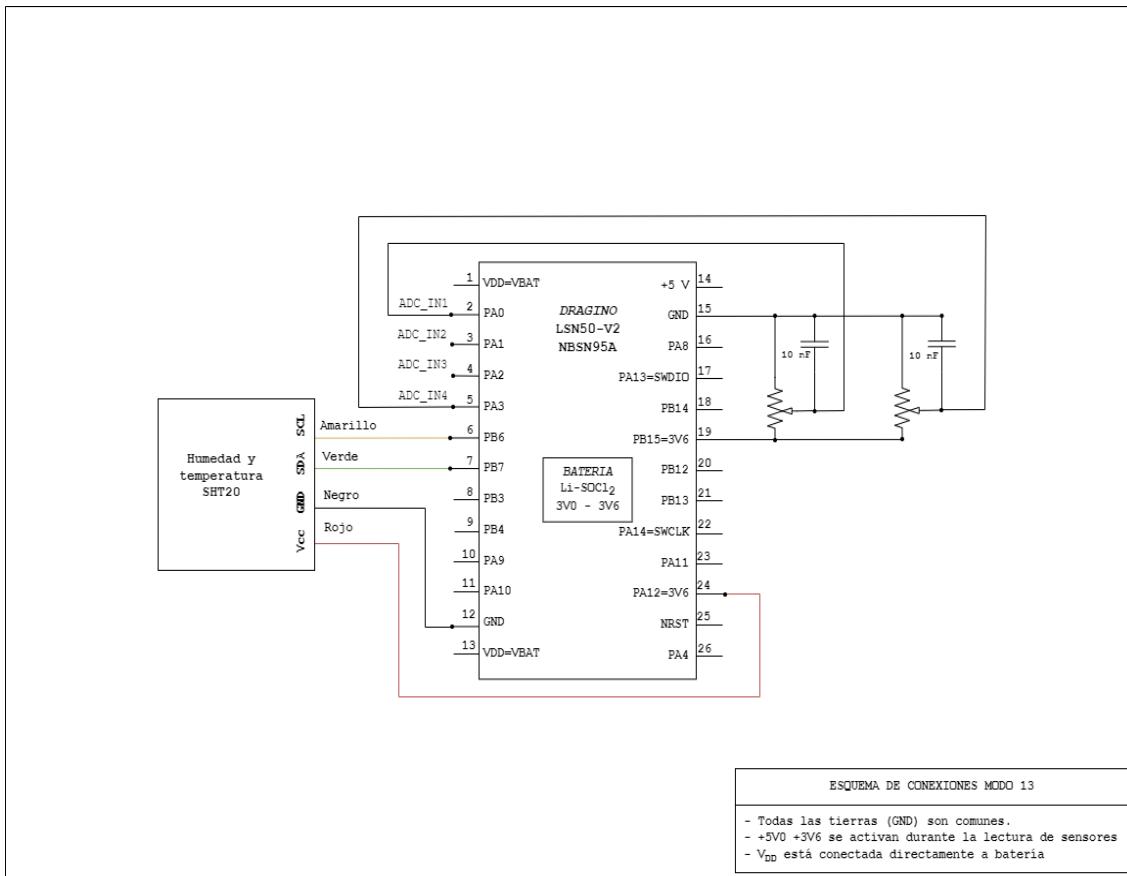


Esquema electrónico de las conexiones del segundo tipo de nodo implementado









Apéndice C

Código utilizado para despejar los rangos de cobertura del primer y segundo nodo implementado.

Cálculo rango de cobertura primer nodo

```
%% Balance de enlace cobertura real
clear all
close all
clc
% Datos a utilizar
syms R
lambda = (3*10^8) / (433*10^6);
Pt = 24,87;
Pr = -125;
Gt = 3;
Gr = 3;
Pers11t = 0.96;
Pers11r = 0.96;
Lpol = 0.9;
Lrg = 1.2*R; % db/km
Lrf = 0.035*R; % db/km
Lzb = 0.09*R; % db/km
% Cálculo del balance de enlace para cobertura
dist = double(solve(Pr-Gt-Gr-20*log10(lambda/(4*pi*R*10^3))...
    -10*log10(Pers11t)-10*log10(Pers11r)-10*log10(Lpol)+Lrg+Lrf+Lzb-
    Pt));
disp(['Radio cobertura = ',num2str(dist), ' km'])
```

Cálculo rango de cobertura segundo nodo

```
%% Balance de enlace cobertura real
clear all
close all
clc
% Datos a utilizar
syms R
lambda = (3*10^8) / (868*10^6);
Pt = 20;
Pr = -148;
Gt = -2.8;
Gr = -8;
Pers11t = 0.93;
Pers11r = 0.93;
Lpol = 0.9;
Lrg = 0.9*R; % db/km
Lrf = 0.035*R; % db/km
Lzb = 0.18*R; % db/km
% Cálculo del balance de enlace para cobertura
dist = double(solve(Pr-Gt-Gr-20*log10(lambda/(4*pi*R*10^3))...
    -10*log10(Pers11t)-10*log10(Pers11r)-10*log10(Lpol)+Lrg+Lrf+Lzb-
    Pt));
disp(['Radio cobertura = ',num2str(dist), ' km'])
```

Apéndice D

Para comprobar que el módulo *Ra-02* de LoRa funcionaba correctamente y para desarrollar el posterior código para la comunicación entre el nodo sensor y central, se hizo uso de la librería, de Yandiev Ruslan, *sx1278-LoRa-RaspberryPi*.

A continuación, se muestran los códigos *ping.c* y *pong.c* usados para la prueba.

ping.c

```
#include "LoRa.h"

void tx_f(txData *tx){
    LoRa_ctl *modem = (LoRa_ctl *)(tx->userPtr);
    printf("tx done;\t");
    printf("sent string: \"%s\"\n\n", tx->buf); //Data we've sent

    LoRa_receive(modem);
}

void * rx_f(void *p){
    rxData *rx = (rxData *)p;
    LoRa_ctl *modem = (LoRa_ctl *)(rx->userPtr);
    LoRa_stop_receive(modem); //manually stoping RxCont mode
    printf("rx done;\t");
    printf("CRC error: %d;\t", rx->CRC);
    printf("Data size: %d;\t", rx->size);
    printf("received string: \"%s\";\t", rx->buf); //Data we've received
    printf("RSSI: %d;\t", rx->RSSI);
    printf("SNR: %f\n", rx->SNR);

    LoRa_sleep(modem);
    free(p);

    return NULL;
}

int main(){

    char txbuf[255];
    LoRa_ctl modem;

    //See for typedefs, enumerations and there values in LoRa.h header
    //file
    modem.spiCS = 0; //Raspberry SPI CE pin number
    modem.tx.callback = tx_f;
    modem.tx.data.buf = txbuf;
    modem.rx.callback = rx_f;
}
```

```

modem.rx.data.userPtr = (void *)(&modem); //To handle with chip from
rx callback
modem.tx.data.userPtr = (void *)(&modem); //To handle with chip from
tx callback
memcpy(modem.tx.data.buf, "Ping", 5); //copy data we'll sent to buffer
modem.tx.data.size = 5; //Payload len
modem.eth.preambleLen=6;
modem.eth.bw = BW62_5; //Bandwidth 62.5KHz
modem.eth.sf = SF12; //Spreading Factor 12
modem.eth.ecr = CR8; //Error coding rate CR4/8
modem.eth.CRC = 1; //Turn on CRC checking
modem.eth.freq = 434800000; // 434.8MHz
modem.eth.resetGpioN = 4; //GPIO4 on lora RESET pin
modem.eth.dio0GpioN = 17; //GPIO17 on lora DIO0 pin to control Rxdone
and Txdone interrupts
modem.eth.outPower = OP20; //Output power
modem.eth.powerOutPin = PA_BOOST; //Power Amplifire pin
modem.eth.AGC = 1; //Auto Gain Control
modem.eth.OCP = 240; // 45 to 240 mA. 0 to turn off protection
modem.eth.implicitHeader = 0; //Explicit header mode
modem.eth.syncWord = 0x12;
//For detail information about SF, Error Coding Rate, Explicit
header, Bandwidth, AGC, Over current protection and other features refer
to sx127x datasheet https://www.semtech.com/uploads/documents/DS\_SX1276-7-8-9\_W\_APP\_V5.pdf

LoRa_begin(&modem);
LoRa_send(&modem);

printf("Time on air data - Tsym: %f;\t", modem.tx.data.Tsym);
printf("Tpkt: %f;\t", modem.tx.data.Tpkt);
printf("payloadSymbNb: %u\n", modem.tx.data.payloadSymbNb);

while(LoRa_get_op_mode(&modem) != SLEEP_MODE){
    sleep(1);
}

printf("end\n");

LoRa_end(&modem);
}

```

pong.c

```

#include "LoRa.h"

void tx_f(txData *tx){
    LoRa_ctl *modem = (LoRa_ctl *)(tx->userPtr);
    printf("tx done;\t");
    printf("sent string: \"%s\"\n", tx->buf); //Data we've sent

    LoRa_sleep(modem);
}

void * rx_f(void *p){
    rxData *rx = (rxData *)p;
    LoRa_ctl *modem = (LoRa_ctl *)(rx->userPtr);
    LoRa_stop_receive(modem); //manually stoping RxCont mode
    printf("rx done;\t");
    printf("CRC error: %d;\t", rx->CRC);
    printf("Data size: %d;\t", rx->size);
    printf("received string: \"%s\";\t", rx->buf); //Data we've received
    printf("RSSI: %d;\t", rx->RSSI);
    printf("SNR: %f\n\n", rx->SNR);

    memcpy(modem->tx.data.buf, "Kong", 5); //copy data we'll sent to
buffer
    modem->tx.data.size = 5; //Payload len

    LoRa_send(modem);
    printf("Time on air data - Tsym: %f;\t", modem->tx.data.Tsym);
    printf("Tpkt: %f;\t", modem->tx.data.Tpkt);
    printf("payloadSymbNb: %u\n", modem->tx.data.payloadSymbNb);
    free(p);

    return NULL;
}

int main(){

    char txbuf[255];
    char rxbuf[255];
    LoRa_ctl modem;

    //See for typedefs, enumerations and there values in LoRa.h header
file
    modem.spiCS = 0; //Raspberry SPI CE pin number
    modem.tx.callback = tx_f;
    modem.tx.data.buf = txbuf;
    modem.rx.callback = rx_f;
}

```

```

modem.rx.data.userPtr = (void *)(&modem); //To handle with chip from
rx callback
modem.tx.data.userPtr = (void *)(&modem); //To handle with chip from
tx callback
modem.eth.preambleLen=6;
modem.eth.bw = BW62_5; //Bandwidth 62.5KHz
modem.eth.sf = SF12; //Spreading Factor 12
modem.eth.ecr = CR8; //Error coding rate CR4/8
modem.eth.CRC = 1; //Turn on CRC checking
modem.eth.freq = 434800000; // 434.8MHz
modem.eth.resetGpioN = 4; //GPIO4 on lora RESET pin
modem.eth.dio0GpioN = 17; //GPIO17 on lora DI00 pin to control Rxdone
and Txdone interrupts
modem.eth.outPower = OP20; //Output power
modem.eth.powerOutPin = PA_BOOST; //Power Amplifire pin
modem.eth.AGC = 1; //Auto Gain Control
modem.eth.OCP = 240; // 45 to 240 mA. 0 to turn off protection
modem.eth.implicitHeader = 0; //Explicit header mode
modem.eth.syncWord = 0x12;
//For detail information about SF, Error Coding Rate, Explicit
header, Bandwidth, AGC, Over current protection and other features refer
to sx127x datasheet https://www.semtech.com/uploads/documents/DS\_SX1276-7-8-9\_W\_APP\_V5.pdf

LoRa_begin(&modem);
LoRa_receive(&modem);

while(LoRa_get_op_mode(&modem) != SLEEP_MODE){
    sleep(1);
}

printf("end\n");

LoRa_end(&modem);
}

```

Apéndice E

Al igual que en el [apéndice D](#), se hizo uso del repositorio, de Yandiev Ruslan, *sx1278-LoRa-RaspberryPi* para generar el código que se va a utilizar para transmitir la información medida por el primer tipo de nodo sensor al nodo central.

transmit_implicit.c

```
#include "LoRa.h"
#define SIZE_DATA 200 // tamaño del fichero con el dato (dejar margen)

int obtenerDatos (char *NombreFichero, char *Dato){
    FILE *fich;
    int lectura=0;
    if ((fich = fopen(NombreFichero, "r")) == NULL) { /* control del
error de apertura */
        printf (" Error en la apertura. Es posible que el fichero no
exista \n ");
    }
    else{
        //while (!feof(fich)){
            //lectura+=fscanf(fich,"%c",Dato[lectura]);
            lectura+=fread(Dato,sizeof(char),SIZE_DATA,fich);
        //}
        //fprintf(stderr, "%s\n datos leidos=%d\n", Dato, lectura);

        fclose (fich);
    }
    return 0;
}

void tx_f(txData *tx){
    printf("tx done \n");
}

int main(){

    char txbuf[255];
    LoRa_ctl modem;
    char *NombreFichero="tx_data.json";
    char tx_data[SIZE_DATA];

    while(1){

        //Transmision data
        obtenerDatos(NombreFichero,tx_data);
    }
}
```

```

fprintf(stderr, "Datos a transmitir por el nodo sensor: %s\n",
tx_data);

//See for typedefs, enumerations and there values in LoRa.h header
file
modem.spiCS = 0;//Raspberry SPI CE pin number
modem.tx.callback = tx_f;
modem.tx.data.buf = txbuf;
memcpy(modem.tx.data.buf, tx_data, SIZE_DATA);//copy data we'll sent
to buffer
modem.eth.payloadLen = 5;//Payload len in implicit header mode
modem.eth.preambleLen=6;
modem.eth.bw = BW62_5;//Bandwidth 62.5KHz
modem.eth.sf = SF12;//Spreading Factor 12
modem.eth.ecr = CR8;//Error coding rate CR4/8
modem.eth.CRC = 1;//Turn on CRC checking
modem.eth.freq = 434800000;// 434.8MHz
modem.eth.resetGpioN = 4;//GPIO4 on lora RESET pi
modem.eth.dio0GpioN = 17;//GPIO17 on lora DI00 pin to control Rxdone
and Txdone interrupts
modem.eth.outPower = OP20;//Output power
modem.eth.powerOutPin = PA_BOOST;//Power Amplifire pin
modem.eth.AGC = 1;//Auto Gain Control
modem.eth.OCP = 240;// 45 to 240 mA. 0 to turn off protection
modem.eth.implicitHeader = 1;//implicit header mode
modem.eth.syncWord = 0x12;
//For detail information about SF, Error Coding Rate, Explicit
header, Bandwidth, AGC, Over current protection and other features refer
to sx127x datasheet https://www.semtech.com/uploads/documents/DS_SX1276-
7-8-9_W_APP_V5.pdf

LoRa_begin(&modem);
LoRa_send(&modem);

printf("Tsym: %f\n", modem.tx.data.Tsym);
printf("Tpkt: %f\n", modem.tx.data.Tpkt);
printf("payloadSymbNb: %u\n", modem.tx.data.payloadSymbNb);

printf("sleep %d seconds to transmitt complete\n",
(int)modem.tx.data.Tpkt/1000);
sleep(10);

}

printf("end\n");

LoRa_end(&modem);
}

```

receive_implicit.c

```
#include "LoRa.h"
#define SIZE_DATA 200 // tamaño del fichero con el dato (dejar margen)

int introducirDato (char *NombreFichero, char *Dato){
    FILE *fich;
    int escritura=0;
    int i=0;
    if ((fich = fopen(NombreFichero, "a")) == NULL) { /* control del
error de apertura */
        printf ( " Error en la apertura. Es posible que el fichero no
exista \n ");
    }
    else{
        while (Dato[i]!=0) {
            escritura+=fwrite(&Dato[i],sizeof(char),1,fich);
            i=i+1;
        }
        fwrite("\n",sizeof(char),1,fich);

        //fprintf(stderr, "%s\n datos escritos=%d\n", Dato, escritura);

        fclose (fich);
    }
    return 0;
}

void * rx_f(void *p){
    rxData *rx = (rxData *)p;

    char *NombreFichero="rx_data.json";

    //Reception data
    introducirDato(NombreFichero,rx->buf);

    printf("rx done \n");
    printf("CRC error: %d\n", rx->CRC);
    printf("Datos recibidos por el nodo central: %s\n", rx->buf);
    //printf("string: %s\n", rx->buf);//Data we've received
    printf("RSSI: %d\n", rx->RSSI);
    printf("SNR: %f\n", rx->SNR);
    free(p);
    return NULL;
}

int main(){

    while (1){
```

```

LoRa_ctl modem;
char rx_data[SIZE_DATA];

//See for typedefs, enumerations and there values in LoRa.h
header file
modem.spiCS = 0;//Raspberry SPI CE pin number
modem.rx.callback = rx_f;
modem.eth.payloadLen = SIZE_DATA;//payload len used in implicit
header mode
modem.eth.preambleLen=6;
modem.eth.bw = BW62_5;//Bandwidth 62.5KHz
modem.eth.sf = SF12;//Spreading Factor 12
modem.eth.ecr = CR8;//Error coding rate CR4/8
modem.eth.CRC = 1;//Turn on CRC checking
modem.eth.freq = 434800000;// 434.8MHz
modem.eth.resetGpioN = 4;//GPIO4 on lora RESET pin
modem.eth.dio0GpioN = 17;//GPIO17 on lora DIO0 pin to control
Rxdone and Txdone interrupts
modem.eth.outPower = OP20;//Output power
modem.eth.powerOutPin = PA_BOOST;//Power Amplifire pin
modem.eth.AGC = 1;//Auto Gain Control
modem.eth.OCP = 240;// 45 to 240 mA. 0 to turn off protection
modem.eth.implicitHeader = 1;//Implicit header mode
modem.eth.syncWord = 0x12;
//For detail information about SF, Error Coding Rate, Explicit
header, Bandwidth, AGC, Over current protection and other features refer
to sx127x datasheet https://www.semtech.com/uploads/documents/DS\_SX1276-7-8-9\_W\_APP\_V5.pdf

LoRa_begin(&modem);
LoRa_receive(&modem);

sleep(10);
printf("Recieved data\n");

LoRa_end(&modem);
}

}

```

Apéndice F

Para programar el segundo nodo sensor, se utilizó la librería del fabricante NBSN95 (46). La integración es compleja, pero se muestran los dos códigos más significativos para entender el procedimiento del sistema.

El código `sensores.c` es el código realizado para la adquisición y almacenamiento temporal de los datos provenientes de los sensores. En cambio, el código `main.c` indica la elección de los sensores en función del modo y se encarga de establecer un orden a la hora de encender los sensores, tomar los datos, almacenarlos y transmitirlos.

`sensores.c`

```
#include "sensores.h"
#include "uart.h"
#include "main.h"
#include "tim.h"
#include "common.h"

#define TIME_WINDOW_SECONDS 30
#define SENSECAP_NUMBER 3
#define COUNTERS_NUMBER 3
#define SHT2X_NUMBER 4

#define LAT_LON_STR_SZ 8

#define TX_RS485_Pin GPIO_PIN_10
#define TX_RS485_GPIO_Port GPIOA

#define GPS_PWR1_Pin GPIO_PIN_12
#define GPS_PWR1_GPIO_Port GPIOA

#define GPS_PWR2_Pin GPIO_PIN_15
#define GPS_PWR2_GPIO_Port GPIOB

#define ONEW_PWR_Pin GPIO_PIN_12
#define ONEW_PWR_GPIO_Port GPIOA

#define SHT2x_I2C_ADDR 0x40
#define SHT2x_HOLD_MASTER 1
#define SHT2x_READ_TEMP_HOLD 0xe3
#define SHT2x_READ_RH_HOLD 0xe5
#define SHT2x_READ_TEMP_NOHOLD 0xf3
#define SHT2x_READ_RH_NOHOLD 0xf5
#define SHT2x_WRITE_REG 0xe6
#define SHT2x_READ_REG 0xe7
#define SHT2x_SOFT_RESET 0xfe
#define SHT2x_TIMEOUT 1000
```

```

typedef enum SHT2x_Resolution {
    RES_14_12 = 0x00,
    RES_12_8 = 0x01,
    RES_13_10 = 0x80,
    RES_11_11 = 0x81,
} SHT2x_Resolution;

typedef struct sens_gps_s{
    int32_t lat;
    int32_t lon;
}sens_gps_s;

typedef struct sens_meteo_{
    uint16_t wd;          // --> dirección de viento
    int16_t t;            // --> temperatura ambiente
    uint8_t hum;          // --> humedad ambiente
    uint16_t ws;          // --> velocidad del viento
    uint8_t gs;           // --> velocidad de ráfaga
    uint16_t rf;          // --> lluvia
    uint16_t uvi;         // --> índice ultravioleta
    uint32_t l:24;        // --> cantidad de luz
    uint32_t p:24;        // --> presión
    uint8_t errors;
}sens_meteo_s;

typedef struct sens_sensecap_{
    uint16_t t;
    uint16_t vwc;
    uint16_t ec;
    uint16_t sal;
    uint16_t tds;
    uint16_t eps;
}sens_sensecap_s;

typedef struct sens_sht20_{
    int16_t t;
    uint16_t hum;
}sens_sht20_s;

typedef struct power_pin{
    GPIO_TypeDef *GPIOx;
    uint16_t GPIO_Pin;
}power_pin_s;

extern uint8_t uart2_user_recieve_flag;
extern uint8_t uart4_user_recieve_flag;
extern uint8_t rxlen_u1;
extern uint8_t rxDATA_u1[];
extern uint8_t rdbuf_u1;
extern uint8_t rxDATA_u4[];

```

```

extern uint8_t rxbuf_u4;
extern uint8_t rxlen_u4;

const power_pin_s
SHT2X_power_pins[SHT2X_NUMBER]={{GPIOB,GPIO_PIN_12},{GPIOA,GPIO_PIN_11},{GPIOB,GPIO_PIN_12},{GPIOB,GPIO_PIN_3}};

volatile int flag30s = 0;
int sensecap_retry = 0;
int current_sensecap = 0;

uint8_t sensors_flags=0;
#define SENSOR_FLAG_GPS      0x01
#define SENSOR_FLAG_METEO    0x02
#define SENSOR_FLAG_SENSECAP 0x04

sens_gps_s dat_gps;
sens_meteo_s dat_meteo;
sens_sensecap_s dat_sensecap[SENSECAP_NUMBER];
sens_count_s dat_counters[COUNTERS_NUMBER];
sens_sht20_s dat_sht20[SHT2X_NUMBER];

uint8_t vunkers_payload[60];

void Vunkers_Sensors_Timer_Flag(void){
    flag30s++;// = 1;
}

uint32_t GetIntegerValue(char* str_value)
{
    char* endch = strchr(str_value,'.');
    if(endch!=NULL)
        *endch = 0;
    else{
        endch = strchr(str_value,',');
        if(endch!=NULL)
            *endch = 0;
    }
    return atoi(str_value);
}

uint8_t GPS_CheckFrame()
{
    uint8_t toRet = 0;
    if(uart2_user_recieve_flag){
        rxDATA_u1[rxlen_u1]=0;
        user_main_printf("GPS Trama recibida:%s",rxDATA_u1);
        const char* curr_str = (const char*)rxDATA_u1;

        //$/GNRMC,203544.000,A,0438.9198,N,07404.3962,W,0.00,247.85,150517,,,A*62
    }
}

```

```

if(strncmp("$GNRMC,", curr_str,7) == 0)
{
    user_main_printf("Prefijo encontrado $GNRMC,");
    curr_str += 7; // Primer valor
    char* lat_str;
    char* lon_str;
    uint8_t is_west = 0;
    uint8_t is_south = 0;
    toRet=1;
    for(int i=0; i<5; i++){
        curr_str = strchr(curr_str,',');
        if(curr_str == NULL){
            toRet = 0;
            break;
        }
        curr_str++;
        if(i==1)lat_str=(char*)curr_str;
        if(i==2 && *(curr_str+1)=='S') is_south=1;
        if(i==3)lon_str=(char*)curr_str;
        if(i==4 && *(curr_str+1)=='W') is_west=1;
    }
    if((lat_str[0]==',' || (lon_str[0] == ',')))
    {
        toRet=0;
    }
    if(toRet)
    {
        user_main_printf("LAT str: %s LON str:
%s",lat_str,lon_str);
        dat_gps.lat = GetIntegerValue(lat_str);
        if(is_south) dat_gps.lat = -dat_gps.lat;
        dat_gps.lon = GetIntegerValue(lon_str);
        if(is_west) dat_gps.lon = -dat_gps.lon;
        user_main_printf("LAT %d LON
%d",dat_gps.lat,dat_gps.lon);
    }
}
rxlen_u1 =0;
uart2_user_recieve_flag = 0;
}
return toRet;
}

uint16_t calc_crc16(uint8_t *snd, int num) {
    int i, j;
    uint16_t c, crc = 0xFFFF;
    for (i = 0; i < num; i++) {
        c = snd[i] & 0x00FF;
        crc ^= c;

```

```

        for (j = 0; j < 8; j++) {
            if (crc & 0x0001) {
                crc >>= 1;
                crc ^= 0xA001;
            }
            else  crc >>= 1;
        }
    return crc;
}

char update_crc_meteo(char extract, char crc)
{
    char sum;
    for (char tempI = 8; tempI; tempI--)
    {
        sum = (crc ^ extract) & 0x01;
        crc >>= 1;
        if (sum)
            crc ^= 0x8C;
        extract >>= 1;
    }
    return crc;
}

uint8_t METEO_CheckFrame()
{
    uint8_t toRet = 0;

    if(uart4_user_recieve_flag == 2)
    {
        user_main_printf("METEO Trama
recibida:24%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x",
rxDATA_u4[0], rxDATA_u4[1], rxDATA_u4[2], rxDATA_u4[3], rxDATA_u4[4], rxDATA_u4[5], rxDATA_u4[6], rxDATA_u4[7], rxDATA_u4[8], rxDATA_u4[9], rxDATA_u4[10], rxDATA_u4[11], rxDATA_u4[12], rxDATA_u4[13], rxDATA_u4[14], rxDATA_u4[15]/*, rxDATA_u4[16], rxDATA_u4[17]*/);

        dat_meteo.wd = rxDATA_u4[1] |
(((uint16_t)(rxDATA_u4[2]&0x80))<<1);
        user_main_printf("METEO WD= %04x %d", dat_meteo.wd,
dat_meteo.wd);

        dat_meteo.t = (((uint16_t)(rxDATA_u4[2]&0x07))<<8) |
rxDATA_u4[3];
        user_main_printf("METEO T= %04x %d", dat_meteo.t, dat_meteo.t);
    }
}

```

```

dat_meteo.hum = rxDATA_u4[4];
user_main_printf("METEO HUM= %02x %d", dat_meteo.hum,
dat_meteo.hum);

dat_meteo.ws = (((uint16_t)(rxDATA_u4[2]&0x10))<<4) |
rxDATA_u4[5];
user_main_printf("METEO WS= %04x %d", dat_meteo.ws,
dat_meteo.ws);

dat_meteo.gs = rxDATA_u4[6];
user_main_printf("METEO GS= %02x %d", dat_meteo.gs,
dat_meteo.gs);

dat_meteo.rf = (((uint16_t)rxDATA_u4[7])<<8) | rxDATA_u4[8];
user_main_printf("METEO RF= %04x %d", dat_meteo.rf,
dat_meteo.rf);

dat_meteo.uvi = (((uint16_t)rxDATA_u4[9])<<8) | rxDATA_u4[10];
user_main_printf("METEO UVI= %04x %d", dat_meteo.uvi,
dat_meteo.uvi);

dat_meteo.l = (((uint32_t)rxDATA_u4[11])<<16) |
((uint16_t)rxDATA_u4[12])<<8) | rxDATA_u4[13];
user_main_printf("METEO L= %06x %d", dat_meteo.l, dat_meteo.l);

uint8_t crc = 0;
uint8_t crc_full = 0;
uint8_t checksum = 0x24;
for(int i=0; i<15;i++)
{
    checksum += rxDATA_u4[i];
}

crc_full = update_crc_meteo(0x24,crc_full);
crc_full = update_crc_meteo(rxDATA_u4[0],crc_full);

for(int i=1; i<14;i++)
{
    crc_full = update_crc_meteo(rxDATA_u4[i],crc_full);
    crc = update_crc_meteo(rxDATA_u4[i],crc);
}

user_main_printf("METEO CRC= %02x, Calculated CRC only payload:
%04x, full %04x", rxDATA_u4[14], crc, crc_full);

user_main_printf("METEO CHCK= %02x, calculated %02x", rxDATA_u4[15],
checksum);

//dat_meteo.p = (((uint32_t)rxDATA_u4[15])<<16) |
((uint16_t)rxDATA_u4[16])<<8) | rxDATA_u4[17];

```

```

//user_main_printf("METEO P= %06x %d", dat_meteo.p, dat_meteo.p);

dat_meteo.errors = ((rxDATA_u4[2]&0x08)>>2);
if(dat_meteo.wd == 0x1FF) dat_meteo.errors |= 0x01;
if(dat_meteo.t == 0xFF) dat_meteo.errors |= 0x04;
if(dat_meteo.hum == 0xFF) dat_meteo.errors |= 0x08;
if(dat_meteo.gs == 0xFF) dat_meteo.errors |= 0x10;
if(dat_meteo.uvi == 0xFFFF) dat_meteo.errors |= 0x20;
if(dat_meteo.l == 0xFFFF) dat_meteo.errors |= 0x40;
user_main_printf("METEO ERR= %02x %d", dat_meteo.errors,
dat_meteo.errors);

toRet=1;
uart4_user_recieve_flag = 0;
rxlen_u4=0;
}

return toRet;
}

uint8_t SENSECAP_CheckFrame(uint8_t num_sens)
{
    uint8_t toRet = 0;

    if(uart4_user_recieve_flag == 2)
    {
        toRet = 1;

        user_main_printf("SENSECAP Trama
recibida:%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x%02x
%02x",rxDATA_u4[0], 

rxDATA_u4[1],rxDATA_u4[2],rxDATA_u4[3],rxDATA_u4[4],rxDATA_u4[5],rxDATA_u
4[6],rxDATA_u4[7],rxDATA_u4[8],rxDATA_u4[9],rxDATA_u4[10], 

rxDATA_u4[11],rxDATA_u4[12],rxDATA_u4[13],rxDATA_u4[14],rxDATA_u4[15],rxD
ATA_u4[16]/*,rxDATA_u4[17]*/);

        if (rxDATA_u4[0] != num_sens && rxDATA_u4[1] != 0x04) { // Se
comprueba que el ID y el modo es igual al enviado en la instrucción
            user_main_printf("Error en cabecera de trama Sensecap");
            toRet=0;
        }
        /*else*/
        {
            uint16_t crc = calc_crc16(rxDATA_u4, SENSECAP_PAYLOAD_LENGTH-
2);
            uint16_t crc_frame = rxDATA_u4[SENSECAP_PAYLOAD_LENGTH - 1] +
(((uint16_t)rxDATA_u4[SENSECAP_PAYLOAD_LENGTH - 2])<<8);
        }
    }
}

```

```

        user_main_printf("SENSECAP CRC= %04x, Calculated CRC: %04x",
crc_frame, crc);
    }
    user_main_printf("SENSECAP Data length= %02x", rxDATA_u4[2]);

    if(toRet)
    {
        dat_sensecap[num_sens].t =
(((uint16_t)rxDATA_u4[3])<<8) | rxDATA_u4[4];
        user_main_printf("SENSECAP %d t= %04x %d", num_sens,
dat_sensecap[num_sens].t, dat_sensecap[num_sens].t);
        dat_sensecap[num_sens].vwc =
(((uint16_t)rxDATA_u4[5])<<8) | rxDATA_u4[6];
        user_main_printf("SENSECAP %d vwc= %04x %d", num_sens,
dat_sensecap[num_sens].vwc, dat_sensecap[num_sens].vwc);
        dat_sensecap[num_sens].ec =
(((uint16_t)rxDATA_u4[7])<<8) | rxDATA_u4[8];
        user_main_printf("SENSECAP %d ec= %04x %d", num_sens,
dat_sensecap[num_sens].ec, dat_sensecap[num_sens].ec);
        dat_sensecap[num_sens].sal =
(((uint16_t)rxDATA_u4[9])<<8) | rxDATA_u4[10];
        user_main_printf("SENSECAP %d sal= %04x %d", num_sens,
dat_sensecap[num_sens].sal, dat_sensecap[num_sens].sal);
        dat_sensecap[num_sens].tds = (((uint16_t)rxDATA_u4[11])<<8)
| rxDATA_u4[12];
        user_main_printf("SENSECAP %d tds= %04x %d", num_sens,
dat_sensecap[num_sens].tds, dat_sensecap[num_sens].tds);
        dat_sensecap[num_sens].eps =
(((uint16_t)rxDATA_u4[13])<<8) | rxDATA_u4[14];
        user_main_printf("SENSECAP %d eps= %04x %d", num_sens,
dat_sensecap[num_sens].eps, dat_sensecap[num_sens].eps);
    }

    rxlen_u4=0;
    uart4_user_recieve_flag = 0;
}

return toRet;
}

void SENSECAP_SendCommand(uint8_t num_sens) {

    // comando de lectura
    uint8_t cmd[] = {0x00, 0x04, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00};
    cmd[0]= num_sens;
    // CRC
    uint16_t crc = calc_crc16(cmd, 6);
    cmd[6] = (char)(crc & 0xFF);
    cmd[7] = (char)((crc >> 8) & 0xFF);
    // Envío
}

```

```

        uart4_user_recieve_flag=1;
        rxlen_u4=0;
        HAL_GPIO_WritePin(TX_RS485_GPIO_Port, TX_RS485_Pin,
GPIO_PIN_SET);      // Pull DE high to enable TX operation
        HAL_UART_Transmit(&huart4, (uint8_t *)cmd, sizeof(cmd) , 1000);
        HAL_GPIO_WritePin(TX_RS485_GPIO_Port, TX_RS485_Pin,
GPIO_PIN_RESET);    // Pull RE Low to enable RX operation
    }

void inline Vunkers_counter_irq(int counter_index)
{
    dat_counters[counter_index].count++;
    uint32_t millis = HAL_GetTick();
    if(dat_counters[counter_index].prev_millis>=0)
    {
        uint32_t period = millis -
dat_counters[counter_index].prev_millis;
        if(period<dat_counters[counter_index].period_min)
            dat_counters[counter_index].period_min=period;
        if(period>dat_counters[counter_index].period_max)
            dat_counters[counter_index].period_max=period;
    }
    dat_counters[counter_index].prev_millis=millis;
}

int append_counter_payload(int counter_index, int payload_index)
{
    user_main_printf("CONTADOR %d count=%d max=%d min=%d",
counter_index,
dat_counters[counter_index].count,
dat_counters[counter_index].period_max,
dat_counters[counter_index].period_min);
    // 12 bytes cada contador, cuidado con el tamaño de vunkers_payload
!!!!
    vunkers_payload[payload_index++]=dat_counters[counter_index].count>>24;
    vunkers_payload[payload_index++]=dat_counters[counter_index].count>>16;
    vunkers_payload[payload_index++]=dat_counters[counter_index].count>>8;
    vunkers_payload[payload_index++]=dat_counters[counter_index].count;
    /*vunkers_payload[payload_index++]=TIME_WINDOW_SECONDS>>24;
    vunkers_payload[payload_index++]=TIME_WINDOW_SECONDS>>16;
    vunkers_payload[payload_index++]=TIME_WINDOW_SECONDS>>8;
    vunkers_payload[payload_index++]=TIME_WINDOW_SECONDS*/
    vunkers_payload[payload_index++]=dat_counters[counter_index].period_max>>
24;
    vunkers_payload[payload_index++]=dat_counters[counter_index].period_max>>
16;
}

```

```

vunkers_payload[payload_index++]=dat_counters[counter_index].period_max>>
8;

vunkers_payload[payload_index++]=dat_counters[counter_index].period_max;

vunkers_payload[payload_index++]=dat_counters[counter_index].period_min>>
24;

vunkers_payload[payload_index++]=dat_counters[counter_index].period_min>>
16;

vunkers_payload[payload_index++]=dat_counters[counter_index].period_min>>
8;

vunkers_payload[payload_index++]=dat_counters[counter_index].period_min;

    return payload_index;
}

uint16_t SHT2x_GetRaw(uint8_t cmd) {
    uint8_t val[3] = { 0 };
    HAL_I2C_Master_Transmit(&hi2c1, SHT2x_I2C_ADDR << 1, &cmd, 1,
SHT2x_TIMEOUT);
    HAL_I2C_Master_Receive (&hi2c1, SHT2x_I2C_ADDR << 1, val, 3,
SHT2x_TIMEOUT);
    return val[0] << 8 | val[1];
}

uint8_t SHT2x_ReadUserReg(void) {
    uint8_t val;
    uint8_t cmd = SHT2x_READ_REG;
    HAL_I2C_Master_Transmit(&hi2c1, SHT2x_I2C_ADDR << 1, &cmd, 1,
SHT2x_TIMEOUT);
    HAL_I2C_Master_Receive (&hi2c1, SHT2x_I2C_ADDR << 1, &val, 1,
SHT2x_TIMEOUT);
    return val;
}

void SHT2x_SetResolution(SHT2x_Resolution res ) {
    uint8_t val = SHT2x_ReadUserReg();
    val = (val & 0x7e) | res;
    uint8_t temp[2] = { SHT2x_WRITE_REG, val };
    HAL_I2C_Master_Transmit(&hi2c1, SHT2x_I2C_ADDR << 1, temp, 2,
SHT2x_TIMEOUT);
}

void SHT20_read(int i){
    HAL_GPIO_WritePin(SHT2X_power_pins[i].GPIOx,
SHT2X_power_pins[i].GPIO_Pin , GPIO_PIN_SET);
}

```

```

SHT2x_SetResolution(RES_14_12);
//char cmd_temp = (hold ? SHT2x_READ_TEMP_HOLD:
SHT2x_READ_TEMP_NOHOLD);
dat_sht20[i].t = SHT2x_GetRaw(SHT2x_READ_TEMP_HOLD);
// uint8_t cmd_hum = (hold ? SHT2x_READ_RH_HOLD :
SHT2x_READ_RH_NOHOLD);
dat_sht20[i].hum = SHT2x_GetRaw(SHT2x_READ_RH_HOLD);
HAL_GPIO_WritePin(SHT2X_power_pins[i].GPIOx,
SHT2X_power_pins[i].GPIO_Pin , GPIO_PIN_RESET);
}

void SHT20_init_pwr_pin(int i){
GPIO_InitTypeDef GPIO_InitStruct={0};
GPIO_InitStruct.Pin = SHT2X_power_pins[i].GPIO_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
HAL_GPIO_Init(SHT2X_power_pins[i].GPIOx, &GPIO_InitStruct);
HAL_GPIO_WritePin(SHT2X_power_pins[i].GPIOx,
SHT2X_power_pins[i].GPIO_Pin , GPIO_PIN_RESET);
}

int append_sht2x_payload(int sht20_index, int payload_index)
{
    user_main_printf("SHT2X %d t=%d h=%d", sht20_index,
dat_sht20[sht20_index].t, dat_sht20[sht20_index].hum);

    // 12 bytes cada contador, cuidado con el tamaño de vunkers_payload
!!!!
vunkers_payload[payload_index++]=dat_sht20[sht20_index].t>>8;
vunkers_payload[payload_index++]=dat_sht20[sht20_index].t;
vunkers_payload[payload_index++]=dat_sht20[sht20_index].hum>>8;
vunkers_payload[payload_index++]=dat_sht20[sht20_index].hum;
return payload_index;
}

uint8_t Vunkers_Sensors_Read()
{
    flag30s = 0;
sensors_flags=0;

    if(sys.mod < model12){
        sensors_flags |= SENSOR_FLAG_GPS;
        uart2_Init();
        HAL_UART_Receive_IT(&huart2,(uint8_t*)&rxbuf_u1,RXSIZE);

        // Inicializar pines para GPS PWR
        GPIO_InitTypeDef GPIO_InitStruct={0};
        GPIO_InitStruct.Pin = GPS_PWR1_Pin;
        GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
}

```

```

GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
HAL_GPIO_Init(GPS_PWR1_GPIO_Port, &GPIO_InitStruct);
HAL_GPIO_WritePin(GPS_PWR1_GPIO_Port, GPS_PWR1_Pin,
GPIO_PIN_SET);

GPIO_InitStruct.Pin = GPS_PWR2_Pin;
HAL_GPIO_Init(GPS_PWR2_GPIO_Port, &GPIO_InitStruct);
HAL_GPIO_WritePin(GPS_PWR2_GPIO_Port, GPS_PWR2_Pin,
GPIO_PIN_SET);

}

if(sys.mod == model7){
    sensors_flags |= SENSOR_FLAG_METEO;
    uart4_Init();
    HAL_UART_Receive_IT(&huart4,(uint8_t*)&rxbuf_u4,RXSIZE);
}

if(sys.mod == model8 || sys.mod == model9)
{
    sensors_flags |= SENSOR_FLAG_SENSECAP;
    sensecap_retry = 0;
    current_sensecap = 0;
    // Inicializar pin para RS_485
    GPIO_InitTypeDef GPIO_InitStruct={0};
    GPIO_InitStruct.Pin = TX_RS485_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
    HAL_GPIO_Init(TX_RS485_GPIO_Port, &GPIO_InitStruct);
    HAL_GPIO_WritePin(TX_RS485_GPIO_Port, TX_RS485_Pin,
GPIO_PIN_RESET);

    memset(dat_sensecap, 0xFF, sizeof(dat_sensecap));

    uart4_Init();
}
if(sys.mod >= model8 && sys.mod <= model12)
{
    dat_counters[0].prev_millis=-1;
    dat_counters[0].count=0;
    dat_counters[0].period_min=0xFFFFFFFF;
    dat_counters[0].period_max=0;
}
if(sys.mod == model12)
{
    for(int i=1;i<COUNTERS_NUMBER;i++){
        dat_counters[i].prev_millis=-1;
        dat_counters[i].count=0;
        dat_counters[i].period_min=0xFFFFFFFF;
        dat_counters[i].period_max=0;
}

```

```

        }
    }
    if(sys.mod == model8 || sys.mod == model10 || sys.mod == model13)
    {
        SHT20_init_pwr_pin(0);
    }
    if(sys.mod == model10)
    {
        for(int j=1; j<4; j++)
        {
            SHT20_init_pwr_pin(j);
        }
    }
    if(sys.mod == model11)
    {
        // Inicializar pines para GPS PWR
        GPIO_InitTypeDef GPIO_InitStruct={0};
        GPIO_InitStruct.Pin = ONEW_PWR_Pin;
        GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
        GPIO_InitStruct.Pull = GPIO_NOPULL;
        GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
        HAL_GPIO_Init(ONEW_PWR_GPIO_Port, &GPIO_InitStruct);
        HAL_GPIO_WritePin(ONEW_PWR_GPIO_Port, ONEW_PWR_Pin,
        GPIO_PIN_SET);
    }
    if(sys.mod == model13){
        /*Configure GPIO pins : PH0 PH1 */
        GPIO_InitTypeDef GPIO_InitStruct={0};
        GPIO_InitStruct.Pin =
        GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3;
        GPIO_InitStruct.Mode = GPIO_MODE_ANALOG;
        GPIO_InitStruct.Pull = GPIO_NOPULL;
        HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
    }
    HAL_TIM_Base_MspInit(&htim2);
    HAL_TIM_Base_Start_IT(&htim2);      //Inicia el timer de 30 segundos

    user_main_printf("SENSORES: Iniciando temporizador de 30
segundos");

    while(flag30s < TIME_WINDOW_SECONDS) {                  //3 - Respetar los
tiempos de inicio de los sensores
        if(sensors_flags & SENSOR_FLAG_GPS){
            if(GPS_CheckFrame())
                sensors_flags &= ~SENSOR_FLAG_GPS;
        }
        if(sensors_flags & SENSOR_FLAG_METEO){
            if(METEO_CheckFrame())
                //sensors_flags &= ~SENSOR_FLAG_METEO;
        }
    }
}

```

```

        }
        if(sensors_flags & SENSOR_FLAG_SENSECAP){
            if(current_sensecap<SENSECAP_NUMBER &&
SENSECAP_CheckFrame(current_sensecap))
                current_sensecap++; //sensors_flags &=
SENSECAP_FLAG_SENSECAP;
        }
        if(current_sensecap<SENSECAP_NUMBER && sensecap_retry <=
flag30s)
    {
        SENSECAP_SendCommand(current_sensecap);
        sensecap_retry=flag30s+3;
    }

    __WFI();
    HAL_IWDG_Refresh(&hiwdg);
    //HAL_Delay(100);
    //
}
HAL_TIM_Base_Stop_IT(&htim2);      //Para el timer de 30 segundos
HAL_TIM_Base_MspDeInit(&htim2);

user_main_printf("SENSORES: Finalizado temporizador de 30
segundos");

int i=0;

if(sys.mod < model12){
    // GPS 4 bytes
    vunkers_payload[i++]=dat_gps.lat>>8;
    vunkers_payload[i++]=dat_gps.lat;
    vunkers_payload[i++]=dat_gps.lon>>8;
    vunkers_payload[i++]=dat_gps.lon;
    HAL_UART_AbortReceive_IT(&huart2);
    uart2_IoDeInit();

    HAL_GPIO_WritePin(GPS_PWR1_GPIO_Port, GPS_PWR1_Pin,
GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPS_PWR2_GPIO_Port, GPS_PWR2_Pin,
GPIO_PIN_RESET);
}
if(sys.mod == model7){
    // METEO 19 bytes
    vunkers_payload[i++]=dat_meteo.wd>>8;
    vunkers_payload[i++]=dat_meteo.wd;
    vunkers_payload[i++]=dat_meteo.t>>8;
    vunkers_payload[i++]=dat_meteo.t;
    vunkers_payload[i++]=dat_meteo.hum;
    vunkers_payload[i++]=dat_meteo.ws>>8;
    vunkers_payload[i++]=dat_meteo.ws;
}

```

```

vunkers_payload[i++]=dat_meteo.gs;
vunkers_payload[i++]=dat_meteo.rf>>8;
vunkers_payload[i++]=dat_meteo.rf;
vunkers_payload[i++]=dat_meteo.uvi>>8;
vunkers_payload[i++]=dat_meteo.uvi;
vunkers_payload[i++]=dat_meteo.l>>16;
vunkers_payload[i++]=dat_meteo.l>>8;
vunkers_payload[i++]=dat_meteo.l;
vunkers_payload[i++]=dat_meteo.p>>16;
vunkers_payload[i++]=dat_meteo.p>>8;
vunkers_payload[i++]=dat_meteo.p;
vunkers_payload[i++]=dat_meteo.errors;

HAL_UART_AbortReceive_IT(&huart4);
uart4_IoDeInit();
}

if(sys.mod == model8 || sys.mod == model9)
{
    for(int j=0; j<SENSECAP_NUMBER; j++)
    {
        vunkers_payload[i++]=dat_sensecap[j].t>>8;
        vunkers_payload[i++]=dat_sensecap[j].t;
        vunkers_payload[i++]=dat_sensecap[j].vwc>>8;
        vunkers_payload[i++]=dat_sensecap[j].vwc;
        vunkers_payload[i++]=dat_sensecap[j].ec>>8;
        vunkers_payload[i++]=dat_sensecap[j].ec;
        vunkers_payload[i++]=dat_sensecap[j].sal>>8;
        vunkers_payload[i++]=dat_sensecap[j].sal;
        vunkers_payload[i++]=dat_sensecap[j].tds>>8;
        vunkers_payload[i++]=dat_sensecap[j].tds;
        vunkers_payload[i++]=dat_sensecap[j].eps>>8;
        vunkers_payload[i++]=dat_sensecap[j].eps;
    }

    HAL_UART_AbortReceive_IT(&huart4);
    uart4_IoDeInit();
}
if(sys.mod >= model8 || sys.mod <= model12)
{
    // CONTADOR 1: 12 bytes
    i=append_counter_payload(0, i);
}
if(sys.mod == model12)
{
    // CONTADORES 2 y 3: 24 bytes
    i=append_counter_payload(1, i);
    i=append_counter_payload(2, i);
}
if(sys.mod == model8 || sys.mod == model10 || sys.mod == model13
)

```

```

{
    // SHT 0 -> 4 bytes
    SHT20_read(0);
    i=append_sht2x_payload(0, i);
}
if(sys.mod == model10)
{
    // SHT 1,2,3 -> 16 bytes
    for(int j=1; j<4; j++)
    {
        SHT20_read(j);
        i=append_sht2x_payload(j, i);
    }
}
if(sys.mod == model11)
{
    // ds18b20 -> 2 bytes
    uint16_t dat_ds18b20 = DS18B20_GetTemp_SkipRom(1)*10;
    HAL_GPIO_WritePin(ONEW_PWR_GPIO_Port, ONEW_PWR_Pin,
GPIO_PIN_RESET);
    user_main_printf("DS18B20: %d",dat_ds18b20);
    vunkers_payload[i++]=dat_ds18b20>>8;
    vunkers_payload[i++]=dat_ds18b20;
}
if(sys.mod == model13){
    uint16_t adc0 = ADCModel(ADC_CHANNEL_0);
    user_main_printf("adc0: %d",adc0);
    uint16_t adc1 = ADCModel(ADC_CHANNEL_1);
    user_main_printf("adc1: %d",adc1);
    uint16_t adc2 = ADCModel(ADC_CHANNEL_2);
    user_main_printf("adc2: %d",adc2);
    uint16_t adc3 = ADCModel(ADC_CHANNEL_3);
    user_main_printf("adc3: %d",adc3);
}
printf("VUNKERS PAYLOAD (%d) -> ",i);
for (int j=0; j<i; j++)
{
    printf("%02X", vunkers_payload[j]);
}
printf("\r\n");

return i;
}

```

main.c

```
/* USER CODE END Header */
/* Includes -----
-----*/
#include "main.h"
#include "adc.h"
#include "dma.h"
#include "i2c.h"
#include "iwdg.h"
#include "usart.h"
#include "rtc.h"
#include "gpio.h"
#include "tim.h"

/* Private includes -----
-----*/
/* USER CODE BEGIN Includes */
#include "at.h"
#include "lowpower.h"
#include "nbInit.h"
#include "sensores.h"
/* USER CODE END Includes */

/* Private typedef -----
-----*/
/* USER CODE BEGIN PTD */

/* debug macro definition -----
-----*/
#ifndef ST_DEBUG
#define RXDATA_UART2_SZ 100
#define RXDATA_UART4_SZ 17
/* NBIOT task macro definition, the task in NBTASK will not be executed
after commenting out -*/
#define NBIOT

#ifndef ST_DEBUG
    #define lowpower_enter
#endif
/* USER CODE END PTD */

/* Private define -----
-----*/
/* USER CODE BEGIN PD */
/* USER CODE END PD */

/* Private macro -----
-----*/
/* USER CODE BEGIN PM */
```

```
/* USER CODE END PM */

/* Private variables -----
----*/

/* USER CODE BEGIN PV */
static uint8_t rdbuf = 0;
static uint16_t rxlen = 0;
static uint8_t rxDATA[240]={0}; //300 230ok 250nok
static uint8_t uart_user_recieve_flag = 0;
uint8_t uart2_user_recieve_flag = 0;
uint8_t uart4_user_recieve_flag = 0;

static uint8_t rdbuf_1p;
static uint8_t lpuart_recieve_flag = 0;

uint8_t rdbuf_u1 = 0; //uart2
uint8_t rxDATA_u1[RXDATA_UART2_SZ+1]={0};
volatile uint8_t rxlen_u1 = 0;
uint8_t rdbuf_u4 = 0; //uart4
uint8_t rxDATA_u4[RXDATA_UART4_SZ]={0};
uint8_t rxlen_u4 = 0;
extern uint8_t getsensorvalue_flag;
static uint8_t pwd_time_count = 0; //Password time count times
extern uint8_t controlMCU_flag;
static uint16_t dns_time_count = 0; //DNS time count times

static uint8_t task_num = _AT_IDLE; //NB task directory

static uint8_t error_num = 0; //Error count

static uint8_t uplink_time_num = 0;
extern uint8_t dns_id_flags;
extern uint8_t bc35tobc95_flags; //Switch module flag
static uint8_t dns_reset_num;
uint8_t join_network_num = 0;
uint8_t join_network_flag = 0;
uint8_t join_network_time = 0;
uint8_t join_network_timer = 0;
/* USER CODE END PV */

/* Private function prototypes -----
----*/
void SystemClock_Config(void);
/* USER CODE BEGIN PFP */
static void USERTASK(void);
/* USER CODE END PFP */
```

```
/* Private user code -----
---*/
/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/***
 * @brief  The application entry point.
 * @retval int
 */
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/
    /* Reset of all peripherals, Initializes the Flash interface and the
Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_DMA_Init();
    MX_USART1_UART_Init();
    MX_LPUART1_UART_Init();
#ifndef ST_DEBUG
    MX_IWDG_Init();
#endif
    RTC_CLK_Config();
    MX_ADC_Init();
    MX_I2C1_Init();
    MX_USART2_UART_Init();
    MX_USART4_UART_Init();
    //MX_USART5_UART_Init();
    MX_TIM2_Init();
    /* USER CODE BEGIN 2 */
```

```

new_firmware_update();
config_Get();
reboot_information_print();
product_information_print();
List_Init(sys.list);
led_on(1000);
HAL_Delay(3000);
HAL_UART_Receive_IT(&huart1,(uint8_t*)&rxbuf,RXSIZE);
HAL_UART_Receive_IT(&hluart1,(uint8_t*)&rxbuf_lp,RXSIZE);
My_UARTEx_StopModeWakeUp(&huart1); //Enable serial port
wake up
My_UARTEx_StopModeWakeUp(&hluart1); //Enable serial port
wake up
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
#ifndef NBIOT
    if((*(__IO uint8_t *)EEPROM_USER_START_FDR_FLAG) == 0x01)
    {
        if(controlMCU_flag==0)
        {
            task_num = _AT_FLAG_INIT;
        }
        else if(controlMCU_flag ==1)
        {
            task_num = _AT_IDLE;
        }
        HAL_FLASHEx_DATAEEPROM_Unlock();

        HAL_FLASHEx_DATAEEPROM_Program(FLASH_TYPEPROGRAMDATA_WORD,EEPROM_USER_STA
RT_FDR_FLAG,0x00);
        HAL_FLASHEx_DATAEEPROM_Lock();
    }
    else
    {
        if(controlMCU_flag==0)
        {
            task_num = _AT_FLAG_INIT;
        }
        else if(controlMCU_flag==1)
        {
            task_num = _AT_IDLE;
        }
    }
#else
    task_num = _AT_IDLE;
#endif
    while (1)
{

```

```

/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */

#ifndef NBIOT
if(controlMCU_flag==0)
{
    if(task_num != _AT_IDLE)
    {
        HAL_Delay(1000);

        if(NBTASK(&task_num) == _AT_ERROR)
        {
            error_num++;
        }
    }
    if((error_num > 3 && nb.uplink_flag == no_status)|| (error_num >
6))
    {
        user_main_printf("Restart the module...");
        task_num = _AT_NRB;
        error_num = 0;
    }
    if(dns_reset_num > 2 )
    {
        NVIC_SystemReset();
        dns_reset_num=0;
    }
    if(nb.recieve_flag == NB_RECIEVE && nb.uplink_flag == send)
    {
        task_num = _AT_URI;
        nb.recieve_flag = NB_IDIE;
    }
    if(getsensorvalue_flag == 1 && nb.uplink_flag == no_status)
    {
        task_num = _AT_CSQ;
        nb.uplink_flag = send;
        NBTASK(&task_num);
        getsensorvalue_flag = 0;
    }
}
#endif

USERTASK();

#ifndef lowpower_enter
    if(task_num == _AT_IDLE && uart_user_recieve_flag==0)
    {
        LPM_EnterStopMode(SystemClock_Config);
    }

```

```

#endif
}
/* USER CODE END 3 */
}

/** @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
    RCC_PeriphCLKInitTypeDef PeriphClkInit = {0};
    /** Configure the main internal regulator output voltage
    */
    __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);
    /** Configure LSE Drive Capability
    */
    HAL_PWR_EnableBkUpAccess();
    __HAL_RCC_LSEDRIVE_CONFIG(RCC_LSEDRIVE_LOW);

    /** Initializes the RCC Oscillators according to the specified
    parameters
     * in the RCC_OscInitTypeDef structure.
    */
    RCC_OscInitStruct.OscillatorType =
RCC OSCILLATORTYPE_HSI|RCC OSCILLATORTYPE_LSI
                    |RCC OSCILLATORTYPE_LSE;
    RCC_OscInitStruct.LSEState = RCC_LSE_ON;
    RCC_OscInitStruct.HSISState = RCC_HSI_ON;
    RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
    RCC_OscInitStruct.LSISState = RCC_LSI_ON;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
    RCC_OscInitStruct.PLL.PLLMUL = RCC_PLLMUL_4;
    RCC_OscInitStruct.PLL.PLLDIV = RCC_PLLDIV_2;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }
    /** Initializes the CPU, AHB and APB buses clocks
    */
    RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                                |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
}

```

```

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_1) != HAL_OK)
{
    Error_Handler();
}
PeriphClkInit.PeriphClockSelection =
RCC_PERIPHCLK_USART1|RCC_PERIPHCLK_USART2|RCC_PERIPHCLK_LPUART1
|RCC_PERIPHCLK_I2C1|RCC_PERIPHCLK_RTC;
PeriphClkInit.Usart1ClockSelection = RCC_USART1CLKSOURCE_HSI;
PeriphClkInit.Usart2ClockSelection = RCC_USART2CLKSOURCE_HSI;
PeriphClkInit.Lpuart1ClockSelection = RCC_LPUART1CLKSOURCE_HSI;
PeriphClkInit.I2c1ClockSelection = RCC_I2C1CLKSOURCE_HSI;
PeriphClkInit.RTCClockSelection = RCC_RTCCLKSOURCE_LSE;
if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
{
    Error_Handler();
}

/* USER CODE BEGIN 4 */
static void USERTASK(void)
{
    if(sys.pwd_flag==1 && sys.pwd_flag == 0 && pwd_time_count == 0)
    {
        HAL_UART_Transmit(&huart1,(uint8_t*)"Password
timeout\r\n",20,20);
    }

    if(sys.pwd_flag == 0 && uart_user_recieve_flag)
    {
        rtrim((char*)rxDATA);
        if(strcmp((char*)rxDATA,(char*)sys.pwd) == 0 &&
strlen((char*)rxDATA) < 9)
        {
            user_main_printf("Password Correct");
            sys.pwd_flag = 1;
        }
        else
        {
            user_main_printf("Password Incorrect");
        }
    }
    else if(sys.pwd_flag != 0 && uart_user_recieve_flag)
    {
        ATError_t AT_State = ATInsPro((char*)rxDATA);
        if(AT_State == AT_OK)
            printf("%s\r\n",ATError_description[AT_OK]);
        else if(AT_State == AT_PARAM_ERROR)
            printf("%s",ATError_description[AT_PARAM_ERROR]);
        else if(AT_State == AT_BUSY_ERROR)
            printf("%s",ATError_description[AT_BUSY_ERROR]);
    }
}

```

```

else if(AT_State == AT_TEST_PARAM_OVERFLOW)
    printf("%s",ATError_description[AT_TEST_PARAM_OVERFLOW]);
else if(AT_State == AT_RX_ERROR)
    printf("%s",ATError_description[AT_RX_ERROR]);
else if(AT_State == AT_ERROR)
{
    nb.usart.len = 0;
    memset(nb.usart.data,0,NB_RX_SIZE);
    rxDATA[strlen((char*)rxDATA)] = '\n';

HAL_UART_Transmit_DMA(&hlpuart1,(uint8_t*)rxDATA,strlen((char*)rxDATA));

    HAL_Delay(700); //Waiting to Send
}
}

if(uart_user_recieve_flag == 1)
{
    rxlen = 0;
    uart_user_recieve_flag = 0;
    memset(rxDATA,0,sizeof(rxDATA));
}
if(lpuart_recieve_flag == 1)
{
    printf("%s",nb.usart.data);
    nb.usart.len = 0;
    lpuart_recieve_flag = 0;
    memset(nb.usart.data,0,NB_RX_SIZE);
}

if(nb.recieve_flag == NB_RECIEVE && nb.dns_flag == running)
{
    int dns_num=3;
    nb.recieve_flag = NB_IDIE;
    nb.dns_flag = no_status;
    while(dns_num--)
    {
        HAL_Delay(1500);
        if(NBTask[_AT_QDNS].get(NULL) == NB_CMD_SUCC)
        {
            task_num = _AT_UPLOAD_START;
            dns_reset_num=0;
            break;
        }
    else
    {
        task_num = _AT_NRB;
        if(dns_num==0)
        dns_reset_num++;
    }
}

```

```

        }
    }

void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
    if(huart == &hlpuart1)
    {
        nb.usart.data[nb.usart.len++] = rxbuf_lp;
        if(task_num == _AT_IDLE)
        {
            if(rxbuf_lp == '\r' || rxbuf_lp == '\n')
            {
                lpuart_recieve_flag = 1;
            }
        }
        HAL_UART_Receive_IT(&hlpuart1,(uint8_t*)&rxbuf_lp,RXSIZE);
    }
    else if(huart == &huart1)
    {
        rxDATA[rxlen++] = rxbuf;
        if(rxbuf == '\r' || rxbuf == '\n')
        {
            uart_user_recieve_flag = 1;
        }
        HAL_UART_Receive_IT(&huart1,(uint8_t*)&rxbuf,RXSIZE);
    }
    else if(huart == &huart2)
    {
        if(sys.mod >= model7)
        {
            if(uart2_user_recieve_flag)
            {
                HAL_UART_Receive_IT(&huart2,(uint8_t*)&rxbuf_u1,RXSIZE);
                return;
            }
            if(rxbuf_u1==GPS_SOF)
            {
                rxlen_u1=0;
            }
        }

        if(rxlen_u1 < RXDATA_UART2_SZ)
            rxDATA_u1[rxlen_u1++] = rxbuf_u1;

        if((rxbuf_u1 == '\r' || rxbuf_u1 == '\n') && (rxlen_u1>0))
        {
            uart2_user_recieve_flag = 1;
        }
    }
}

```

```

        HAL_UART_Receive_IT(&huart2,(uint8_t*)&rxbuf_u1,RXSIZE);
    }
    else if(huart == &huart4)
    {
        if(sys.mod==model7)
        {
            if(uart4_user_recieve_flag==0)
            {
                if(rxbuf_u4==METEO_ID)
                {
                    uart4_user_recieve_flag=1;
                    rxlen_u4=0;
                }
            }
            else if(uart4_user_recieve_flag==1)
            {
                rxDATA_u4[rxlen_u4++] = rxbuf_u4;
                if(rxlen_u4==METEO_PAYLOAD_LENGTH)
                    uart4_user_recieve_flag=2;
            }
        }
        else if(sys.mod==model8 || sys.mod==model9)
        {
            if(uart4_user_recieve_flag==1)
            {
                rxDATA_u4[rxlen_u4++] = rxbuf_u4;
                if(rxlen_u4==SENSECAP_PAYLOAD_LENGTH)
                    uart4_user_recieve_flag=2;
            }
        }
        HAL_UART_Receive_IT(&huart4,(uint8_t*)&rxbuf_u4,RXSIZE);
    }
}

void HAL_RTC_AlarmAEventCallback(RTC_HandleTypeDef *hrtc)
{
    if(controlMCU_flag==0)
    {
        if(nb.net_flag == no_status)
            task_num = _AT_FLAG_INIT;
        else if(nb.net_flag == success && nb.uplink_flag == no_status)
        {
            task_num = _AT_CSQ;
            nb.uplink_flag = send;
        }
    }
    LPM_DisableStopMode();
}

void HAL_RTCEx_AlarmBEventCallback(RTC_HandleTypeDef *hrtc)

```

```

{
    HAL_IWDG_Refresh(&hiwdg);
        if(controlMCU_flag==0)
    {
#define NBIOT
        if(sys.pwd_flag==1)
        {
            pwd_time_count++;
            if(pwd_time_count == 17)
            {
                sys.pwd_flag = 0;
                pwd_time_count = 0;
            }
        }
#endif

        if(nb.net_flag == fail && join_network_flag==1)
        {
            task_num = _AT_CSQ;
            join_network_timer=1;
            join_network_flag=0;
        }
        if(nb.net_flag == success && nb.uplink_flag == send)
        {
            uplink_time_num++;
            if(uplink_time_num>=7)
            {
                uplink_time_num = 0;
                error_num++;
                if(sys.protocol == COAP_PRO)
                {
                    if(bc35tobc95_flags==0)
                    {
                        task_num=_AT_COAP_CLOSE;
                    }else{
                        task_num=_AT_COAP_CLOSE_BC95GV;
                    }
                }
                else if(sys.protocol == UDP_PRO) {task_num =
_AT_UDP_CLOSE;}
                else if(sys.protocol == MQTT_PRO) {task_num =
_AT_MQTT_CLOSE;}
                else if(sys.protocol == TCP_PRO) {task_num =
_AT_TCP_CLOSE;}
            }
            else
                uplink_time_num = 0;
        }
        if(dns_id_flags==1)
    }
}

```

```

{
    dns_time_count++;
    if(dns_time_count == 2400)
    {
        if(sys.protocol == COAP_PRO)           {task_num = _AT_QDNS;}
        else if(sys.protocol == UDP_PRO)      {task_num = _AT_QDNS;}
        dns_time_count=0;
    }
}

if(join_network_timer==1)
{
    join_network_time++;
    if(join_network_time==(sys.csq_time*3))
    {
        join_network_time=0;
        join_network_timer=0;
        nb.net_flag = no_status;
        task_num = _AT_CFUNOFF;
    }
}
My_AlarmInit(18,1);
}

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == GPIO_PIN_14)
    {
        if(sys.mod >= model8 && sys.mod <= model12)
        {
            Vunkers_counter_irq(0);
        }
        else if(sys.mod == model16)
            sensor.exit_count++;
        else if(nb.net_flag == success && sys.mod != model16 && (task_num
< _AT_COAP_CONFIG || task_num > _AT_TCP_CLOSE))
        {
            if(nb.uplink_flag == no_status)
            {
                task_num = _AT_CSQ;
                nb.uplink_flag = send;
                sys.exit_flag = 1;
                sensor.exit_state = 1;
                LPM_DisableStopMode();
            }
        }
    }
    else if(GPIO_Pin == GPIO_PIN_13)
}

```

```

{
    nb.receive_flag = NB_RECEIVE;
}
else if(sys.mod == model12)
{
    if(GPIO_Pin == GPIO_PIN_11)
        Vunkers_counter_irq(1);
    if(GPIO_Pin == GPIO_PIN_12)
        Vunkers_counter_irq(2);
}
/* USER CODE END 4 */

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error return state */

    /* USER CODE END Error_Handler_Debug */
}

#ifndef USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source line
number
 * @param file: where the assert_param error has occurred.
 * @param line: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line
number,
     * tex: printf("Wrong parameters value: file %s on line %d\r\n", file,
line) */
    /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) {
    Vunkers_Sensors_Timer_Flag();
}

```

Apéndice G

Código utilizado en el nodo central del primer nodo sensor para la transmisión de los datos a la BD *InfluxDB*.

```
import pandas as pd
import requests
import time
import json

while True:

    # Adecua fichero y lo lee como JSON
    with open('rx_data.json', 'r') as f:
        aux = f.read()
        aux = '[\n\r' + aux.replace('\n', ',')[:-1] + '\n\r]\0' # Agregar corchetes y eliminar la última coma
        data = json.loads(aux)

    # Se establece los valores necesarios para la conexión:
    INFLUX_TOKEN = '3Zu9aYT2zmRkzzR39VmoS-RC95BQsbdsHHMP3SX5AhtgrEZeI_Ah0t-IS50T0cz2ZdteppqUaBuLvC7aS3XREQ=='
    ORG = 'Lolo'
    ORGID = 'f386e6a8f9501a3a'
    INFLUX_CLOUD_URL='https://europe-west1-1.gcp.cloud2.influxdata.com'
    BUCKET_NAME ='Control_forestal_v0'

    # Establecemos la precisión en MILISEGUNDOS:

    QUERY_URI='{} /api/v2/write?org={}&orgID={}&bucket={}&precision=ms'.format(
        INFLUX_CLOUD_URL, ORG, ORGID, BUCKET_NAME)

    # Se establece la cabecera con el token correspondiente
    headers = {}
    headers['Authorization'] = 'Token {}'.format(INFLUX_TOKEN)
    measurement_name = 'nodo'

    # Se leen todas las filas del CSV y se van enviando a InfluxDB
    ID_Nodo = 1
    i=0
    print (data)
    for row in data:
        i=i+1
        timestamp = str(int(row['timestamp']*1000))
        Humidity = str(row['humidity'])
        Leaf_Oxygen = str(row['oxygen'])
        Leaf_CO2 = str(row['co2'])
        PH_value = str(row['pH'])
```

```
dataE = []
dataE.append('{measurement},id_nodo={id_node}
humedad_hoja={Hum},oxigeno_hoja={leaf_oxi},co2_hoja={leaf_co2},ph={ph_val
} {ts}'.

format(measurement=measurement_name,id_node=ID_Nodo,Hum=Humidity,leaf_oxi
=Leaf_Oxygen,leaf_co2=Leaf_CO2,ph_val=PH_value,ts=timestamp))

dataEnvio = '\n'.join(dataE)
r = requests.post(QUERY_URI, data=dataEnvio, headers=headers)
time.sleep(0.0002)
print(i)

# Se eliminan los datos
with open('rx_data.json', 'w') as json_empty:
    json_empty.write('')

time.sleep(1200) # Espera 20 minutos a que lleguen nuevos datos
```

Apéndice H

En este apéndice se muestra el código decodificador de las tramas transmitidas por el segundo nodo sensor para sendos dispositivos de *Dragino*.

```
/** Decoder */
// decode payload to string
var payloadStr = decodeToString(payload);
// decode payload to JSON
// var data = decodeToJson(payload);

var split = metadata.topic.split("/");
var deviceName = deviceName(payloadStr);
var deviceType = 'Gelada';

var groupName = 'nb-iot';
// use assetName and assetType instead of deviceName and deviceType
// to automatically create assets instead of devices.
//var customerName = 'customer';
// var assetName = 'Asset A';
// var assetType = 'building';

// Result object with device/asset attributes/telemetry data
var result = {
    // Use deviceName and deviceType or assetName and assetType, but not
    both.
    deviceName: deviceName,
    deviceType: deviceType,
    // assetName: assetName,
    // assetType: assetType,
    // customerName: customerName,
    groupName: groupName,
    attributes: {
        mode: mode,
        version: parseInt(payloadStr.slice(16, 20), 16),
        serialNumber: 'SN111',
        integrationName: metadata['integrationName']
    },
    telemetry: {
        rawData: payloadStr,
    }
};

var mode = parseInt(payloadStr.slice(26, 28), 16);
var telemetry = result.telemetry

switch (mode) {
    case 1:
        telemetry.mode = mode;
```

```

telemetry.battery = parseInt(payloadStr.slice(20,
    24),
    16) / 1000;
telemetry.signal = parseInt(payloadStr.slice(24,
    26), 16);
telemetry.timestamp = parseInt(payloadStr.slice(46,
    54),
    16);

telemetry.temperature_ds18b20 = parseInt(payloadStr
    .slice(28, 32), 16) / 10;
telemetry.interrupt = parseInt(payloadStr.slice(32,
    34), 16);
telemetry.ADC = parseInt(payloadStr.slice(34, 38),
    16);
telemetry.temperature = parseInt(payloadStr
    .slice(38, 42), 16) / 10;
telemetry.humidity = parseInt(payloadStr.slice(
    42, 46), 16) / 10;

break;

case 7:
    deviceType =
        telemetry.battery = parseInt(payloadStr.slice(
            20, 24), 16) / 1000;
    telemetry.signal = parseInt(payloadStr.slice(24,
        26), 16);

    telemetry.latitude = parseInt(payloadStr.slice(28,
        32), 16);
    telemetry.longitude = parseInt(payloadStr.slice(32,
        36), 16);

    telemetry.wind_dir = parseInt(payloadStr.slice(36,
        40), 16);
    telemetry.temperature = Number((parseInt(payloadStr
        .slice(40, 44), 16) / 10 - 40).toFixed(
        2));
    telemetry.humidity = parseInt(payloadStr.slice(44,
        46), 16);
    telemetry.wind_burst = parseInt(payloadStr.slice(46,50), 16);
    telemetry.wind_speed = parseInt(payloadStr.slice(50,
        52), 16);
    telemetry.rain = parseInt(payloadStr.slice(52, 56),
        16);
    telemetry.uv = parseInt(payloadStr.slice(56, 60),
        16);
    telemetry.lux = Number(((parseInt(payloadStr.slice(

```

```
    60, 66), 16) / 10) * 0.0079).toFixed(2));  
  
        //telemetry.signal = parseInt(payloadStr.slice(24,26), 16);  
        break;  
    default:  
  
        break;  
    }  
  
/** Helper functions **/  
  
function decodeToString(payload) {  
    return String.fromCharCode.apply(String, payload);  
}  
  
function decodeToJson(payload) {  
    // covert payload to string.  
    var str = decodeToString(payload);  
  
    // parse string to JSON  
    var data = JSON.parse(str);  
    return data;  
}  
  
function deviceName(payloadStr) {  
    //Split topic  
    var splitTopic = metadata.topic.split("/");  
  
    //Check if deviceName is on topic otherwise is on rawData  
    var deviceName = splitTopic.slice(-1) == "" ? "NCK" +  
        Number(  
            payloadStr.slice(1, 16)) : splitTopic.slice(-1);  
  
    return deviceName;  
}  
  
return result;
```

Apéndice I

Ya explorado e implementado todo el sistema, se hace imprescindible exponer los costes resultantes de la realización del proyecto. Este apéndice se hace necesario al otorgar una visión general y realista de la inversión que se ha de realizar para poder llevar a cabo el presente TFM.

Se procede a mostrar todos los costes que se asocian a la adquisición de materiales, sensores, herramientas HW y SW, licencias, entre otras. Todo ello para diseñar la presente red IoT.

Se comienza estudiando los distintos gastos asociados a los sensores que forman parte del sistema, estos son los que se muestran a continuación:

Unid	Modelo	IVA	Precio unid	Precio total
1	<i>LAT-B3</i>	21%	225,00 €	272,25 €
1	<i>Sensecap</i>	21%	81,25 €	98,31 €
1	<i>PYTHOS-31</i>	21%	180,00 €	217,80 €
1	<i>SEN0322</i>	21%	63,99 €	77,43 €
1	<i>MH-Z19C</i>	21%	45,00 €	54,45 €
1	<i>Soil NPK Sensor</i>	21%	52,55 €	63,59 €
1	<i>SEN0249</i>	21%	116,85 €	141,39 €
1	<i>314990620</i>	21%	79,74 €	96,49 €
<i>Sainlogic WS 3500 Plus. Estación Meteorológica</i>		21%	185,00 €	447,70 €
1	<i>DS18B20</i>	21%	3,49 €	4,22 €
1	<i>SHT20</i>	21%	8,23 €	9,96 €
1	<i>NEO-6M</i>	21%	18,16 €	21,97 €
1	<i>YF-S201</i>	21%	10,99 €	13,30 €
1	<i>SEN0313</i>	21%	41,75 €	50,52 €
Total			1.569,38€	

En la tabla siguiente se expone el coste asociado al resto de componentes que forman parte del sistema. Se puede observar los gastos asociados a la alimentación, transmisión de datos y desarrollo de la red.

Unid	Modelo	IVA	Precio unid	Precio total con descuentos
1	Batería Gens Ace	21%	114,88 €	139,00 €
1	Batería ES-261520/W	21%	17,00 €	20,57 €
1	Placa solar	21%	0,38 €	0,46 €
5	MMPCBAv1.0	21%	0,38 €	2,30 €
5	MMPCBSv1.0	21%	0,38 €	2,30 €
5	MMPCBHv1.0	21%	0,38 €	2,30 €
2	Raspberry Pi 4	21%	80,00 €	193,60 €
1	Arduino UNO	21%	24,00 €	29,04 €
2	Módulo Ra-02 LoRa SX1278	21 %	7,43 €	17,98 €
2	DollaTek 433MHz	21 %	1,49 €	3,60 €
1	NBSN95	21%	44,80 €	54,21 €
1	LSN50	21%	49,50 €	59,90 €
5	MAX485	21%	1,16 €	7,00 €
3	Cables	21%	2,46 €	8,93 €
1	Circuito impreso	21%	1,88 €	2,27 €
10	Amplificador Operacional OP292GSZ-REELCT-ND	21%	6,20 €	55,69 €
15	Condensador 1uF 1276-6470-1-ND	21%	0,10 €	1,11 €
15	Condensador 15 uF 399-C1210C156K8PAC7800CT-ND	21%	0,82 €	9,35 €
10	Resistencia l LT5400BCMS8E-1#PBF-ND	21%	9,38 €	87,97 €
10	Resistencia LT5400BCMS8E-2#PBF-ND	21%	9,38 €	87,97 €
10	Resistencia 505-LT5400BCMS8E-6#PBF-ND	21%	9,38 €	87,97 €
15	Convertidor 1470-VR05S05-ND	21%	2,50 €	23,48 €
10	Resistencia LT5400BHMS8E-7#PBF	21%	11,12 €	101,11 €
10	Buffer LM358DR2G	21%	0,42 €	3,40 €
5	Convertidor K7805-3AR3	21%	4,27 €	21,08 €
5	Convertidor R-739.0P	21%	27,11 €	132,38 €
10	Regulador LM317LID	21%	0,88 €	7,85 €
10	Resistencia RG3216P-1000-B-T1	21%	0,56 €	3,98 €
10	PMOS FDN338P	21%	0,41 €	3,50 €
10	Regulador LM1085ISX-5.0/NOPB	21%	1,90 €	17,06 €
10	Resistencia FDV303N	21%	0,39 €	2,91 €
10	NMOS TSR 1-2490	21%	8,48 €	83,52 €
20	Conversor 450LLE22MEFC12.5X25	21%	1,74 €	28,14 €
50	Condensador 22 uF EKXF451ELL100MJ20S	21%	1,00 €	34,27 €
10	Condensador 10 uF C0603X103J4HAC7867	21%	0,18 €	1,18 €
10	Condensador 0,1 uF CL10B104MB8NNNC	21%	0,09 €	0,24 €
1	Iot gateway lora 8 mikrotik knot lr8 kit	21 %	150,04 €	181,55 €
TOTAL				1.519,17 €

$$\text{Materias primas} = \text{Presupuesto en sensores} + \text{resto del sistema} = 3088,55 \text{ €}$$

En la siguiente tabla se muestra el coste del ejemplar adquirido (terrenos y bienes naturales) para poder analizar el sistema de monitorización implementado.

Cantidad	Modelo	IVA	Precio unid	Precio total
1	Pino	10 %	26,86 €	29,55 €
TOTAL				29,55 €

En la tabla que se muestra a continuación se recogen los gastos en transporte, tanto del material solicitado, como del transporte del ejemplar.

Cantidad	Modelo	Precio unid	Precio total
1	<i>Transportes de componentes y plantas</i>	123,70 €	123,70 €
TOTAL			123,70 €

En la posterior tabla, se muestran los gastos asociados al salario de los participantes en el proyecto.

Categoría	Horas	Coste/hora	Total
<i>Tutor proyecto</i>	20h	30,00 €	600,00 €
Ingeniero	200h	19,00 €	3.800,00 €
TOTAL			4.400 €

A su vez, en la siguiente tabla se muestran los gastos en impuestos a la Seguridad Social.

Categoría	Total (S.S)
S.S del Tutor proyecto	60,75 €
S.S del Ingeniero	256,50 €
Total	317,25 €

En la tabla que se muestra a continuación, se desarrollan los gastos asociados al material informático precisado.

Equipo	Precio	Vida útil (años)	Uso (meses)	Coste
PC	1.900,00 €	8	9	178,13 €
Ordenador portátil	800,00 €	8	5	66,66 €
ADALM-2000	267,57 €	4	9	50,17 €
Total				294,96 €

En la siguiente tabla, se exponen los gastos asociados a las licencias de las herramientas SW utilizadas para el desarrollo y simulación del proyecto. También, vienen reflejados los precios asociados al uso de las BD.

Categoría	Total
Arduino IDE	0,00 €
KiCad 7.0	0,00 €
Keil µ Vision 5	4.987,83 €
MATLAB	262,00 €
Draw.io	0,00 €
Office	30,00 €
InfluxDB Cloud	230,54 €
ThingsBoard	9,22 €
Ecowitt	0 €
Wunderground	0 €
Wheathercloud	0 €
WheatherObservations	0 €
Tarjeta SIM	10,00 €
Total	5.529,69€

Finalmente, se recoge en la tabla que viene a continuación los gastos anteriormente vistos, divididos en costes fijos y variables.

Costes fijos	Valor	Costes variables	Valor
<i>Sueldos y salarios personal</i>	4.400,00 €	<i>Materias primas</i>	3.088,55 €
<i>Seguridad social</i>	317,25 €	<i>Terrenos y bienes naturales</i>	29,55 €
<i>Recursos y equipos informáticos</i>	294,96 €	<i>Transporte</i>	123,70 €
<i>Licencias de herramientas de diseño y BD</i>	5.529,69 €		
Total costes fijos	10.541,90 €	Total costes variables	3.241,80 €

$$\text{Total costes} = \text{costes fijos} + \text{costes variables} = 13.783,70 €$$