ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELECTRÍCA Y ELECTRÓNICA

DISEÑO Y SIMULACIÓN DEL SISTEMA DE ADQUISICIÓN DE DATOS EN IOT CON LORAWAN PARA EL PROCESO DE SECADO DEL CAFÉ.

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN.

SEBÁSTIAN ALEXANDER SÁNCHEZ MACAS

sebastian.sanchez@epn.edu.ec

DIRECTOR: Dr. GEOVANNY DANILO CHÁVEZ GARCÍA

danilo.chavez@epn.edu.ec

Quito, enero 2021

CERTIFICACIONES

Yo, Sebástian Alexander Sánchez Macas declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

SEBÁSTIAN ALEXANDER SÁNCHEZ MACAS

Certifico que el presente trabajo de integración curricular fue desarrollado por Sebástian Alexander Sánchez Macas, bajo mi supervisión.

Dr. GEOVANNY DANILO CHÁVEZ GARCÍA
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

SEBÁSTIAN ALEXANDER SANCHEZ MACAS

Dr. GEOVANNY DANILO CHÁVEZ GARCÍA

DEDICATORIA

A mi madre y abuelita por todo el esfuerzo, paciencia y apoyo incondicional durante toda mi vida, este logro es por y para ustedes.

AGRADECIMIENTO

A mi madre Rocío por su apoyo incondicional en todo momento, por todo el esfuerzo que hizo para que pueda cumplir este objetivo, gracias por creer en mí y por todo el apoyo que me ha dado a lo largo de estos años.

A mi abuelita Isabel por ser la persona que estuvo conmigo en todo momento apoyándome, cuidándome y aconsejándome.

A mis hermanos Dilan y Brandon y a mi hermana Hanna por su apoyo y por todos los momentos que hemos podido compartir juntos.

Para Ana que es una parte importante en mi vida y que gracias a todo su apoyo incondicional pude llegar a este momento.

Al Dr. Danilo Chávez por su ayuda, consejos y conocimiento en la elaboración de este proyecto.

A la Escuela Politécnica Nacional y todos sus profesores por todo el conocimiento y formación que supieron brindarme en una de las etapas más importantes de mi vida.

ÍNDICE DE CONTENIDO

CERTIFICACIONES I
DECLARACIÓN DE AUTORÍA II
DEDICATORIA III
AGRADECIMIENTO IV
ÍNDICE DE CONTENIDO V
ÍNDICE DE TABLAS VII
ÍNDICE DE FIGURAS VII
RESUMEN X
ABSTRACT XI
1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO 1
1.1. Objetivo general 2
1.2. Objetivos específicos 2
1.3. Alcance 2
1.4. Marco teórico 3
1.4.1. El café en el Ecuador 3
1.4.2. Proceso de Secado del Café 5
1.4.3. Adquisición de datos 8
1.4.4. Microcontroladores 11
1.4.5. Internet de las cosas (IoT) 13
1.4.6. LoRa 13
2. METODOLOGÍA 16
2.1. Elementos Seleccionados 16
2.1.1. Microcontrolador 16
2.1.2. Sensor 18
2.1.3. Batería 18
2.2. Diseño y programación del sistema de adquisición de datos 19
2.2.1. Diseño del nodo final con Arduino 20
2.2.2. Enlace del nodo final a un servidor mediante LoRaWAN 25
2.3. Algoritmo de control 39
2.3.1. Obtención de datos 39
2.3.2. Diseño de la planta 39
2.3.3. Diseño del controlador 41
2.3.4. Simulación del controlador 41
3. RESULTADOS, CONCLUSIONES Y RECOMENDACIONES 43

3.1. Resultados ***43***
3.1.1. Conectividad entre el nodo final y la red TTN 43
3.1.2. Comunicación entre la red TTN y aplicaciones IoT 48
3.1.3. Simulación del controlador PID 53
3.2. Conclusiones ***58***
3.3. Recomendaciones ***60***
4. REFERENCIAS BIBLIOGRÁFICAS 61
5. ANEXOS 63
ANEXO I ***64***
ANEXO II ***65***
ANEXO III ***66***
ANEXO IV ***69***
ANEXO V ***70***

ÍNDICE DE TABLAS

Tabla 1.1 Clasificación del café según la altitud de la zona de cultivo [1].....4
Tabla 1.2 Clasificación del café según el método de beneficio del café [1]. 4
Tabla 1.3 Clasificación del café según las características físicas y organolépticas del grano (arábigo) [1]. 4
Tabla 1.4 Clasificación del café según las características físicas y organolépticas del grano (robusto) [1]. 5
Tabla 1.5 Área de secado necesaria en la finca [8]. 6
Tabla 1.6 Especificación técnica del sensor DHT11 [10]. 9
Tabla 1.7 Especificación técnica del sensor AM2303 [3]. 10
Tabla 1.8 Especificación técnica de la placa Arduino MKR WAN 1310 [14]. 12
Tabla 1.9 Comunicaciones inalámbricas par aplicaciones IoT [17]. 14
Tabla 1.10 Tecnologías LPWAN [17]. 14
Tabla 2.1 Comparación de microcontroladores Arduino MKR. 16
Tabla 2.2 Especificación técnica de la batería UBP103450/PCM [25]. 18
Tabla 2.3 Parámetros método de Smith.....40

ÍNDICE DE FIGURAS

Figura 1.1 Secador solar parabólico [8]. 7
Figura 1.2 Procedimiento de secado al sol [9]. 7
Figura 1.3 Procedimiento de secado mecánico [9]. 8
Figura 1.4 Sensor de Temperatura y Humedad DHT11 [10]. 9
Figura 1.5 Sensor de Temperatura y Humedad AM2302/DHT22 [11]. 10
Figura 1.6 Tarjeta Arduino MKR WAN 1310 [13]. 11
Figura 1.7 Arquitectura LoRaWAN de doble estrella [18]. 15
Figura 1.8 Distribución por capas de la tecnología LoRa [20]. 15
Figura 2.1 Diagrama de distribución de pines Arduino MKR 1310 [14]. 17
Figura 2.2 Sensor AM2302 [24]. 18
Figura 2.3 Batería UBP103450/PCM Li-Po recargable [25]. 19
Figura 2.4 Ruta de comunicación LoRaWAN [26]. 20
Figura 2.5 Sitio de descarga del software Arduino IDE [27]. 20
Figura 2.6 Ventana de inicio Arduino IDE. 21
Figura 2.7 Instalación de la tarjeta Arduino MKR WAN 1310 en Arduino IDE. 21
Figura 2.8 Selección tarjeta MKR WAN. 22
Figura 2.9 Adición librería DHT. 22
Figura 2.10 Instalación librería DHT. 23
Figura 2.11 Instalación de dependencias de la librería DHT. 23
Figura 2.12 Fragmento 1 código configuración del sensor DHT. 23
Figura 2.13 Fragmentos 2 código lectura del sensor DHT. 24
Figura 2.14 Tiempo de comunicación del sensor AM2302/DHT22 [23]. 24
Figura 2.15 Fragmento 3 código visualización de datos. 25
Figura 2.16 Esquema de conexión nodo final. 25
Figura 2.17 Adición librería MKRWAN. 26
Figura 2.18 Instalación librería MKRWAN. 26
Figura 2.19 Gateways a nivel global de la red TTN [28]. 27
Figura 2.20 Registro a TTN [29]. 27
Figura 2.21 Consola TTN. 28
Figura 2.22 Seleccionar región TTN. 28
Figura 2.23 Inicio de sesión en TTS 28
Figura 2.24 Consola TTN. 29
Figura 2.25 Crear nueva aplicación en TTN. 29
Figura 2.26 Añadir dispositivo a la aplicación. 30
Figura 2.27 Registro del nodo final. 30
Figura 2.28 Datos de registro nodo final. 31
Figura 2.29 Nodo final correctamente registrado en TTN. 31
Figura 2.30 Configuración de localización del nodo final en TTN. 32
Figura 2.31 Código para obtener el numero EUI de la tarjeta Arduino MKRWAN 1310. 32
Figura 2.32 Creación del archivo de cabecera. 33
Figura 2.33 Guardar credenciales como archivo de cabecera. 33
Figura 2.34 Fragmento 1 código inclusión de librerías. 34
Figura 2.35 Fragmento 2 código configuración general. 34
Figura 2.36 Fragmento 3 código conversión de los datos del sensor a cadena de caracteres. 35
Figura 2.37 Fragmento 4 código conversión del dato de humedad a hexadecimal. 35
Figura 2.38 Fragmento 5 código envío de datos a la red TTN. 35
Figura 2.39 Fragmento 6 código envío de datos concatenados a la red TTN. 36
Figura 2.40 Fragmento 7 código recepción de datos de la red TTN al nodo final. 36
Figura 2.41 Decodificador de la red TTN. 36
Figura 2.42 Selección del decodificador de la red TTN. 37
Figura 2.43 Código y test de decodificación de datos en la red TTN. 37

Figura 2.44 Plataformas IoT compatibles con TTN. 38
Figura 2.45 Plataformas IoT integradas a la aplicación en TTN. 38
Figura 2.46 Porcentaje de humedad relativa del gano de café con respecto al tiempo [32]. 39
Figura 2.47 Aplicación método de Smith. 40
Figura 2.48 Configuración del controlador PID. 42
Figura 2.49 Ajuste de parámetros del controlador PID. 42
Figura 3.1 Método de activación OTAA [33]. 44
Figura 3.2 Simulación de envío de mensaje a la red TTN. 44
Figura 3.3 Envío de mensaje exitoso. 44
Figura 3.4 Visualización en vivo del mensaje enviado. 45
Figura 3.5 Sistemas de numeración [34]. 45
Figura 3.6 Simulación envío de datos del nodo final a la red TTN. 46
Figura 3.7 Visualización en vivo de los datos enviados. 46
Figura 3.8 Simulación envío de datos de humedad y temperatura a la red. 47
Figura 3.9 Simulación envío de datos de humedad y temperatura a la red. 47
Figura 3.10 Tablero principal AnyViz. 48
Figura 3.11 Simulación recepción de datos desde la red TTN a AnyViz. 49
Figura 3.12 Simulación recepción de datos desde la red TTN a AnyViz. 49
Figura 3.13 Simulación recepción de datos desde la red TTN a AnyViz. 50
Figura 3.14 Panel de alarmas de AnyViz. 50
Figura 3.15 Tablero principal Datacake. 51
Figura 3.16 Visualización de datos en Datacake. 51
Figura 3.17 Simulación recepción de datos desde la red TTN a Datacake. 52
Figura 3.18 Simulación recepción de datos desde la red TTN a Datacake 52
Figura 3.19 Simulación recepción de datos desde la red TTN a Datacake 53
Figura 3.20 Interfaz de Datacake para móviles. 53
Figura 3.21 Planta simulada en Matlab Simulink. 54
Figura 3.22 Simulación modelo aproximado de la planta. 54
Figura 3.23 Controlador PID simulado en Matlab Simulink. 54
Figura 3.24 Respuesta del controlado PID. 55
Figura 3.25 Respuesta del controlado PID ajustado. 55
Figura 3.26 Respuesta del controlador ante cambios de referencia. 56
Figura 3.27 Respuesta del controlador ante cambios de referencia. 57

RESUMEN

En este trabajo se presenta el diseño y simulación de un sistema de adquisición de datos con tecnología de comunicación LoRaWAN, diseñando un nodo final que se conecta a la red de The Things Network para posteriormente vincularse con aplicaciones IoT para la visualización y análisis de datos.

En el capítulo 1 se definen los conceptos básicos relacionados con la tecnología LoRa y el protocolo de comunicación LoRaWAN, se determina las variables influyentes en el proceso de secado del café, se detalla las características de la tarjeta Arduino y de diversos sensores y se analiza la importancia del IoT en las aplicaciones modernas vinculadas a LoRa.

En el capítulo 2 se detalla y justifica la selección de los elementos para el diseño del nodo final, se realiza la programación para la adquisición de datos del sensor y la programación para la comunicación del nodo final con la red mediante LoRaWAN, se detalla el procedimiento a seguir para vincular la red a aplicaciones IoT y se realiza el algoritmo de control para la humedad en el proceso de secado del café.

En el capítulo 3 se muestran las simulaciones de envió de datos del nodo final a la red y de la red hacia las aplicaciones IoT, las interfaces gráficas, histogramas, mapas etc. que ofrecen estas aplicaciones para la visualización y análisis de datos, y la respuesta del controlador diseñado ante cambios de referencia. Se señalan las conclusiones y recomendaciones del desarrollo del trabajo en función del resultado de las simulaciones.

PALABRAS CLAVE: LoRa, LoRaWAN, nodo final, gateway, TTN, IoT.

ABSTRACT

This Degree Work presents the design and simulation of a data acquisition system with LoRaWAN communication technology, designing a final node that connects to The Things Network to later be linked to IoT applications for data visualization and analysis

Chapter 1 defines the basic concepts related to LoRa technology and the LoRaWAN communication protocol, determines the influential variables in the coffee drying process, details the characteristics of the Arduino card and various sensors, and analyzes the importance of IoT in modern applications linked to LoRa.

Chapter 2 details and justifies the selection of the elements for the design of the final node, the programming for the acquisition of sensor data and the programming for the communication of the final node with the network through LoRaWAN is carried out, the procedure is detailed. to follow to link the network to IoT applications and the control algorithm for humidity in the coffee drying process is carried out.

Chapter 3 shows the simulations of sending data from the end node to the network and from the network to IoT applications, graphical interfaces, histograms, maps, etc. offered by these applications for data visualization and analysis, and the response of the designed controller to reference changes. The conclusions and recommendations of the development of the work are indicated based on the result of the simulations.

KEYWORDS: LoRa, LoRaWAN, final node, gateway, TTN, IoT.

1. DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

La caficultura es una actividad con suma importancia económica, social y ambiental en el Ecuador, la producción de café significa generación de ingresos tanto para los caficultores como para los acopiadores, transportistas y comerciantes. [1]

El proceso de producción de café en el Ecuador es uno de los más tradicionales y artesanales vigentes en la actualidad, para una correcta producción de un café de calidad es de vital importancia la presencia de personal altamente capacitado en el ámbito, la calidad del café dependerá tanto de la destreza del personal como de sus conocimientos en el área, el problema radica en que dichos conocimientos no pueden ser replicados tan fácilmente, es por esta razón que se plantea desarrollar un sistema de adquisición de datos con ayuda de IoT, por medio del protocolo LoRaWAN.

LoRaWAN es un protocolo desarrollado por LoRa Alliance®. LoRa Alliance® es una asociación abierta y sin fines de lucro, donde sus miembros trabajan y comparten experiencias para promover, mejorar e impulsar el desarrollo del protocolo estándar LoRaWAN como un estándar global abierto para conectividad IoT LPWAN con el fin de abordar una amplia gama de aplicaciones IoT estáticas y móviles.

Hoy en día la tecnología del internet de las cosas o mejor conocido por sus siglas en ingles IoT está en auge y constante evolución, IoT nos permite conectar todo tipo de objetos y dispositivos usando diferentes medios de red, siendo el objetivo principal facilitar las actividades del ser humano volviéndolas mucho más dinámicas y convenientes. [2]

El usar tecnologías IoT con conectividad LoRaWAN nos permite enlazarnos en una red de largo alcance y consumiendo la menor cantidad de potencia posible, de este modo transmitiendo los datos brindados por los sensores instalados en el proceso donde es muy complicado tener fuentes de energías potentes y constantes, como en el proceso del secado de café, permitiendo que cualquier persona con autorización previa pueda observar en cualquier instante el estado de los sensores, los datos adquiridos, los histogramas desarrollados con el tiempo y las diferentes interfaces gráficas que nos ofrece las aplicaciones IoT.

El tener todos estos datos y herramientas en cualquier momento tan solo con conectarnos a la red nos permite tener un análisis a profundidad del estado del café en cualquier instante. Con esta recopilación de datos se espera que expertos en el área de la producción de café puedan encontrar un patrón de guía en el cual se desglosen las condiciones óptimas para producir un café de altísima calidad y así replicarlo con mucha mayor facilidad.

1.1. Objetivo general

El objetivo general de este Trabajo de Integración Curricular es: Diseñar y Simular un sistema de adquisición de datos basado en IoT con LoRaWAN pera el proceso de secado del café.

1.2. Objetivos específicos

Los objetivos específicos de este Trabajo de Integración Curricular son:

1. Determinar las variables importantes a ser adquiridas y procesadas sobre el proceso de secado del café.

2. Determinar los sensores y actuadores requeridos en el proceso del secado del café.

3. Analizar los conceptos sobre IoT con la tecnología de LoRaWAN con el fin de determinar los componentes requeridos

4. Diseñar y simular el sistema de adquisición de datos basados en IoT con LoRaWAN para el proceso de secado del café.

5. Realizar el algoritmo de control para determinar la humedad del grano del café.

6. Realizar pruebas de la adquisición de datos del sistema propuesto.

1.3. Alcance

En el presente Trabajo de Integración Curricular se obtendrá una base de datos sobre el proceso de secado del café, mismo que se lleva a cabo dentro de un invernadero estilo domo, los datos serán adquiridos por medio de una serie de sensores instalados dentro del domo, estos sensores son de temperatura y humedad, datos de suma importancia en el proceso de secado del café.

Se usará un sensor de temperatura y humedad para la adquisición de datos, la salida del sensor es digital por lo que podemos esperar una buena precisión en los datos adquiridos. Este sensor posee un rango de medición de temperara de -40ºC a 125 ºC con una precisión de -+ 0.2 ºC a 25 ºC y una resolución de 0.1 ºC (8-bit), mientras que por el lado de la humedad posee un rango de 0% RH a 100% RH con una precisión de -+ 2% RH y con una resolución de 0.1% RH [3]. Este sensor tiene un bajo costo y fácil implementación con sistemas Arduino.

Los datos adquiridos de temperatura y humedad serán procesados por una tarjeta Arduino que incluye un módulo radio LoRa, este módulo se encargara de enviar los datos por medio del protocolo LoRaWAN hacia un Gateway o Concentrador y a su vez el Gateway se conectara a un “network server” para posteriormente enlazarse con un “application server”.

Con los datos enviados hacia el servidor, pueden ser visualizados por medio de una página web visitando las diversas aplicaciones IoT disponibles, ya sea en un dispositivo móvil o en

un ordenador, en estas aplicaciones IoT se mostrarán los datos adquiridos además de una serie de histogramas, niveles de temperatura y humedad y gráficos relacionados con la adquisición de datos, todo en una interfaz agradable al usuario.

El sistema estará alimentado por una pequeña batería Li-Po recargable, la cual sustentará tanto a la placa Arduino como al sensor implementado, el sistema contará con una adecuada protección y mantenimiento periódico para asegurar su funcionamiento.

El sistema se ubicará dentro del domo de secado de café para si obtener los datos periódicamente, el sistema debe ser aislado del ingreso de agua u otras sustancias que puedan comprometer su correcto funcionamiento.

El algoritmo de control de la humedad se realizará considerando que la cámara de sacado es un domo cerrado, por el cual se permite el paso de los rayos solares para el secado del grano.

Finalmente, las pruebas de adquisición de datos se los realizara de forma simulada y aproximada ya que por el momento en Ecuador se dispone de pocos Gateways construidos con los cuales se pueda enlazar a la red LoRa. La construcción de un Gateway requiere muchos más elementos fuera del alcance de este proyecto.

1.4. Marco teórico

1.4.1. El café en el Ecuador

La caficultura para los ecuatorianos es una actividad con una destacada importancia económica, social y ambiental. El café en el país significa generación de ingresos para los caficultores, acopiadores, transportistas y comercializadores, así como ingresos de divisas que ayudan a intensificar la economía rural en los territorios productores.

El café, además, cumple un importante papel social en Ecuador, directamente relacionado con la participación de los diversos pueblos y etnias, hombres y mujeres, en los procesos de producción, transformación y comercialización del café, así como en la generación de empleo, especialmente, durante las actividades de cosecha. Los sectores cafeteros en 23 de las 24 provincias del país conforman una amplia red social con un gran impacto multisectorial. [1]

La categorización del café ecuatoriano se rige por la Norma Técnica Ecuatoriana NTE INEN 285:2006, la cual tiene por objeto establecer la clasificación y los requisitos del café verde en grano. Esta norma, además de la obvia diferenciación de los cafés según su especie, contempla otros elementos claves como la altitud de cultivo, el método de beneficio o las características físicas para definir y clasificar el café producido en este país. [1].

Según la altitud de la zona de cultivo:

Tabla 1.1 Clasificación del café según la altitud de la zona de cultivo [1].

Café de estricta altura	Se produce en las zonas ubicadas más arriba de los 1200 m.s.n.m.
Café de altura	Se produce entre 800 y 1200 m.s.n.m.
Café estándar	Se produce en las zonas por debajo de los 800 m.s.n.m.

Según el método de beneficio del café.

Tabla 1.2 Clasificación del café según el método de beneficio del café [1].

Natural	Cuando el beneficio es realizado por la vía seca.
Lavado	Cuando el beneficio es por vía húmeda. En la actualidad también se prepara café lavado mediante los métodos: húmedo enzimático y subhúmedo (beneficio con equipos desmucilaginadores)
Semilavado	También conocido como café “honey”, el método que se basa en la cosecha de frutos maduros, despulpado, secado con todo mucílago y trilla.

Ecuador al ser un país privilegiado en clima, altura y posición geográfica produce un café de excelente calidad, considerado como producto de exportación, entre sus principales variedades exportables están el café arábigo y el café robusto, ambos son cultivados en las cuatro regiones del país según información del Ministerio de Agricultura, Ganadería, Acuacultura y Pesca – MAGAP. [4]

La clasificación del café según las características físicas y organolépticas del grano es:

Café arábigo:

Tabla 1.3 Clasificación del café según las características físicas y organolépticas del grano (arábigo) [1].

Café Grado 1	Granos de café Arábica lavado de la cosecha actual, con beneficio húmedo óptimo, de tamaño uniforme y estricta altura, de olor intensamente fresco, color homogéneo y con una calidad organoléptica de medio alto a alto.
Café Grado 2	Granos de café Arábica lavado de la cosecha actual, bien beneficiado, provenientes de zonas altas o de baja altura, de olor fresco, de tamaño y color homogéneos y con una calidad organoléptica superior a la media. Este café se conoce como supremo.
Café Grado 3	Granos de café Arábica lavado de la cosecha actual, bien preparados, de olor fresco, de tamaño y color homogéneos y de taza sin defectos.
Café Grado 4	Granos de café Arábica beneficiados por la vía seca, de cosecha actual con taza limpia y libre de sabores extraños.

Café robusto:

Tabla 1.4 Clasificación del café según las características físicas y organolépticas del grano (robusto) [1].

Café Grado 1	Grano de café Robusto beneficiado por la vía húmeda, de cosecha actual, tamaño grande, taza limpia y libre de sabores extraños.
Café Grado 2	Grano de café Robusto beneficiado por la vía seca, de cosecha actual, tamaño grande, taza limpia y libre de sabores extraños.
Café Grado 3	Grano de café Robusto beneficiado por la vía seca, de cosecha actual, tamaño de pequeño a mediano, taza limpia y libre de sabores extraños.

El café en el Ecuador es una de las fuentes más grandes de economía en el país, todo esto debido a la diversidad de sus suelos y la calidad de café que se puede llegar a producir.

El precio del café varia en torno a su calidad, llegando a costar más de $500 dólares un quintal de café de excelente calidad, es por este motivo que es necesario identificar las variables más influyentes al momento de obtener un café de alta calidad.

1.4.2. Proceso de Secado del Café

La cosecha de café en Ecuador se realiza manualmente con obreros y productores, quienes recolectan los frutos semi-maduros y maduros de las plantaciones de café. Estos granos son despulpados con máquinas despulpadoras para ser almacenados por un día; después, lavados y secados en diferentes condiciones [5].

El secado de café es uno de los procesos más complicados a comparación de otros tipos de granos ya que después del proceso del lavado el grano de café contiene aproximadamente una humedad del 55% el cual debe ser reducido para continuar con el proceso de tostado; si se emplean altas temperaturas puede afectar gravemente el aroma y sabor del café [6].

Por este motivo el proceso de secado es de suma importancia debido a que, gracias al tipo de secado, se puede percibir diferentes sabores en el café. Algunos métodos de secado logran que el café tenga sabores más frutales, otros que sea más dulce o incluso que sea una taza agradable, limpia y fácil de disfrutar. Muchos cafés desarrollan sabores exóticos y únicos gracias al proceso de secado [7].

La calidad del grano seco está determinada por el conjunto de características físicas y organolépticas que motivan a un comprador a pagar un precio diferenciado por el producto, lo que representa mayor ingreso y rentabilidad al caficultor [5].

El tueste permite que el café desarrolle e intensifique sus sabores y para tostar los granos se

debe reducir la humedad aplicando los distintos métodos de secado desarrollados con el tiempo.

Como norma vigente para la comercialización, el café pergamino seco debe tener entre el 10% y el 12% de humedad [8].

1.4.2.1. Secado al sol

Tradicionalmente el café se seca con el sol, hoy en día existen procesos de secados con máquinas especializadas en esa labor, sin embargo, en el país el método de secado más utilizado continúa siendo el secado al sol, debido a que las fincas productoras de café generan una cantidad menor a 500 arrobas de café seco al año, por lo que no les es rentable usar métodos industriales de secado.

El secado de café se realiza en patios de cemento, carros secadores, elbas o casa elbas y marquesinas o secadores parabólicos, siendo este último el método más eficiente en secado al sol [8].

El área de secado que se necesita depende estrictamente de la cantidad de café producido al año, en la siguiente tabla se pude apreciar la distribución recomendada.

Tabla 1.5 Área de secado necesaria en la finca [8].

Producción de la finca [@/año]	Área de secado [m^2]
40	12
60	18
80	24
100	30
200	6
300	90
400	120
500	150

El secador solar parabólico es una plancha de cemento cubierta con una estructura de madera y recubierta con plástico, que permite la circulación interna del aire [8].

Figura 1.1 Secador solar parabólico [8].

La altura recomendada para el secador solar parabólico es de 2.10 [m] y en su totalidad debe ser recubierto por plástico, así evitando alteraciones en la temperatura interior del domo.

La altura de grano de café colocado en las planchas de secado no debe superar los 3.5 [cm] ya que así se garantiza un secado uniforme, siempre y cuando se revuelva el café por lo menos cuatro veces al día

Aplicando dicha técnica el tiempo requerido para secar el café está entre 7 y 15 días, dependiendo de la temperatura del lugar y las lluvias [8].

Figura 1.2 Procedimiento de secado al sol [9].

1.4.2.2. Secado mecánico

El secado de café mecánico se recomienda para fincas con producciones que superen las 500 arrobas al año.

El secado mecánico del café se realiza en cuartos cerrados o cámaras especializadas, en la cuales se introduce aire caliente a una temperatura máxima de 50°C, impulsado por un ventilador, el cual atraviesa la masa del fruto de café, el aire se puede calentar mediante estufas o quemadores, siendo su principal fuente de energía la quema de combustibles, carbón mineral o energía eléctrica [8].

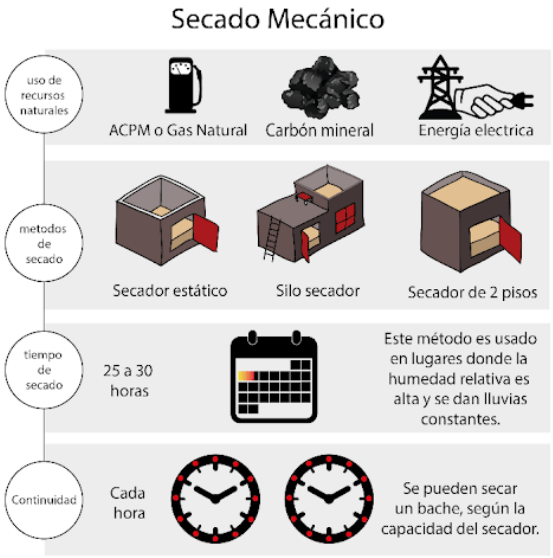

Figura 1.3 Procedimiento de secado mecánico [9].

El tiempo de secado demora normalmente entre 25 a 30 horas, este método permite alcanzar el grado de humedad recomendado en el grano de café en un corto lapso, sin embargo, expertos afirman que al usar un secado mecánico se pierden varias propiedades que brindan aroma y sabor al café.

1.4.3. Adquisición de datos

En la industria del café existen distintas variables de gran importancia para determinar la calidad de este, dentro de estas variables las que destacan y tienen prioridad en ser adquiridas son la temperatura y humedad, justamente dichas variables son las principales influyentes en el proceso de secado del café.

Como se analizó anteriormente el secado consiste en reducir el contenido de humedad presente en el pergamino húmedo de café del 55 % (aproximadamente) hasta un 12 %, puesto que para ser almacenado el grano de café debe poseer un grado de humedad bajo. [6]

Uno de los objetivos de esta investigación es poder medir y controlar la humedad en tiempo real en el proceso de secado de café, así obteniendo los datos de humedad y temperatura de forma precisa y continua, mediante la instalación de un sensor sencillo de temperatura y humedad relativa (AM2302/DHT22) ubicado dentro de los secadores solares parabólicos, específicamente cerca de las planchas de cemento donde se encuentra el grano de café.

1.4.3.1. Sensor de temperatura y humedad relativa (DHT11)

Este sensor de temperatura y humedad DHT11 cuenta con un componente de medición de humedad de tipo resistivo y un componente de medición de temperatura NTC, y se conecta a un microcontrolador de 8 bits de alto rendimiento para su posterior procesamiento de datos, el sensor que ofrece excelente calidad y tiempo de respuesta [10].

Figura 1.4 Sensor de Temperatura y Humedad DHT11 [10].

El sensor es de fácil implementación a nivel de software y hardware.

Tabla 1.6 Especificación técnica del sensor DHT11 [10].

Características	**Descripción**
Modelo	DHT11
Voltaje de alimentación	3-5 Vdc
Señal de salida	Señal digital a través de un solo bus
Elemento sensor	Componente resistivo de humedad y NTC para detectar temperatura
Rango de medición	Humedad (20-90% HR); Temperatura (0~50°C)
Precisión	Humedad (+-5% RH); Temperatura (+2°C)
Resolución o sensibilidad	Humedad (1%HR); Temperatura (1°C)
Repetibilidad	Humedad (+-1%HR); Temperatura (+-1°C))
Histéresis de humedad	+-1%RH
Estabilidad a largo plazo	+-1%RH/año
Periodo de adquisición de datos	Promedio: 6s
Intercambiabilidad	Totalmente intercambiable

1.4.3.2. Sensor de temperatura y humedad relativa (AM2302/DHT22)

El AM2302 es un sensor digital de temperatura y humedad relativa de buen rendimiento y bajo costo. Integra un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos (no posee salida analógica). Utilizado en aplicaciones de control automático de temperatura, aire acondicionado, monitoreo ambiental en agricultura y más [3].

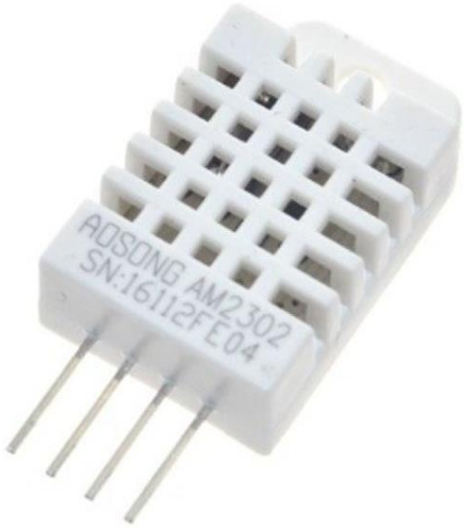

Figura 1.5 Sensor de Temperatura y Humedad AM2302/DHT22 [11].

Este sensor se puede usar en plataformas Arduino/Raspberry Pi/Nodemcu de forma sencilla, ya que a nivel de software disponemos de una amplia gama de librerías, mientras que a nivel de hardware solo en necesario conectar al pin VCC a un voltaje de alimentación de 3-6 Vdc y el pin GND a tierra, los datos se obtendrán cada 2 segundos, asegurando alta estabilidad y fiabilidad a largo plazo.

Tabla 1.7 Especificación técnica del sensor AM2303 [3].

Características	**Descripción**
Modelo	AM2303/DHT22
Voltaje de alimentación	3-6 Vdc
Señal de salida	Señal digital a través de un solo bus
Elemento sensor	Condensador de humedad de polímero y DS18B20 para detectar temperatura
Rango de medición	Humedad (0-100%HR); Temperatura (-40~125°C)
Precisión	Humedad (+-2%RH - Máx. +-5%RH); Temperatura (+-0.2°C)
Resolución o sensibilidad	Humedad (0.1%HR); Temperatura (0.1°C)
Repetibilidad	Humedad (+-1%HR); Temperatura (+-0.2°C))
Histéresis de humedad	+-0.3%RH
Estabilidad a largo plazo	+-0.5%RH/año
Periodo de adquisición de datos	Promedio: 2s
Intercambiabilidad	Totalmente intercambiable

1.4.4. Microcontroladores

El microcontrolador es un circuito integrado cuya característica es la de automatizar procesos y procesar información, es como una pequeña computadora que incluye puertos de entrada y salida para el control de diversos elementos, también posee un procesador y memoria en la cual se puede almacenar el programa y diversos datos.

1.4.4.1. Arduino

Arduino es una plataforma electrónica de código abierto basada en hardware y software de aplicación sencilla, mediante el envío de diversas instrucciones al microcontrolador ubicado en la placa base el sistema Arduino puede leer instrucciones (entradas) y transformarlas en acciones (salidas) [12].

Características generales del sistema Arduino:

- Costo relativamente bajo a comparación con otros sistemas de procesamiento similares.
- Software multiplataforma, disponible para sistemas operativos como Windows, Macintosh y Linux.
- Programación simple y sencilla de implementar, muy flexible para usuarios avanzados.
- Lenguaje de programación AVR C con software libre y extensible, permitiendo que los usuarios puedan agregar bibliotecas C++ que faciliten aún más la programación.
- Hardware extensible, permitiendo que los usuarios puedan crear sus distintos módulos de acuerdo con sus necesidades.

1.4.4.2. Arduino MKR WAN 1310

Arduino MKR WAN 1310 permite conectar diversos sensores y actuadores a muy largas distancias ya que se rige bajo el protocolo inalámbrico de LoRa o redes LoRaWAN.

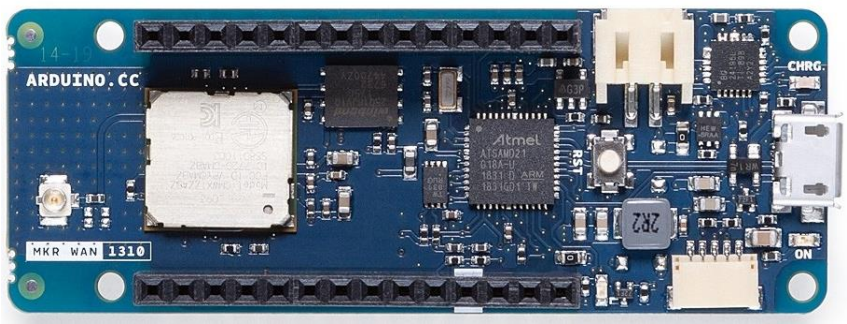

Figura 1.6 Tarjeta Arduino MKR WAN 1310 *[13]*.

Arduino MKR WAN 1310 posee un control mejorado del consumo de energía de la placa obteniendo una duración de batería bastante considerable, configurado correctamente el consumo de energía es de 104 uA, teniendo la posibilidad de alimentar a la tarjeta Arduino mediante el puerto USB (5 V) o con una batería individual [13].

Características generales del dispositivo:

- Placa de código abierto, permitiendo conectividad con:
    - Arduino crear.
    - Red LoRa propia usando Arduino Pro-Gateway para LoRa.
    - Infraestructura LoRa disponible, como “The Things Network”.
    - Placas que dispongan el modo conectividad directa.
- Registro de datos y funciones OTA (Over-the-Air).
- Seguridad al almacenar credenciales y certificados.

Todas estas características convierten a la tarjeta Arduino MKR WAN 1310 en el nodo de IoT perfecto y en el bloque de construcción para dispositivos de IoT de largo alcance y baja potencia [13].

Tabla 1.8 Especificación técnica de la placa Arduino MKR WAN 1310 [14].

Características	**Descripción**
Microcontrolador	SAMD21 Cortex®-M0+ 32bit low power ARM MCU
Modulo radio	CMWX1ZZABZ
Alimentación de la placa (USB/Vin)	5V
Baterías compatibles	Li-Ion recargable, o Li-Po, 1024 mAh capacidad mínima
Voltaje de operación del circuito	3.3V
Puertos digitales E/S	8
Puertos PWM	13 (0 ... 8, 10, 12, 18 / A3, 19 / A4)
UART	1
SPI	1
I2C	1
Puertos de entrada analógica	7 (ADC 8/10/12 bit)
Puertos de salida analógica	1 (DAC 10 bit)
Interrupciones externas	8 (0, 1, 4, 5, 6, 7, 8, 16 / A1, 17 / A2)
Corriente DC por puertos E/S	7 mA
CPU memoria flash	256 KB (interno)
GSPI memoria flash	2MByte (externo)
SRAM	32 KB
EEPROM	no
Velocidad de reloj	32.768 kHz (RTC), 48 MHz
USB	Dispositivo USB de alta velocidad y host integrado

Ganancia de antena	2dB (antena pentabanda)
Frecuencia de carga	433/868/915 MHz
Tamaño	67.64 mm x 25 mm
Peso	32 gr.

1.4.5. Internet de las cosas (IoT)

El internet de las cosas (IoT) por sus siglas en ingles es una red que abarca objetos físicos como dispositivos electrónicos, vehículos e incluso infraestructuras integradas con electrónica, sensores, circuitos, software y conectividad de red, permitiendo que estos objetos recopilen e intercambien información [15].

Las IoT permite que varios objetos se detecten y controlen de forma remota, mediante una red existente [15], esta constante evolución de la tecnología dio paso a un nuevo tipo de comunicación, la comunicación que normalmente conocemos es de tipo humano-humano o humano-maquina, con la inclusión de las IoT ahora existe la comunicación máquina-máquina [16].

El requisito fundamental de las IoT es la interconexión entre dispositivos, es por eso por lo que su arquitectura debe garantizar la operabilidad entre el mundo físico y el mundo virtual [15]. La arquitectura IoT involucra muchos factores como redes, comunicación, seguridad, modelos y más, también considerando que con el paso del tiempo las cosas se pueden mover de forma física o añadir nuevos dispositivos, al diseñar la arquitectura de IoT se debe tener en cuenta la extensibilidad, escalabilidad e interoperabilidad entre dispositivos.

1.4.6. LoRa

LoRa (abreviatura de largo alcance en inglés) es una técnica de modulación de espectro ensanchado proveniente de la tecnología CSS (abreviatura de chirp de espectro ensanchado en inglés), donde un chirp representa una señal senoidal de incremento o disminución de frecuencia con el paso del tiempo, mediante el barrido de frecuencia el CSS codifica la información usando pulsos modulados en frecuencia lineal de banda ancha [17]. LoRa es una plataforma inalámbrica de largo alcance y poco consumo de energía actualmente muy usado en aplicaciones de IoT enfocadas en agricultura, edificación, eficiencia de la infraestructura, cuidades inteligentes, logística y mucho más, con una red que abarca más de 170 países a nivel global [17]. Opera en bandas por debajo de los Giga Hertz, siendo que en América se usa una banda de 915 MHz y en Europa una banda de 868 MHz [18].

1.4.6.1. LoRaWAN

LoRaWAN es un protocolo de red de área amplia y baja potencia (LPWA) por sus siglas en inglés, diseñado para la conexión de dispositivos (alimentados por pequeñas baterías) a redes de internet tanto regionales, nacionales y globales de forma gratuita [19].

Para aplicaciones de IoT que requieran comunicaciones de largo alcance, bajo consumo de energía y una tasa de transmisión de datos baja a un bajo costo (la gran mayoría), el diseño de las redes LPWAN logran satisfacer estas necesidades a comparación de otras redes para aplicaciones de IoT.

Tabla 1.9 Comunicaciones inalámbricas par aplicaciones IoT [17].

Tipo de Redes	**Zigbee**	**Wi-Fi**	**Cellular**	**LPWAN**
Potencia	Baja	Alta	Alta	Baja
Tasa de datos	250 kbps	1 Gbps	7.2 Mbps	<100 kbps
Alcance	100 m	20 m	Rango del celular	>1 Km
Costo	Bajo	Medio	Alto	Bajo

Dentro de las redes LPWAN existen varias tecnologías además de LoRa, como los son SigFox y LTE-M, donde se destaca que SigFox es la que mayor alcance posee, mientras que LTE-M es la que mayor tasa de datos ofrece.

Tabla 1.10 Tecnologías LPWAN [17].

Tecnologías	**LoRa**	**SigFox**	**LTE-M**
Banda de frecuencia	Sin licencia (800-930 MHz)	Sin licencia (800-930 MHz)	Licenciada
Ancho de banda	500/250/125 kHz	100 Hz	20 MHz
Tasa de datos	50 kbps	100 bps	1 Mbps
Modulación	CSS	D-BPSK/GFSK	QPSK/16QAM
EIRP	14 dBm	20 dBm	23 dBm
Mensaje por día	Ilimitado	140	Ilimitado
Carga útil máxima	243 byte	12 byte	1600 byte
Alcance	15 km	50 km	11 km
Encriptación	AES 128b	Sin soporte	Encriptación LTE
Tiempo de vida	8.75 años	7.5 años	
Costo	$5	$5	$13

La tecnología LoRa esta equilibrada entre SigFox y LTE-M, por lo que ha ganado mucha popularidad en los últimos años.

LoRaWAN está implementado por una arquitectura de red tipo doble estrella lo que quiere decir que implementa una topología estrella en la infraestructura LoRa y otra topología estrella en la infraestructura de red.

La topología típica de LoRaWAN se puede ver en la Figura 1.7, donde se aprecia que los sensores se conectan con los diversos gateways en topología estrella, los gateways actúan como un conversor del protocolo de LoRa a Wi-Fi/Ethernet/3G/4G dependiendo el dispositivo [18], para posteriormente acceder a la red y conectarse a un servidor de aplicaciones.

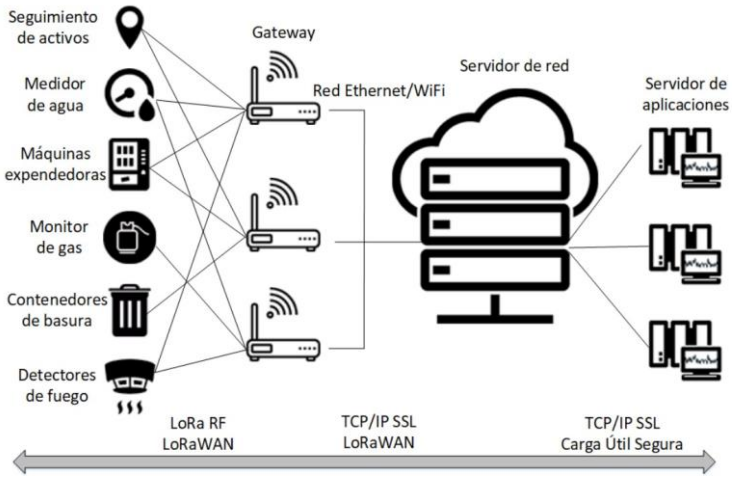

Figura 1.7 Arquitectura LoRaWAN de doble estrella [18].

LoRaWAN es un protocolo de red abierto que brinda una comunicación bidireccional de forma segura, se localiza en la capa MAC, mientras que en la capa física "PHY" se encuentra el servicio de LoRa desarrollado por Semtech, en esta capa se crea el enlace de comunicación de largo alcance [20].

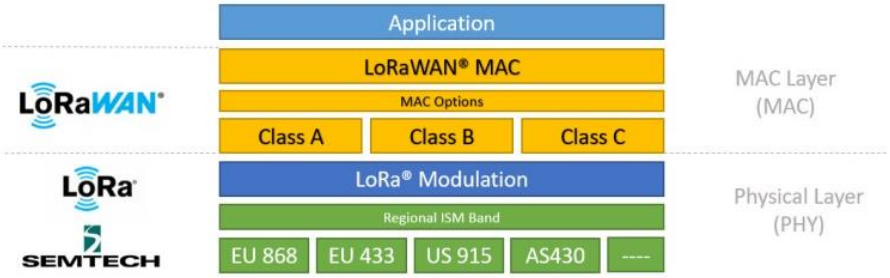

Figura 1.8 Distribución por capas de la tecnología LoRa [20].

LoRaWAN posee tres clases de dispositivos de punto final, cada uno acorde a las necesidades de las diferentes aplicaciones:

- Clase A: Dispositivos finales bidireccionales de menor potencia: Compatible con todos los dispositivos finales, asíncrono, menos consumo de energía con comunicación de enlace ascendente.

- Clase B: Dispositivos finales bidireccionales con latencia de enlace: Sincronizados a la red, que permite una comunicación de enlace descendente con latencia con un consumo de energía un poco mayo a la clase A.

- Clase C: Dispositivos finales bidireccionales con latencia baja y mayor potencia: Posee dos ventanas de enlace descendente lo que provoca una mayor latencia a coste de un consumo de energía elevado, siendo adecuada en aplicaciones con disponibilidad de energía constante.

2. METODOLOGÍA

En este capítulo se detallarán los componentes que serán utilizados en la implementación del sistema, configuración de los dispositivos, desarrollo y gestión del sistema de adquisición de datos.

2.1. Elementos Seleccionados

2.1.1. Microcontrolador

Uno de los requisitos importantes en el diseño del sistema es no requerir energía de forma continua, ya que en la industria de la agricultura muchas veces no se dispone de electricidad en los campos de cultivo y sus derivados, la tarjeta de procesamiento Arduino MKR WAN 1310 posee entre muchas de sus características, la disponibilidad de operar con el protocolo de LoRa y un consumo sumamente bajo de energía, por lo que es viable el uso de la tarjeta con baterías externas.

Arduino MKR WAN 1310 cuenta con un módulo radio CMWX1ZZABZ (tipo ABZ), como se observa en la Tabla 1.8, este módulo radio opera en frecuencias de 868 MHz y 915 MHz [21] esto quiere decir que se puede utilizar tanto en América como en Europa, además integra un chipset de radio Semtech SX1276 que se pueden encontrar en diversos microcontroladores con conectividad LoRa, pero lo que diferencia a la tarjeta Arduino MKR WAN 1310 es que adicionalmente cuenta en el mismo modulo radio un microcontrolador STM32L de ultra bajo consumo.

Con la adición del microcontrolador STM32L, el módulo radio tipo ABZ admite el protocolo inalámbrico LoRaWAN, debido a que lleva instalado un firmware en el microcontrolador embebido, permitiendo conectarse de forma más sencilla y rápida a LoRaWAN. Para otras tarjetas que no dispongan de esta característica, se pueden implementar dichas capas por software mediante el uso de librerías, incrementando la carga adicional de código.

Arduino MKR WAN 1310 posee soporte para baterías recargables Li-Ion o Li-Po, con capacidad mínima de 1024 mAh, a comparación con su versión anterior (MKR WAN 1300) que solo dispone la capacidad de implementar 2 baterías AA o AAA.

Tabla 2.1 Comparación de microcontroladores Arduino MKR.

Microcontroladores	Batería externa	Memoria SRAM	Costo
Arduino MKR WAN 1300	x2 AA, AAA	32 KB	$48.80
Arduino MKR WAN 1310	Li-Ion/Li-Po recargables	32 KB	$46.00

Para reforzar la seguridad de la transmisión de datos, Arduino MKR WAN 1310 contiene un co-procesador criptográfico ATECC508 (no disponible en modelos anteriores), este elemento

permite [22]:

- Protección de nodos de red IoT, verificando la autenticidad de las ID de los nodos, con posibilidad de crear claves de sesión para el cifrado de mensajes.
- Anti-falsificación, validando la autenticidad de un cliente externo, los cuales pueden ser tarjetas secundarias electrónicas u otras piezas de repuesto.
- Protección de firmware, validando el código interno de la memoria flash, evitando falsificaciones o modificaciones.
- Almacenamiento de datos seguros.
- Comprobación de la contraseña de usuario, validando las contraseñas ingresadas por usuarios e intercambiando valores de contraseñas de forma segura con sistemas remotos.

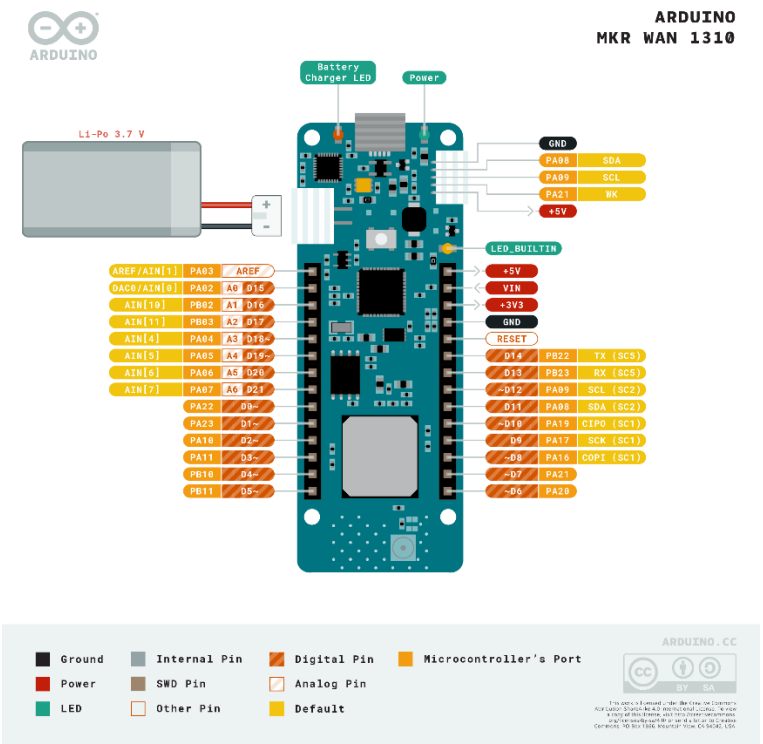

Figura 2.1 Diagrama de distribución de pines Arduino MKR 1310 [14].

El microcontrolador Arduino MKR WAN 1310 cuenta con 8 pines digitales de entrada y salida, 7 pines de entrada analógica, un pin de salida analógica, un puerto tipo S2B-PH-SM4-TB para la conexión de la batería Li-Po de 3.7 [V], una antena con un conector micro UFL y un puerto I2C.

2.1.2. Sensor

El sensor AM2302 se utilizará en el diseño del proyecto, como el único sensor principal, ya que puede tomar datos tanto de humedad como de temperatura, el sensor cubre perfectamente los rangos de humedad y temperatura que se encuentran en el proceso de secado de café al sol.

El sensor puede fallar en ambientes hostiles si y solo si es sometido a temperaturas de entre 50 y 60°C y una humedad <10% HR durante 2 horas, inmediatamente sometido a temperaturas de entre 20 y 30°C y una humedad >70% HR durante 5 horas [23].

Puesto que el sensor debe ser colocado dentro del secador solar parabólico, no existe problema alguno con respecto a daños en el dispositivo ya que el ambiente no es hostil, ni sobrepasa la temperatura de operación recomendada en el uso del sensor.

Figura 2.2 Sensor AM2302 [24].

2.1.3. Batería

La batería que será usada en el diseño del producto es una batería UBP103450/PCM Li-Ion recargable de 3.7 V y una capacidad de 1750 mAh que cumple con los requisitos solicitados por la tarjeta Arduino MKR WAN 1310 como se observa en la Tabla 1.8.

La batería posee gran capacidad de almacenamiento y junto al pequeño consumo de energía del sistema, obtenemos un tiempo de uso bastante largo, las características principales de esta batería son:

Tabla 2.2 Especificación técnica de la batería UBP103450/PCM [25].

Características	**Descripción**
Rango de voltaje	3.0 a 4.2 V
Voltaje promedio	3.7 V
Capacidad	1750 mAh
Peso	41 g
Temperatura de funcionamiento	-20°C a 60°C

Temperatura de almacenamiento	-20°C a 45°C
Autodescarga	< 10% por mes
Tamaño	54x36x11 mm

La tasa de carga recomendada es de 875 mA a 4,2 V en un rango de temperatura de 5 °C a 45 °C. Se debe mantener a 4,2 V hasta que la corriente disminuya a 175 mA. La tasa de carga máxima es de 1,75 A a 20 °C±5 °C [25].

Figura 2.3 Batería UBP103450/PCM Li-Po recargable [25].

Adicionalmente a la batería, se debe adquirir un puerto JST SH para conectarlo a la placa Arduino.

2.2. Diseño y programación del sistema de adquisición de datos

El funcionamiento del sistema de adquisición de datos es relativamente sencillo, primero tenemos que construir nuestro nodo final que consta de nuestra placa Arduino previamente configurada e instalada con los sensores correspondientes a la aplicación, seguido mediante LoRaWAN la información de dichos sensores será enviada hacia un gateway, es posible usar algún gateway que se encuentre en el rango de comunicación (15-20 km) si no se dispone de un gateway cercano, se puede lo puede comprar, ya que se encuentran disponibles varios módulos gateway LoRaWAN en el mercado, o si se desea es posible construir uno.

El gateway recogerá la información y mediante una conversión la enviará por tecnología Wi-Fi/Ethernet/3G/4G a un servidor en la nube, existen muchos servidores gratuitos que pueden ser usados, finalmente podremos ver toda la información desde alguna computadora, tablet o celular con conexión a internet. La ruta del sistema de adquisición de datos se puede ver en la figura 12.

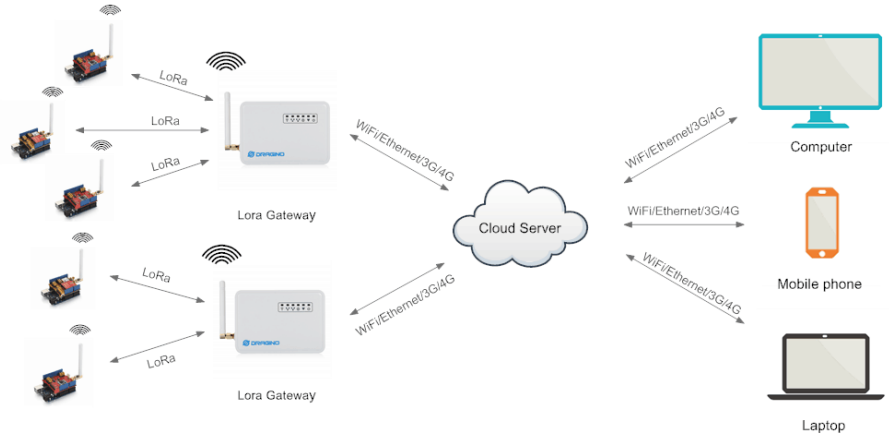

Figura 2.4 Ruta de comunicación LoRaWAN [26].

2.2.1. Diseño del nodo final con Arduino

2.2.1.1. Configuración del sistema

2.2.1.1.1. Arduino IDE

Es la plataforma de Arduino la cual facilita la escritura y la carga del código en las distintas placas, el software se puede usar para todas las placas Arduino del mercado [27].

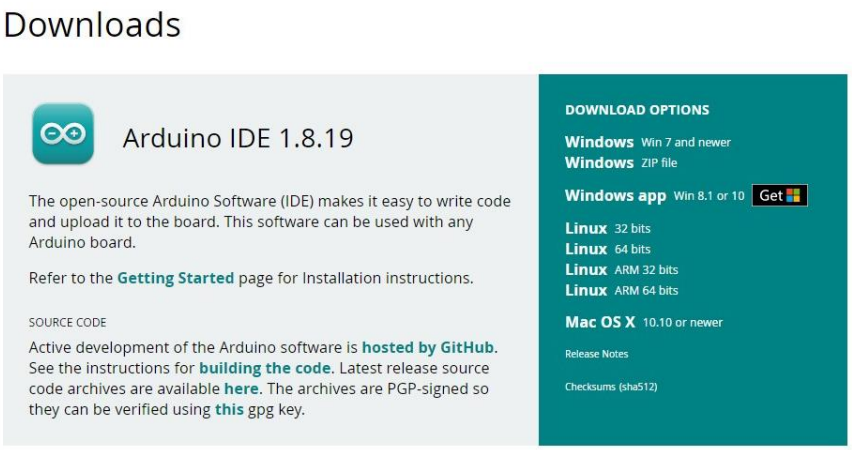

Figura 2.5 Sitio de descarga del software Arduino IDE [27].

Primero se debe descargar e instalar el software de Arduino IDE, como podemos observar en la Figura 2.5 este software está disponible para varios sistemas operativos y sus diversas versiones, ya sea Windows, Linux y Mac OS X. La instalación de Arduino IDE es muy sencilla e intuitiva.

Una vez instalado el software, lo ejecutamos y veremos la siguiente ventana:

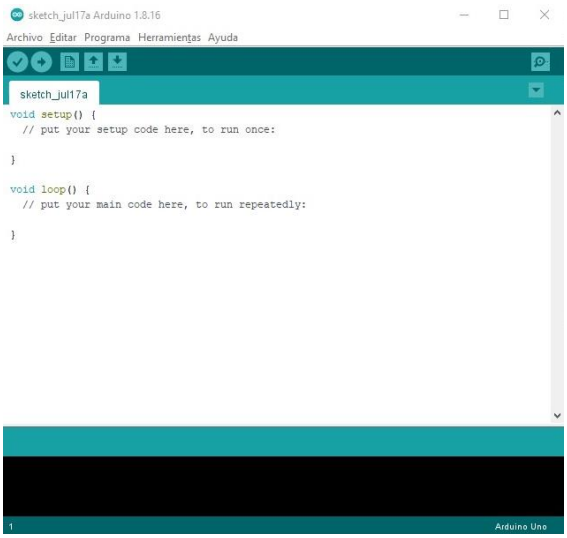

Figura 2.6 Ventana de inicio Arduino IDE.

2.2.1.1.2. Instalación de la tarjeta Arduino MKR WAN 1310

Posteriormente lo primero que debemos realizar es instalar nuestra tarjeta Arduino, para este sistema será la tarjeta Arduino MKR WAN 1310.

Para esto nos dirigimos a la barra de herramientas y seleccionamos las opciones: Herramientas >> Placa >> Gestor de tarjetas.

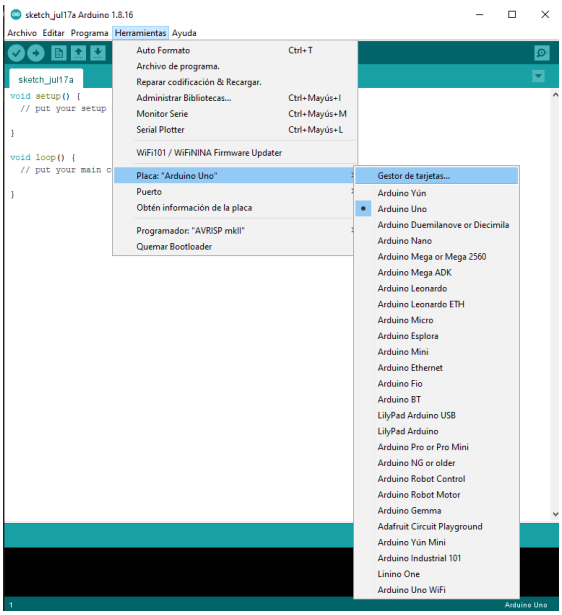

Figura 2.7 Instalación de la tarjeta Arduino MKR WAN 1310 en Arduino IDE.

Se abrirá una nueva pestaña en la cual debemos colocar las palabras claves “MKR WAN” y nos brindará una opción de tarjeta, verificamos si la versión es la que deseamos instalar y procedemos a dar clic en “Instalar”.

Figura 2.8 Selección tarjeta MKR WAN.

2.2.1.2. Lectura del sensor AM2302/DHT22

Una vez instalada la tarjeta correctamente, procedemos a crear el código de lectura para el sensor AM2302/DHT22.

Primero debemos instalar la librería del sensor, para esto nos dirigimos a la barra de herramientas y seleccionamos las opciones: Programa >> Incluir Librería >> Administrar Bibliotecas.

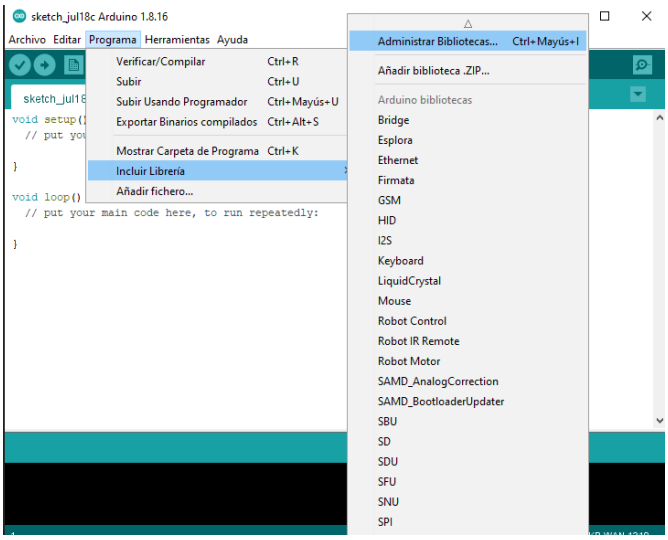

Figura 2.9 Adición librería DHT.

Se abrirá una nueva pestaña en la cual debemos colocar la palabra clave “DHT” para filtrar la librería que deseamos, escogemos la librería del creador Adafruit, ya que esta contiene al sensor DHT22.

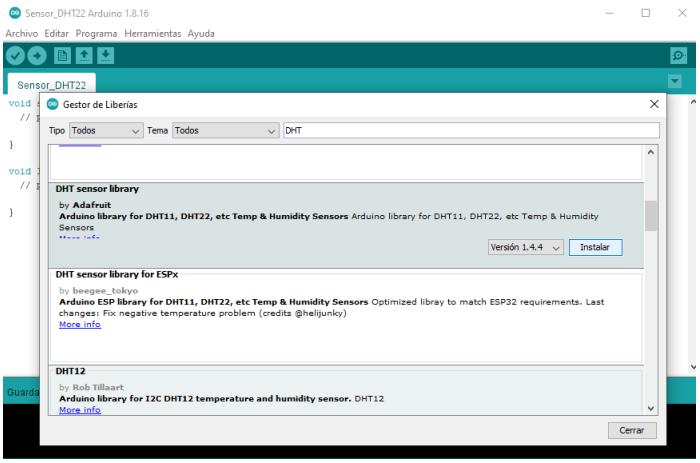

Figura 2.10 Instalación librería DHT.

Al instalar nos mostrara un mensaje que nos indica que la librería DHT necesita de la instalación de otras dependencias para su funcionamiento, por lo que damos clic en “instalar todas”.

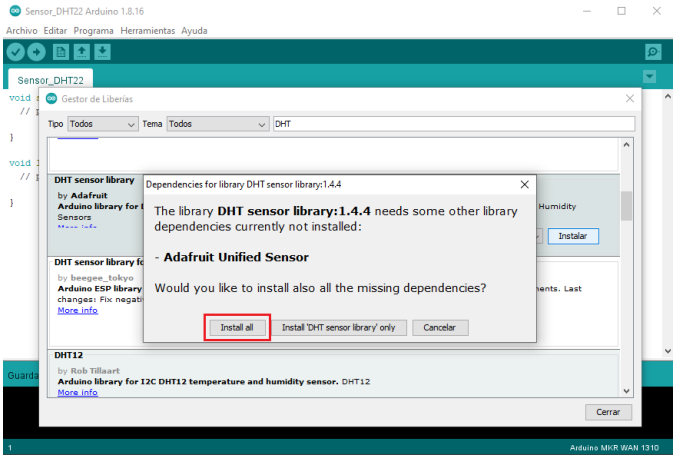

Figura 2.11 Instalación de dependencias de la librería DHT.

Una vez instalada la librería procedemos a obtener los datos del sensor mediante el siguiente código:

Figura 2.12 Fragmento 1 código configuración del sensor DHT.

Incluimos la librería DHT previamente instalada, definimos el tipo de sensor que estamos usando, en este caso el sensor DHT22, definimos el pin de entrada digital donde será conectado el pin de datos del sensor en nuestro caso definimos el pin 7 (puede ser cualquier

pin digital disponible), configuramos la serialización a 9600 baudios e imprimimos un mensaje de inicio.

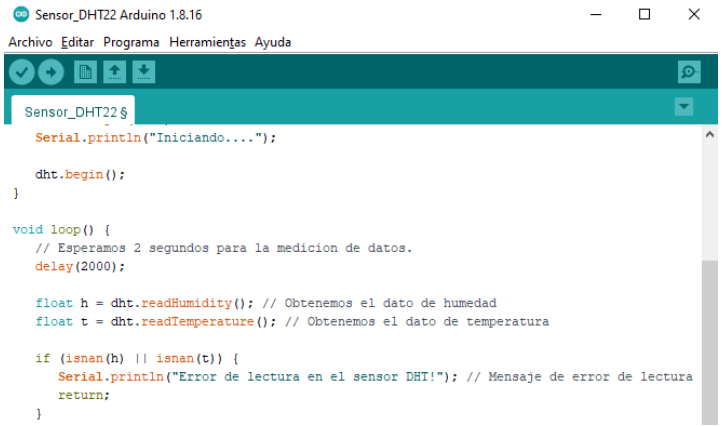

Figura 2.13 Fragmentos 2 código lectura del sensor DHT.

Una vez configurado y determinado los puertos del sensor, inicializamos el mismo. Dentro del lazo principal “void” colocamos un retraso de 2 segundos (mínimo), esto debido a la limitación del sensor.

El funcionamiento del sensor es relativamente sencillo, primero el microcontrolador envía una señal de inicio, el sensor recibe la señal y cambia del estado espera al estado de ejecución, cuando el microcontrolador termine de enviar la señal de inicio el sensor enviara una señal de respuesta de datos de 40 bits, donde se encuentran 16 bits para el dato de humedad, 16 bits para el dato de temperatura y 8 bits de confirmación. El sensor vuelve al estado de espera una vez finalizado la recopilación de datos y cuando el microcontrolador no envía una señal de inicio [23]. Todo este proceso debe durar como mínimo un intervalo 2 segundos, si el intervalo de lectura es menor a 2 segundos la comunicación podría no tener éxito.

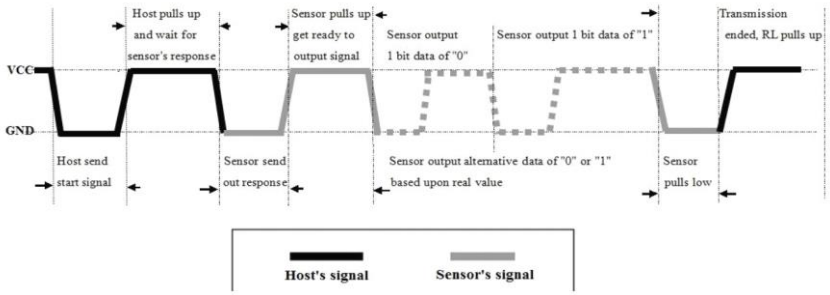

Figura 2.14 Tiempo de comunicación del sensor AM2302/DHT22 [23].

Después de colocar un retraso de 2 segundo escribimos los comandos “readHumidity” y “readTemperature” que nos brinda la librería DHT como se observa en la Figura 2.13, obteniendo los datos de humedad y temperatura. Si existe algún error de lectura, se desplegará un mensaje de error. El tiempo de lectura de datos puede cambiarse de acuerdo con las necesidades del usuario, siendo el tiempo mínimo 2 segundos por las características

del sensor explicas anteriormente.

Si deseamos ver los datos impresos en el monitor serie de Arduino IDE, debemos incluir las siguientes líneas de código dentro del lazo principal.

```
    Serial.print("Humedad: "); //Imprimimos el dato de humedad
    Serial.print(h);
    Serial.print(" %\t");
    Serial.print("Temperatura: "); //Imprimimos el dato de temperatura
    Serial.print(t);
    Serial.print(" *C ");
}
```

Figura 2.15 Fragmento 3 código visualización de datos.

El código completo de la adquisición de datos se encuentra en el ANEXO I.

2.2.1.3. Conexión del nodo final

El esquema de conexión es muy simple, se requiere del sensor AM2302/DHT22, la tarjeta Arduino MKR WAN 1310, una resistencia de pull up con un valor de entre 4.7-10 kΩ y cable de conexión.

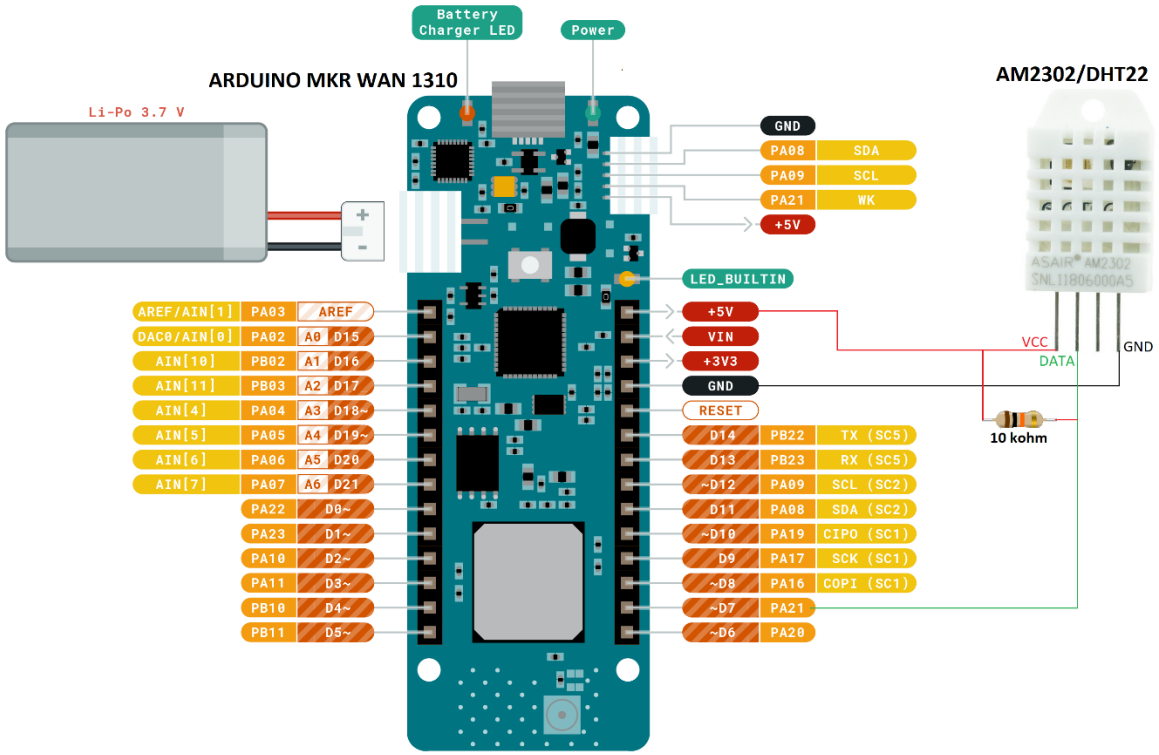

Figura 2.16 Esquema de conexión nodo final.

2.2.2. Enlace del nodo final a un servidor mediante LoRaWAN

Con el nodo final correctamente configurado, procedemos a crear el código para la comunicación hacia un servidor usando la tecnología LoRaWAN, para esto primero debemos instalar la librería “MKRWAN” en el software de Arduino IDE, nos dirigimos a la barra de herramientas y seleccionamos las opciones: Programa >> Incluir Librería >> Administrar

Bibliotecas.

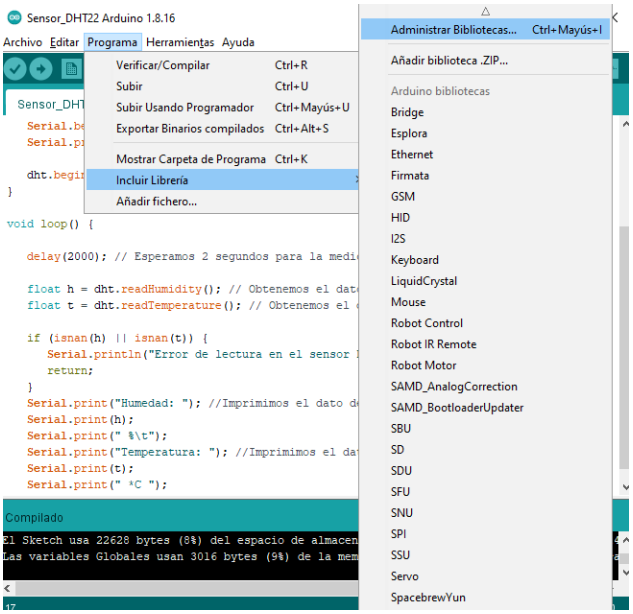

Figura 2.17 Adición librería MKRWAN.

En la nueva pestaña de “Gestor de Librerías” colocamos las palabras claves “MKR WAN” para filtrar la librería que deseamos, escogemos la librería MKRWAN desarrollada por Arduino y la instalamos.

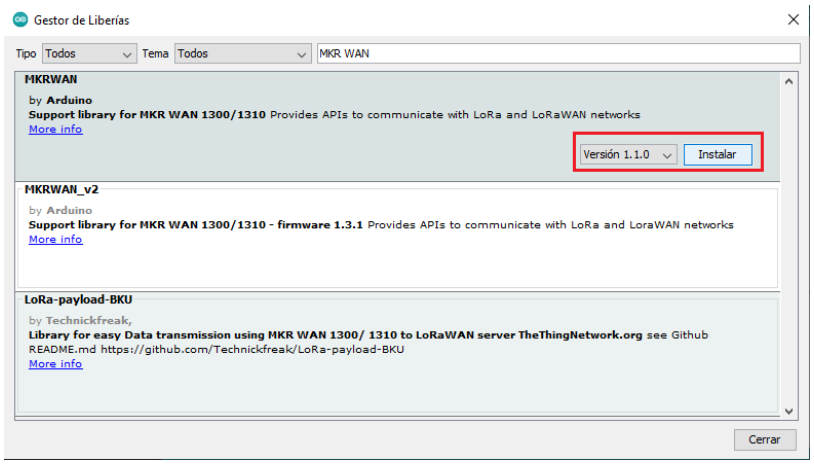

Figura 2.18 Instalación librería MKRWAN.

2.2.2.1. Servidor de red

El servidor de red es el siguiente eslabón para la comunicación por medio de LoRa, existen muchos servidores de red con conectividad LoRa disponibles, para nuestro sistema de adquisición de datos se seleccionó el servidor de red “The Things Network” (TTN por sus siglas en inglés)

2.2.2.1.1. The Things Network (TTN)

The Things Network es una red global y abierta que proporciona un conjunto de herramientas

para el desarrollo de aplicaciones IoT, esta red global va creciendo a base de sus usuarios, mismos que aportan sus dispositivos gateway a la red TTN, para que diversos nodos finales puedan conectarse a ellos y posteriormente establecer comunicación a través de LoRa con la red.

Figura 2.19 Gateways a nivel global de la red TTN [28].

Actualmente en 2022, TTN posee más de veinte mil gateways en 151 países. Concretamente en Ecuador existen menos de 5 gateways activos por lo que para poder usar la red TTN es necesario adquirir un dispositivo gateway o encontrarse dentro del rango de uno de los gateway disponibles.

2.2.2.2. Registro al servidor de red TTN

Debemos ingresar a la página web de The Things Network y nos registramos de forma totalmente gratuita, creando un nuevo usuario y contraseña.

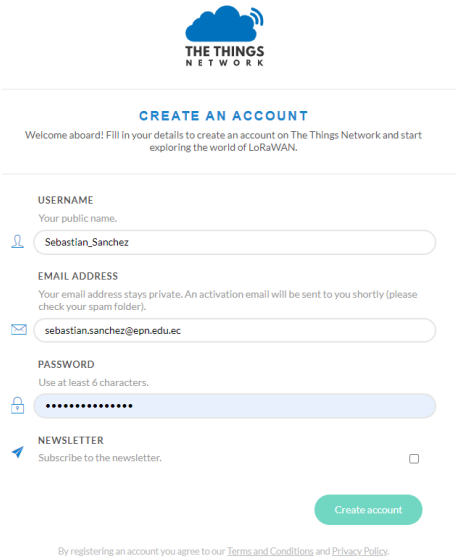

Figura 2.20 Registro a TTN [29].

Una vez registrados nos dirigimos a la pestaña consola en la parte superior de la ventana.

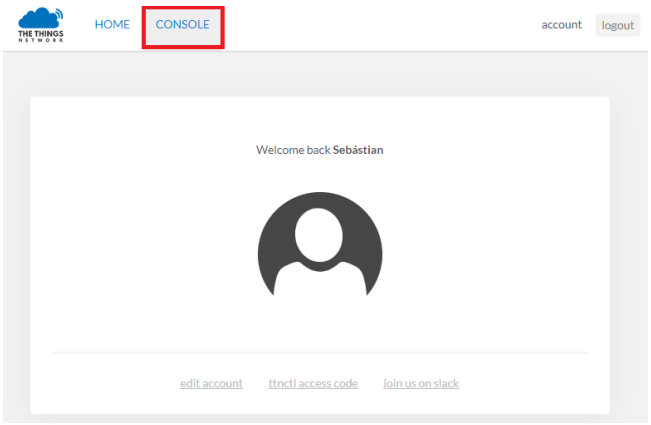

Figura 2.21 Consola TTN.

Se desplegará una nueva ventana donde tenemos que seleccionar la región en la que nos encontramos, por el momento solo está disponible Europa, Norte América y Australia. Para nuestro caso seleccionamos la región más cercana “Norte América”.

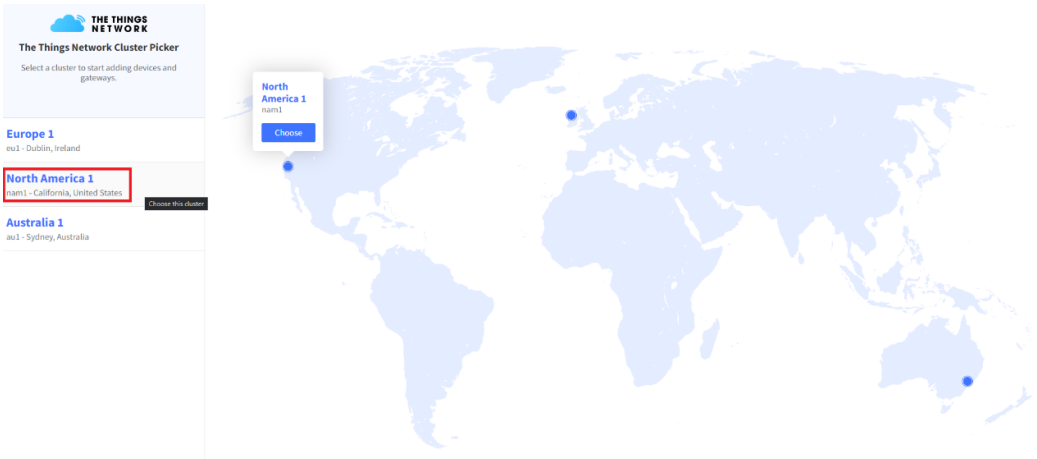

Figura 2.22 Seleccionar región TTN.

Luego nos redirige a la página de The Things Stack, donde nos pide iniciar sesión, aquí ingresamos el usuario o correo y la contraseña registrada anteriormente en TTN.

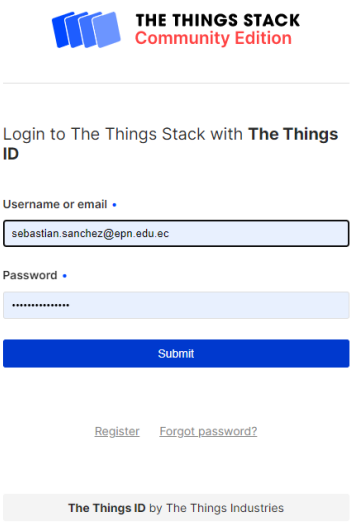

Figura 2.23 Inicio de sesión en TTS

2.2.2.3. Creación de una nueva aplicación en TTN

Una vez iniciada sesión, entramos a la ventana de la consola en nuestro perfil en TTN, teniendo dos opciones a elegir: crear una aplicación y registrar un gateway. En este momento vamos a crear una nueva aplicación y a conectar nuestro nodo final a la red TTN, así que seleccionamos dicha opción.

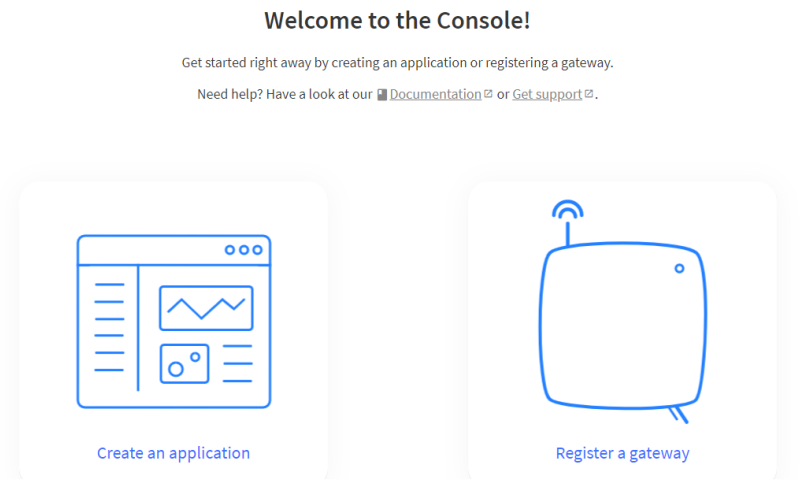

Figura 2.24 Consola TTN.

Al seleccionar la opción de crear una aplicación se abrirá una nueva ventana, donde tenemos que rellenar los siguientes campos: ID de la aplicación, nombre de la aplicación y descripción de la aplicación. Después damos clic en crear aplicación.

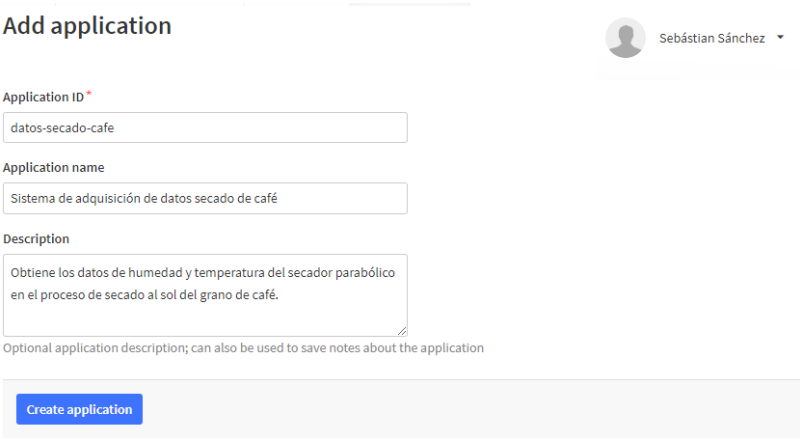

Figura 2.25 Crear nueva aplicación en TTN.

2.2.2.4. Registro del nodo final a la aplicación en TTN

Una vez creada la aplicación, debemos enlazar nuestro nodo final a la misma, para esto damos clic en la en botón “añadir dispositivo” que se encuentra dentro de nuestra aplicación creada.

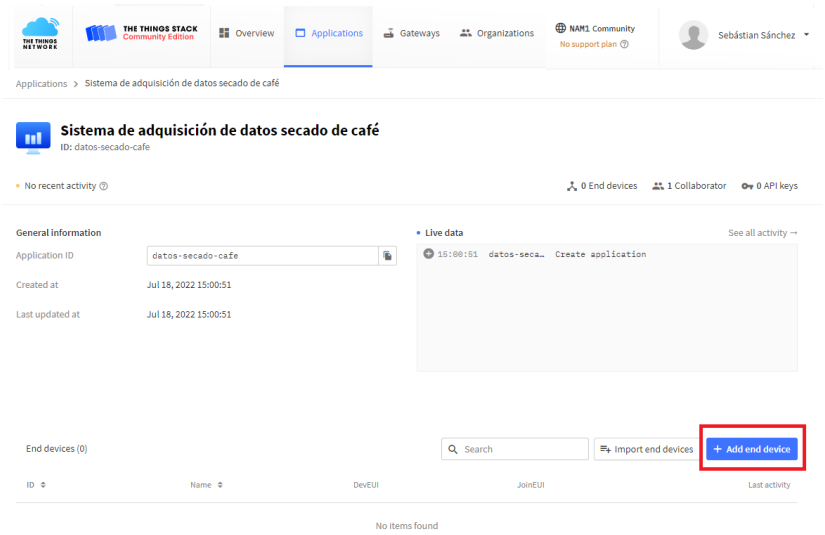

Figura 2.26 Añadir dispositivo a la aplicación.

En la nueva ventana desplegada debemos rellenar los campos solicitados que son:

- Fabricante del dispositivo: para nuestro sistema el fabricante es Arduino.
- Modelo del dispositivo: el modelo seleccionado es la tarjeta MKR WAN 1310.
- Versión de Hardware y Firmware: las versiones usadas son las más actuales.
- Perfil de región: La región seleccionada es América.

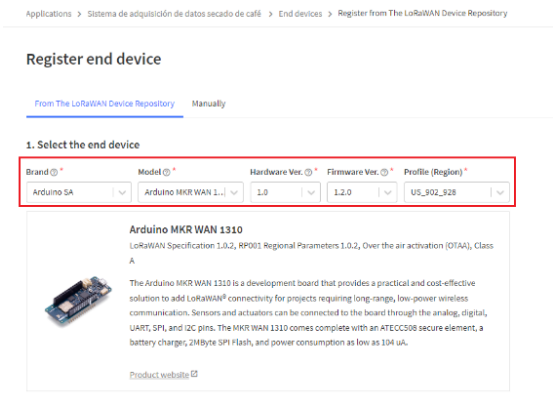

Figura 2.27 Registro del nodo final.

Una vez TTN identifica la tarjeta seleccionada, se requiere llenar los datos de registro de esta, donde:

- Plan de frecuencia: se escoge la frecuencia de América (usada por TTN).
- AppEUI: es un valor que se usa para identificar el servidor durante la activación del nodo, este valor se obtiene por el fabricante del dispositivo final o del dueño del servidor al que se desea conectarse. Si no se obtiene este valor, se puede rellenar este campo con ceros, teniendo en cuenta que se debe usar el mismo número cada

vez que se ingrese al servidor TTN.

- DevEUI: es un numero único de cada dispositivo (no se debe compartir) que se encuentra en la memoria flash de este, para poder extraer este número del nodo final se debe programar unas pequeñas líneas de código.

- AppKey: es generada automáticamente por TTN y sirve para poder enlazar la comunicación entre el nodo final y TTN por medio de LoRaWAN.

- End Device ID: es el nombre que asignamos a nuestro nodo final.

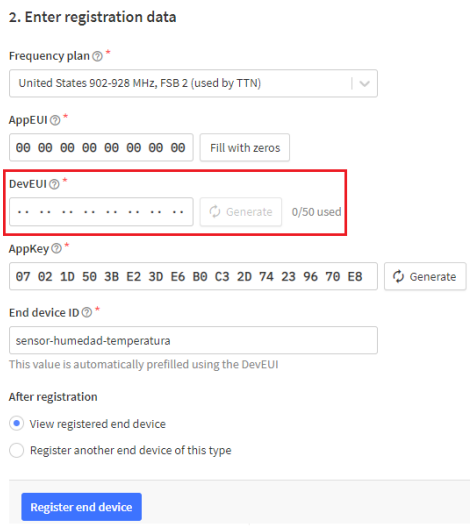

Figura 2.28 Datos de registro nodo final.

Cuando se haya obtenido el DevEUI procedemos a registrar el dispositivo, el cual se mostrará en la sección de dispositivos finales en la página de TTN.

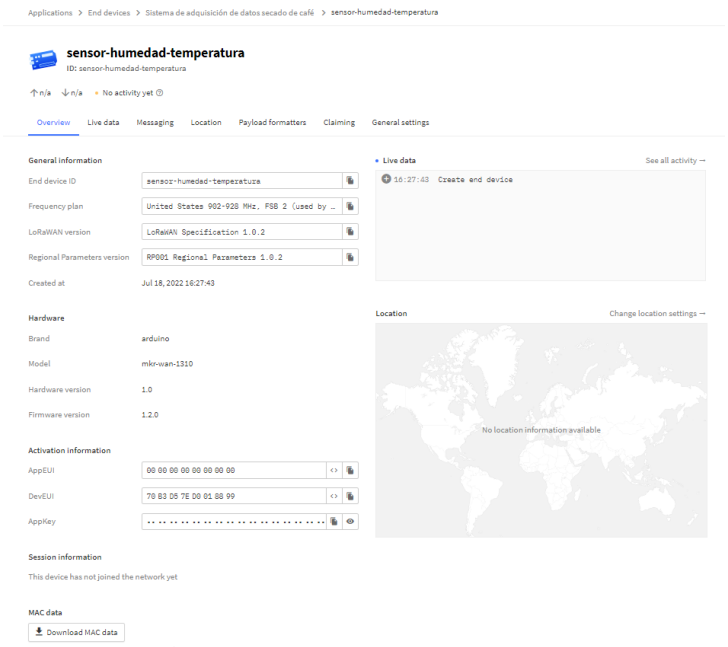

Figura 2.29 Nodo final correctamente registrado en TTN.

En esta ventana podremos observar los distintos parámetros de configuración, así como el numero DevEUI y el numero AppKey, valores muy importantes para la comunicación de nodo final con la red TTN, estos valores no deben ser compartidos con nadie.

A la derecha de la figura 37 podemos ver un mapa de localización, si clicamos en el mapa, podemos configurar la ubicación exacta de donde se encuentra instalado el nodo final.

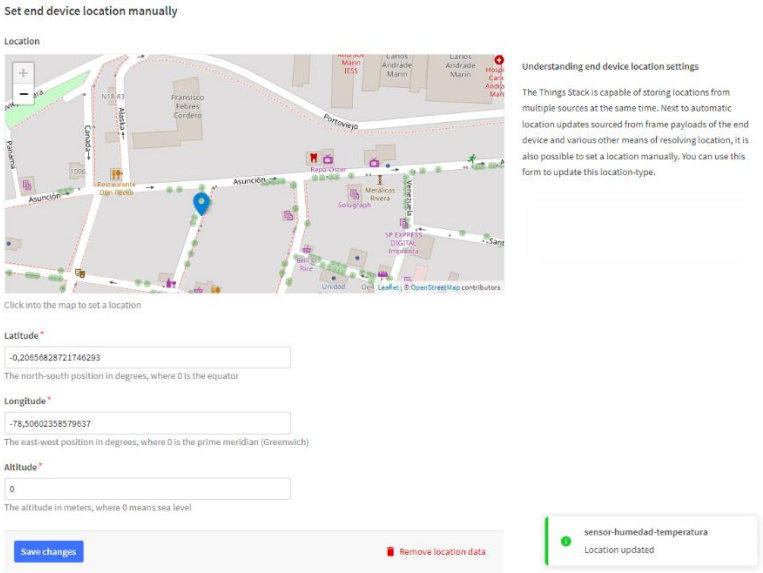

Figura 2.30 Configuración de localización del nodo final en TTN.

2.2.2.4.1. Obtención del número EUI del nodo final

Para obtener el numero EUI de nuestro nodo final Arduino, usamos un comando disponible en la librería MKR de Arduino instalada previamente.

Figura 2.31 Código para obtener el numero EUI de la tarjeta Arduino MKRWAN 1310.

Primero seleccionamos la región en la que nos encontramos, habilitamos un puerto serial para el módulo LoRa e inicializamos el módulo, si el módulo no inicia correctamente, se muestra un

mensaje de error, caso contrario imprime el numero EUI en el monitor serie de Arduino IDE. El código completo para obtener el numero EUI se encuentra en el ANEXO II.

2.2.2.5. Programación del nodo final para comunicarse con TTN

2.2.2.5.1. Archivo de cabecera con las credenciales EUI y AppKey

Primero debemos crear un archivo de cabecera que será incluido en el código, este archivo de cabecera debe incluir los números DevEUI y AppKey.

Los archivos de cabecera que pueden ser leídos por el software Arduino IDE son de tipo “C/C++ Header File” y pueden ser guardados por su abreviatura “.h”.

Para crear un nuevo archivo de cabecera debemos crear un nuevo bloc de notas, donde declaramos en forma de código las variables dándole un nombre con el que sea fácil reconocerlas, y les asignamos los valores que obtuvimos en la consola de TTN en la Figura 2.29.

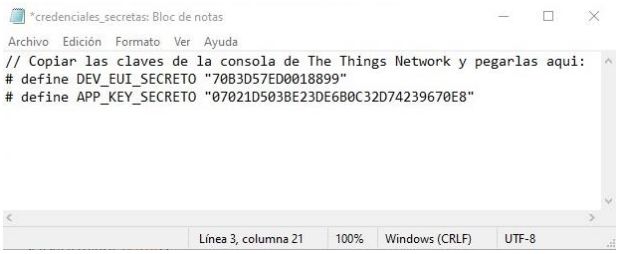

Figura 2.32 Creación del archivo de cabecera.

Luego guardamos el archivo, asignándole un nombre sin espacios ni caracteres especiales seguido de “.h” que es el formato de archivo de cabecera.

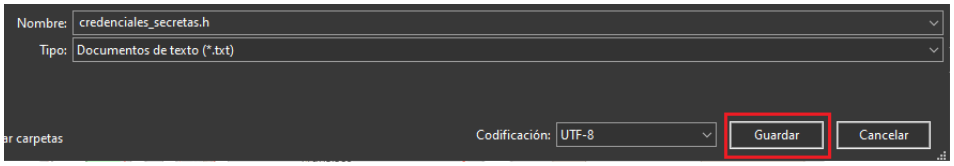

Figura 2.33 Guardar credenciales como archivo de cabecera.

El nuevo archivo de cabecera se debe encontrar en la carpeta del código principal desarrollado para poder ser usado.

2.2.2.5.2. Desarrollo del código para comunicarse con TTN

El código completo para la comunicación del nodo final con la red The Things Network se encuentra en el ANEXO III.

2.2.2.5.2.1. Configuración de parámetros iniciales

Con el archivo de cabecera creado nos dirigimos al software de Arduino IDE y declaramos todas las librerías que van a ser usadas en la programación.

```
#include <MKRWAN.h>       // Libreria MKR de Arduino
#include <avr/dtostrf.h>  // Libreria para la conversion float a cadena de caracteres
#include <stdlib.h>       // Libreria para la conversion float a cadena de caracteres
#include "DHT.h"          // Libreria del sensor DHT22
#define DHTTYPE DHT22     // Definimos el tipo de sensor / DHT22  (AM2302), AM2321
#include "credenciales_secretas.h" // Archivo de cabecera con las credenciales de TTN
String appEui = DEV_EUI_SECRETO;  // Asignamos las claves a dos objetos de tipo String
String appKey = APP_KEY_SECRETO;  // que se encuentran en el archivo de cabecera

const int DHTPin = 7;     // Definimos el pin digital al que va conectado el sensor

DHT dht(DHTPin, DHTTYPE); // Instancia a la clase DHT, creando el objeto dht
LoRaModem modem; // Inatancia a la clase LoRaModem, creando el objeto modem
```

Figura 2.34 Fragmento 1 código inclusión de librerías.

Además de las librerías añadidas anteriormente adicionamos las librerías básicas “stdlib” y “avr/dtostrf” las cuales nos ayudan con la conversión de datos tipo flotante a cadena de caracteres. Adicionalmente incluimos el archivo de cabecera con las credenciales y las asignamos a dos objetos de tipo “String” ya que es el formato de lectura solicitado en la librería MKRWAN.

```
void setup() {

  Serial.begin(9600); // Inicializa el puerto serie a 9600 baudios:

  if (!modem.begin(US915)) { // Cambiamos la banda a nuestra region(AS923, AU915, EU868, KR920, IN865, US915, US915_HYBRID)
  Serial.println("Error al inicializar el modulo"); // Mensaje de error con el modulo
  while (1) {}
  };
  Serial.print("Version de tu modulo: "); // Imprimimos la vercion y el DevEUI del nodo final
  Serial.println(modem.version());        // si no existe problemas con la banda de la region
  Serial.print("Tu device EUI es: ");
  Serial.println(modem.deviceEUI());

  int connected = modem.joinOTAA(appEui, appKey); // Nos conectamos con las credenciales a TTN
  if (!connected) {
    Serial.println("Ocurrio un error en la conexion a TTN"); // Mensaje de error cuando las credenciales fallan
    while (1) {}
  }
  dht.begin(); // Iniciamos el sensor
  modem.minPollInterval(60); // Establece el intervalo de sondeo a 60 secs.

}
```

Figura 2.35 Fragmento 2 código configuración general.

Dentro de la función “void” de configuración inicializamos el puerto serie a 9600 baudios, iniciamos el modem LoRa asignándole un ancho de banda de acuerdo con la región en la que nos encontremos. Si el modem no se inicia correctamente, un mensaje de error es mostrado en el monitor serie de Arduino IDE, caso contrario se muestra la versión del módulo LoRa y el numero EUI de la tarjeta MKR WAN 1310.

Una vez el módulo ha iniciado, se conecta a la red TTN con las credenciales guardadas en el archivo de cabecera, si existe algún error con las credenciales, se muestra un mensaje de error, caso contrario inicializa el sensor AM2302/DHT22 y establece un intervalo de sondeo de datos de 1 minuto.

2.2.2.5.2.2. Envío de los datos del sensor a la red TTN

Con toda la configuración inicial programada, es posible enviar los datos adquiridos por el sensor hacia la red de The Things Network.

Los datos del sensor se obtienen como se pudo ver en la Figura 2.13, estos datos deben ser transformados a un formato hexadecimal, ya que es la forma más optimizada posible para enviarlos a la red TTN, recordando que la tecnología LoRaWAN tiene un ancho de banda muy

pequeño debido al bajo consumo energético y alto alcance de envío.

Primero convertimos los datos del sensor (guardados en una variable tipo float) a cadena de caracteres con ayuda de comandos encontrados en la librería “avr/dtostrf”.

```
char humistring[20]; //Convierte las variable t y h de tipo float a tipo char array de caracteres
dtostrf(h,3,1,humistring); //el 3 es igual al número de digitos totales y el 1 el numero de decimales
char tempstring[20];
dtostrf(t,3,1,tempstring);

//Creamos el objeto humedad y temperatura de la clase String y la pasamos
//al array humistring y tempstring como constructor.
String humedad(humistring);
String temperatura(tempstring);
```

Figura 2.36 Fragmento 3 código conversión de los datos del sensor a cadena de caracteres.

La librería MKR convierte automáticamente los datos tipo “String” a hexadecimal, para posteriormente ser enviados a la red TTN. Para una visualización más dinámica se creó un pequeño lazo “for” para poder ver los datos de forma normal y su conversión a hexadecimal en el monitor serie de Arduino IDE.

```
Serial.print("Convirtiendo Humedad y Temperatura a hexadecimal... ");
Serial.println();
//Muestra en la terminal los bytes de la humedad en hexadecimal que se van a enviar
Serial.print("Enviando Humedad: " + humedad + " - ");
for (unsigned int i = 0; i < humedad.length(); i++) {
  Serial.print(humedad[i] >> 4, HEX);
  Serial.print(humedad[i] & 0xF, HEX);
  Serial.print(" ");
}
  Serial.println();
```

Figura 2.37 Fragmento 4 código conversión del dato de humedad a hexadecimal.

Un lazo similar se realizó para el dato de temperatura.

Cuando los datos del sensor han sido convertidos a cadena de caracteres, es factible enviarlos a la red TTN, para eso hacemos uso de los comandos de envío de la librería MKR.

```
int err; // Variable declarada para errores producidos en la transmisión

modem.beginPacket(); //Inicia el envío de paquetes

modem.print(humedad); // Envío la humedad a TTN
modem.print(temperatura); // Envío la temperatura a TTN
```

Figura 2.38 Fragmento 5 código envío de datos a la red TTN.

Inicializamos el envío de paquetes y con el comando “modem.print” enviamos el dato a la red TTN.

En la Figura 2.38 se puede ver que se envía a la red TTN tanto la humedad como la temperatura, pero se envía por separado, otra opción es enviar dichos datos concatenados ya que es otro formato admisible en la red TTN.

```
//  Paquete de datos unidos:
String datos_unidos(humistring); // String usado para la concatenacion de datos
datos_unidos.concat(temperatura); // Concateno los datos de humedad y temperatura
modem.print(datos_unidos);  // Envio el dato unificado
////////////////////////////////////////////////////////////////////////////////
```

Figura 2.39 Fragmento 6 código envío de datos concatenados a la red TTN.

2.2.2.5.2.3. Recepción de datos de la red TTN al nodo final.

La comunicación por tecnología LoRaWAN es bidireccional por lo que también podemos recibir paquetes de datos desde la red hasta nuestro nodo final.

```
delay(1000); // Espera de 1 segundo para recibir un mensaje desde la pagina de TTN al nodo final
if (!modem.available()) {
  Serial.println("No se recibió ningún mensaje de enlace descendente (downlink) en este momento.");
  return;
}
char rcv[64]; // Variable para almecenar el mensaje de TTN
int i = 0;
while (modem.available()) {
  rcv[i++] = (char)modem.read();
}
Serial.print("Recibido: "); // Imprimo en la terminal el mensaje de TTN
for (unsigned int j = 0; j < i; j++) {
  Serial.print(rcv[j] >> 4, HEX);
  Serial.print(rcv[j] & 0xF, HEX);
  Serial.print(" ");
}
Serial.println();
```

Figura 2.40 Fragmento 7 código recepción de datos de la red TTN al nodo final.

Primero generamos un retardo de 1 segundo para recibir mensajes de la red e inicializamos la recepción de datos, mientras que con el comando “modem.read” de la librería MKR podremos recibir el mensaje y almacenarlo en una variable tipo “char”.

Desde la red TTN el mensaje se envía en tipo hexadecimal, y se recibe en el nodo final como una cadena de caracteres.

2.2.2.6. Visualización dinámica de los datos desde la red

2.2.2.6.1. Decodificador de datos con Payload formatters en TTN

Como vimos anteriormente los datos recibidos en la red TTN son de tipo hexadecimal, por lo que es poco práctico para interpretarlos de esa manera.

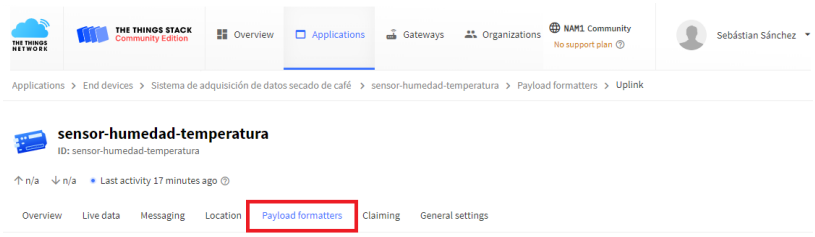

Figura 2.41 Decodificador de la red TTN.

Por esto TTN posee un decodificador de formatos en cual lo podemos encontrar en el nodo final que configuramos en la red TTN.

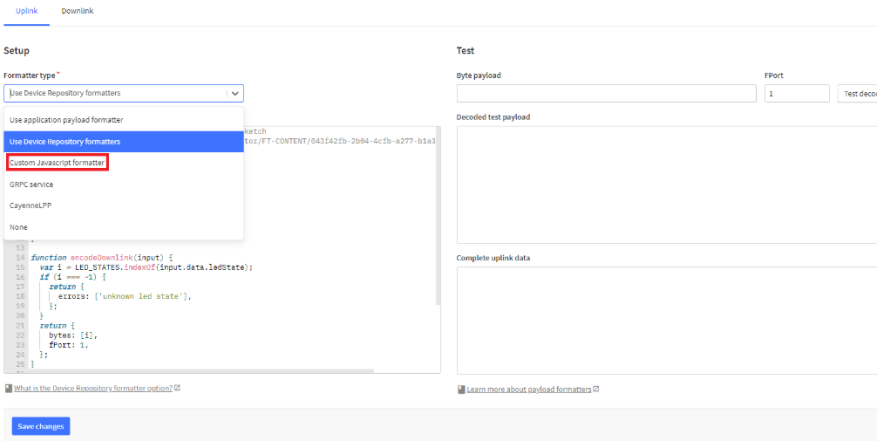

Figura 2.42 Selección del decodificador de la red TTN.

Dentro de la ventana “Payload formatters” podemos escoger el tipo de decodificador, para nuestro caso seleccionamos un decodificador de tipo JavaScript. Además, en la misma ventana tenemos un apartado de pruebas, con el cual podemos verificar el funcionamiento del decodificador que implementemos. El código implementado en JavaScript (ANEXO IV) es el siguiente:

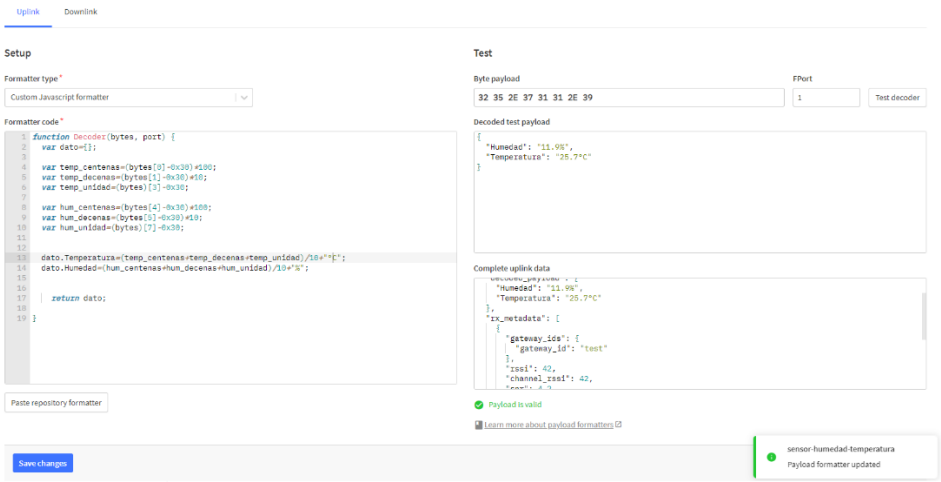

Figura 2.43 Código y test de decodificación de datos en la red TTN.

2.2.2.6.2. Integración con plataformas IoT

Con los datos recibidos del nodo final y decodificados para su mejor manejo, podemos hacer uso de plataformas IoT asociadas con TTN, donde podemos integrar nuestros datos recolectados por los sensores y poder visualizarlos de una forma más dinámica ya sea en figuras, tablas, histogramas, etc.

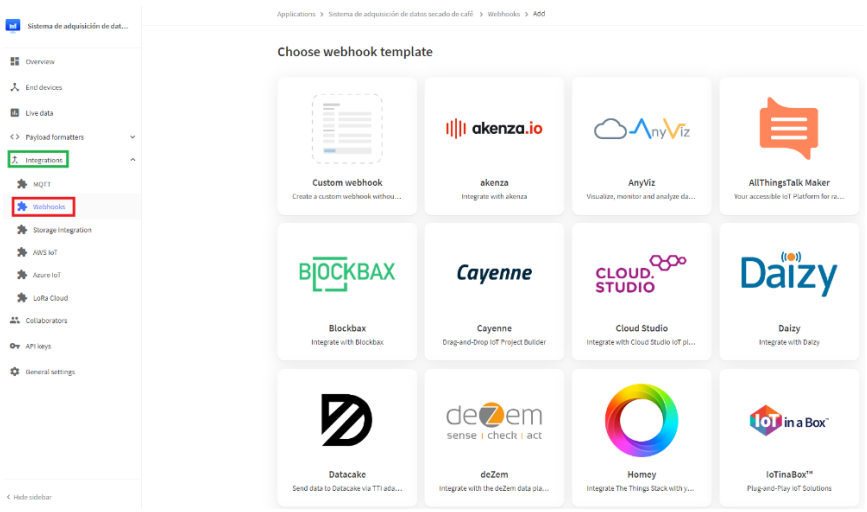

Figura 2.44 Plataformas IoT compatibles con TTN.

Para poder ver estas plataformas, en la parte izquierda de nuestra aplicación creada, se puede ver una barra con distintas pestañas en las que se puede navegar, al dar clic en "Integración" se despliega una nueva lista, nos vamos a enfocar en el apartado "Webhooks" aquí podremos ver todas las plataformas IoT que se pueden integran en nuestra aplicación, cada una de estas plataformas tiene sus requisitos para poder integrarse a la aplicación, TTN posee un sitio web donde se puede ver los requisitos de cada una de estas plataformas y además una guía para poder integrarlos a nuestra aplicación. El link lo podemos encontrar (aquí) [30].

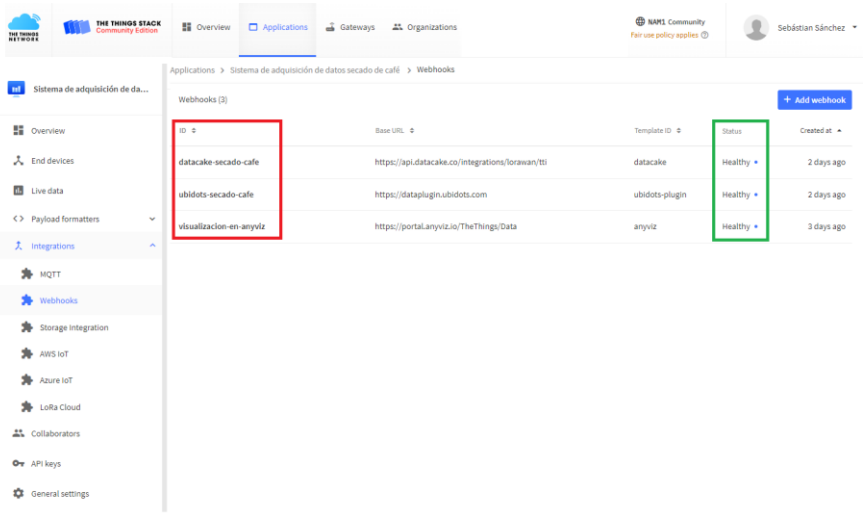

Figura 2.45 Plataformas IoT integradas a la aplicación en TTN.

Una vez se escoge la plataforma IoT que sea del agrado del usuario y se vincule a la aplicación en TTN se podrán ver en la pestaña de "Webhooks" las plataformas integradas, así como el estado de comunicación con TTN, dentro de cada plataforma IoT se manejaran los datos representándolos de la forma que el usuario desee, ya que la mayoría de estas plataformas son intuitivas e interactivas.

2.3. Algoritmo de control

Para el diseño del algoritmo de control de humedad en el grano de café se consideró, según el estudio realizado por Moncayo Solarte y Sánchez Reyes [31], que no existe una estrecha relación entre la temperatura y humedad del aire circundante en el secador parabólico y la temperatura y humedad del grano de café, por lo que es viable estimar los datos de temperatura o humedad del aire en las capas del silo [31].

2.3.1. Obtención de datos

Para la obtención de un modelo aproximado, se basó en los datos de humedad con respecto al tiempo promedio que se obtienen en los secadores parabólicos en países con climas cálidos tropicales como el nuestro. Recordando que la humedad inicial que posee el grano de café recién despulpado es del 55% y su humedad promedio final es del 12%, estos datos dependen de factores ambientales y de ubicación, sin embargo, no suelen variar demasiado.

Según el estudio realizado por Mejía Orozco y Bedoya Yepes [32], el comportamiento aproximado que se obtiene en secadores solares es:

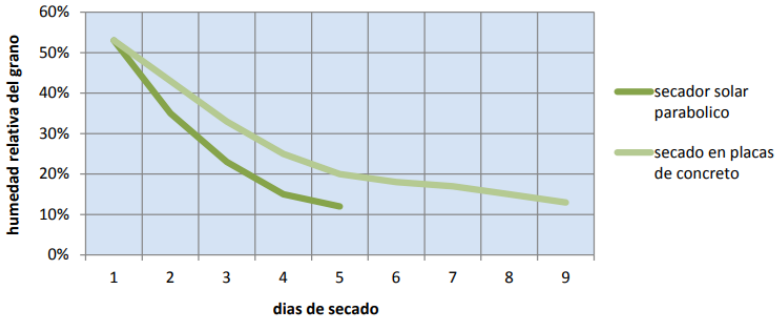

Figura 2.46 Porcentaje de humedad relativa del gano de café con respecto al tiempo [32].

El comportamiento de la humedad en el grano de café es dependiente de la temperatura esto quiere decir que a medida que incrementa la temperatura, la humedad en el grano de café va disminuyendo rápidamente.

2.3.2. Diseño de la planta

Con la curva de reacción obtenida de la Figura 2.46 aplicamos el método de Smith, obteniendo los datos necesarios para completar función de transferencia representada en la Ecuación 1.

$$G = \frac{k}{\tau s + 1} e^{-t_o s}$$

Ecuación 1 Función de transferencia de primer orden método de Smith.

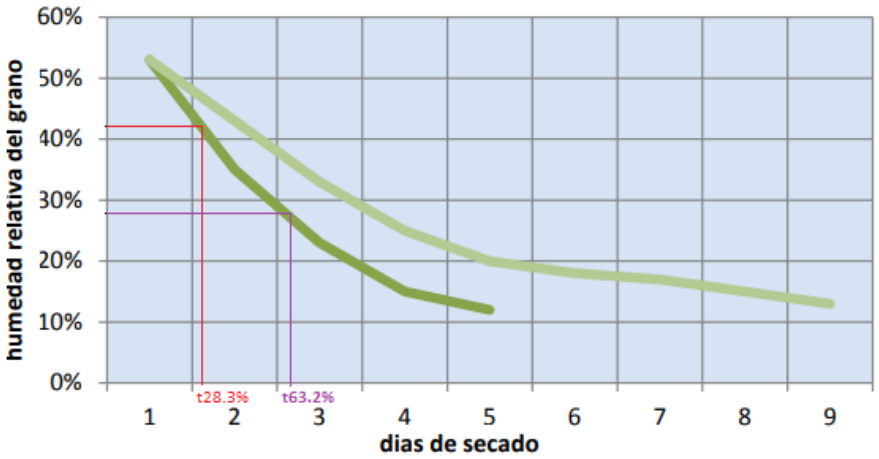

Figura 2.47 Aplicación método de Smith.

Obteniendo los valores de tiempo para un valor del 28.3% y 63.2% de la amplitud total de la curva (k).

Tabla 2.3 Parámetros método de Smith

Parámetro	Valor
k	-43
t63.2%	2.2 días
t28.3%	1.1 días

Con los valores identificados podemos obtener el valor de τ con la Ecuación 2:

$$\tau = 1.5(t_{63.2} - t_{28.3})$$

Ecuación 2 Constante de tiempo tao.

$$\tau = 1.5(2.2 - 1.1) = 1.65$$

Y el valor del coeficiente t_o con la Ecuación 3:

$$t_o = t_{63.2} - \tau$$

Ecuación 3 Tiempo muerto t_o.

$$t_o = 2.2 - 1.65 = 0.55$$

Así obteniendo el modelo aproximado del sistema:

$$G = \frac{-43}{1.65s + 1} e^{-0.55s}$$

Ecuación 4 Función de transferencia modelo aproximado.

Con el modelo aproximado del sistema, diseñamos la planta en el software de simulación Simulink de Matlab (que se puede ver en la Figura 3.21)

2.3.3. Diseño del controlador

Debido a las características del sistema se optó por el uso de un controlador PID que nos proporciona una respuesta mucho más rápida, estabilidad y eliminación de error del controlador.

2.3.3.1. Sintonización por método de Dahlin.

El método de Dahlin hace uso de los parámetros encontrados para la función de transferencia de la Ecuación 4. Para sintonizar el controlador PID Dahlin propone 3 ecuaciones:

$$\mathrm{k_p} = \frac{1}{2k}\left(\frac{t_o}{\tau}\right)^{-1}$$

Ecuación 5 Parámetro de ajuste proporcional.

$$\tau_i = \tau$$

Ecuación 6 Parámetro de ajuste integral.

$$\tau_d = \frac{\mathrm{t_o}}{2}$$

Ecuación 7 Parámetro de ajuste derivativo.

Haciendo uso de la Ecuación 5 reemplazamos los datos obtenidos anteriormente:

$$\mathrm{k_p} = \frac{1}{2(-43)}\left(\frac{0.55}{1.65}\right)^{-1} = -0.03488$$

Obteniendo el parámetro de ajuste proporcional para el controlador.

Posteriormente se hace uso de la Ecuación 6, en el que el valor del parámetro de ajuste integral es el mismo al de la constante de tiempo tao.

$$\tau_i = 1.65$$

Finalmente, haciendo uso de la Ecuación 7, dividimos para dos el valor de la constante de tiempo t_o para obtener el valor del parámetro de ajuste derivativo.

$$\tau_d = \frac{0.55}{2} = 0.275$$

Con los tres parámetros obtenidos, diseñamos el controlador PID haciendo uso del software de simulación Simulink de Matlab (que se puede ver en la Figura 3.23).

2.3.4. Simulación del controlador

Con la planta previamente creada en Simulink, añadimos el bloque “PID Controller”,

configuramos los parámetros de ajuste proporcional, integral y derivativo obtenidos anteriormente, y cerramos el lazo de realimentación.

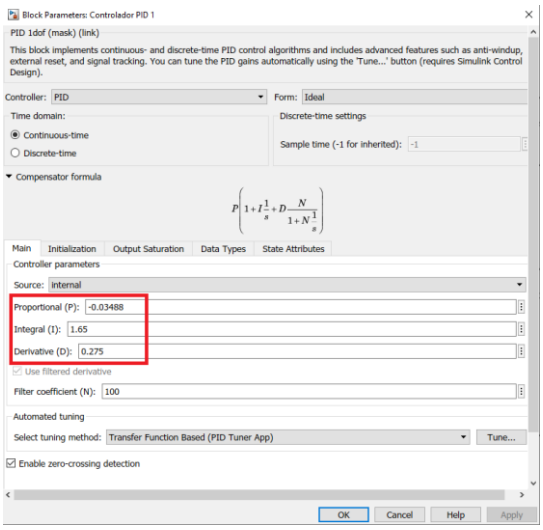

Figura 2.48 Configuración del controlador PID.

Los parámetros determinados por el método de Dahlin nos dan un aproximado a una respuesta estable, sin embargo, hay ocasiones en los que estos parámetros debe ser ajustados, en el caso de nuestro controlador la respuesta presenta un sobre-pico bastante grande y varias oscilaciones (la respuesta a este controlador se puede ver en la Figura 3.24), para corregir esta respuesta se realizó unos pequeños ajustes en los parámetros proporcional e integral.

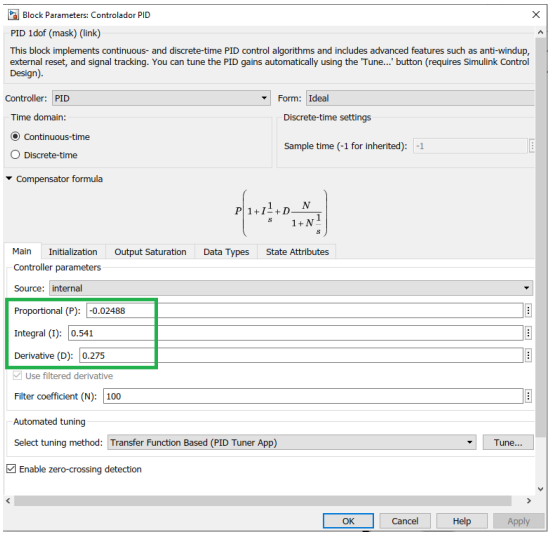

Figura 2.49 Ajuste de parámetros del controlador PID.

Con los parámetros ajustados, la respuesta del controlador cumple con las expectativas, ya que posee un tiempo de estabilización aceptable para el proceso, no presenta sobre picos ni oscilaciones. (la respuesta ajustada a este controlador se puede ver en la Figura 3.25).

3. RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1. Resultados

En esta sección se muestran los resultados obtenidos de las simulaciones del sistema de adquisición de datos y del algoritmo de control de humedad. Interfaces graficas de las distintas integraciones IoT vinculadas a la aplicación principal del proceso de secado de café en la red The Things Network.

La sección 3.1.1 muestra los resultados de la comunicación entre el nodo final y la aplicación en la red TTN mediante la tecnología LoRaWAN. En la sección 3.1.2 se muestra los resultados de la comunicación entre la red TTN y diversas integraciones IoT, el manejo de datos creando interfaces gráficas para un posterior análisis. Finalmente, en la sección 3.1.3 se muestran los resultados y análisis de la respuesta del controlador PID desarrollado en la sección 2.3.

3.1.1. Conectividad entre el nodo final y la red TTN

Con la creación del nodo final para la adquisición de datos de temperatura y humedad en el proceso de secado de café y la creación de su respectiva aplicación en la red TTN como se observó en la sección 2.2 se realizó unas pruebas de conectividad simulando el envío de un mensaje del nodo final hacia la red TTN.

Las características principales de la tecnología LoRaWAN son, el largo alcance que posee sus dispositivos para el envío de datos y su poco consumo de energía al hacerlo, sin embargo, el precio a pagar es el limitado ancho de banda, es por esto por lo que la propia librería MRK de Arduino transforma los datos (que deben estar en formato “String”) a formato hexadecimal, para obtener una comunicación lo más optimizada posible.

El método de activación usado para la comunicación es el método OTAA (Over the air activation) por sus siglas en inglés, esto significa que el nodo final requiere de una serie de credenciales (DevEUI y AppKey), necesarias para la generación de las claves de inicio de sesión (NwkSkey y AppSkey). Las ventajas del método OTAA son que, el dispositivo genera las claves solo cuando es necesario y si el dispositivo debe unirse a una nueva red puede hacerlo generando un nuevo conjunto de claves sin la necesidad de volver a ser programado.

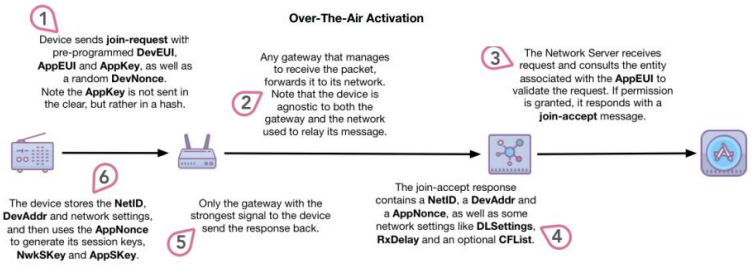

Figura 3.1 Método de activación OTAA [33].

Con el nodo final enlazado a la red, podemos simular el envío de un mensaje en hexadecimal, el primer mensaje en hexadecimal será “31 35”.

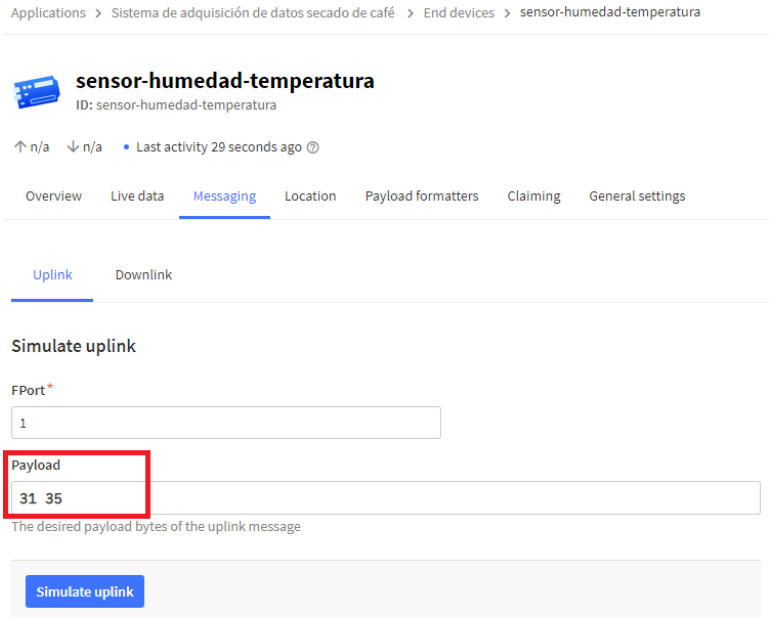

Figura 3.2 Simulación de envío de mensaje a la red TTN.

En la Figura 3.2 podemos ver que dentro del nodo final configurado en TTN existe un apartado de mensaje, aquí se puede seleccionar el puerto de envío y escribir un mensaje, una vez hecho eso se da clic en el botón “Simulate uplink” para simular el envío del mensaje.

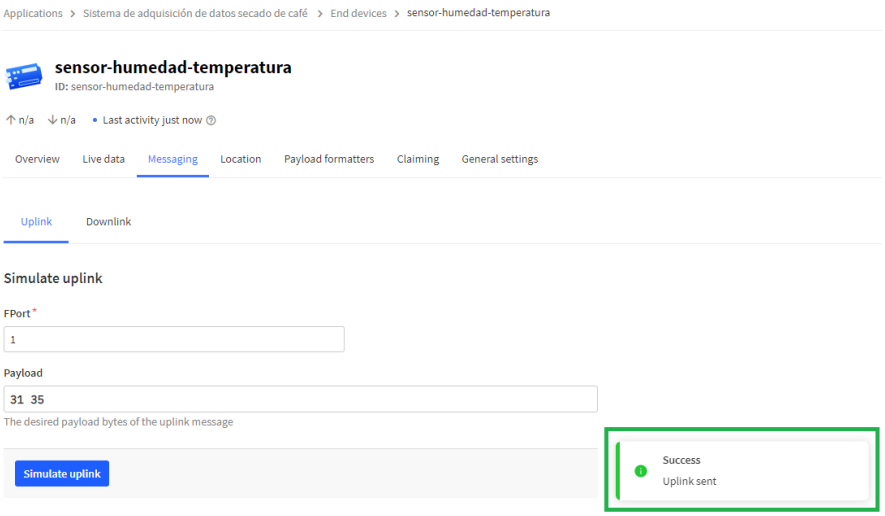

Figura 3.3 Envío de mensaje exitoso.

Si el mensaje fue enviado correctamente, en la esquina inferior derecha de la pantalla aparecerá un mensaje de verificación como se muestra en la Figura 3.3.

Los datos enviados se pueden observar en vivo, en la pestaña “Live data”

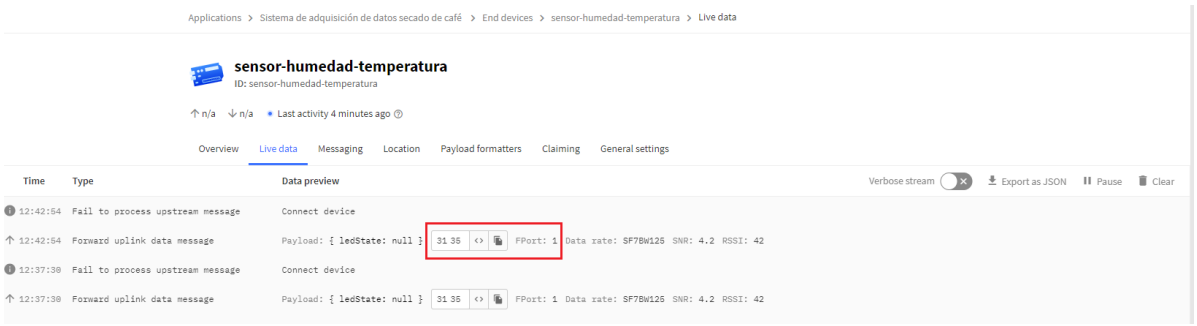

Figura 3.4 Visualización en vivo del mensaje enviado.

Como se puede observar en la Figura 3.4 el mensaje “31 35” antes mencionado aparece en la pantalla de datos en vivo, lo que quiere decir que no existió algún problema en la recepción del dato. En la pestaña de “Live data” también podemos ver la última actividad del nodo final (parte superior de la pantalla) y las opciones de pausar, eliminar y exportar la visualización en vivo (parte superior derecha).

Los mensajes en hexadecimal no son muy prácticos para su interpretación, es por eso por lo que para una mejor visualización de los datos recibidos se realizó un código de decodificación (ANEXO IV) ubicado en la pestaña “Payload formatters”

La decodificación en hexadecimal es muy sencilla, como sabemos que los datos que vamos a recibir de humedad y temperatura son números naturales, buscamos en una tabla de sistemas de numeración el valor en hexadecimal de los números del 0 al 9 con los cuales podemos hacer todas las combinaciones de números posibles.

Binary	Dec	Hex	Glyph
010 0000	32	20	SP
010 0001	33	21	!
010 0010	34	22	"
010 0011	35	23	#
010 0100	36	24	$
010 0101	37	25	%
010 0110	38	26	&
010 0111	39	27	'
010 1000	40	28	(
010 1001	41	29	)
010 1010	42	2A	*
010 1011	43	2B	+
010 1100	44	2C	,
010 1101	45	2D	-
010 1110	46	2E	.
010 1111	47	2F	/
011 0000	48	30	0
011 0001	49	31	1
011 0010	50	32	2
011 0011	51	33	3
011 0100	52	34	4
011 0101	53	35	5
011 0110	54	36	6
011 0111	55	37	7
011 1000	56	38	8
011 1001	57	39	9

Binary	Dec	Hex	Glyph
011 1010	58	3A	:
011 1011	59	3B	;
011 1100	60	3C	<
011 1101	61	3D	=
011 1110	62	3E	>
011 1111	63	3F	?
100 0000	64	40	@
100 0001	65	41	A
100 0010	66	42	B
100 0011	67	43	C
100 0100	68	44	D
100 0101	69	45	E
100 0110	70	46	F
100 0111	71	47	G
100 1000	72	48	H
100 1001	73	49	I
100 1010	74	4A	J
100 1011	75	4B	K
100 1100	76	4C	L
100 1101	77	4D	M
100 1110	78	4E	N
100 1111	79	4F	O
101 0000	80	50	P
101 0001	81	51	Q
101 0010	82	52	R
101 0011	83	53	S

Binary	Dec	Hex	Glyph
101 0100	84	54	T
101 0101	85	55	U
101 0110	86	56	V
101 0111	87	57	W
101 1000	88	58	X
101 1001	89	59	Y
101 1010	90	5A	Z
101 1011	91	5B	[
101 1100	92	5C	\
101 1101	93	5D	]
101 1110	94	5E	^
101 1111	95	5F	_
110 0000	96	60	`
110 0001	97	61	a
110 0010	98	62	b
110 0011	99	63	c
110 0100	100	64	d
110 0101	101	65	e
110 0110	102	66	f
110 0111	103	67	g
110 1000	104	68	h
110 1001	105	69	i
110 1010	106	6A	j
110 1011	107	6B	k
110 1100	108	6C	l

Binary	Dec	Hex	Glyph
110 1101	109	6D	m
110 1110	110	6E	n
110 1111	111	6F	o
111 0000	112	70	p
111 0001	113	71	q
111 0010	114	72	r
111 0011	115	73	s
111 0100	116	74	t
111 0101	117	75	u
111 0110	118	76	v
111 0111	119	77	w
111 1000	120	78	x
111 1001	121	79	y
111 1010	122	7A	z
111 1011	123	7B	{
111 1100	124	7C	|
111 1101	125	7D	}
111 1110	126	7E	~

Figura 3.5 Sistemas de numeración [34].

Como podemos observar en la Figura 3.5 los números del 0 al 9 tienen un valor del 30 al 39 en hexadecimal, por lo tanto, en el código de decodificación cada byte recibido se le resta el valor “0x30” para obtener el dato numérico.

Con el código de decodificación funcionando vamos a enviar un nuevo mensaje el cual será : “31 32 2E 31 32 34 2E 35” donde los cuatro primeros bytes corresponden al dato de humedad y los cuatro últimos bytes corresponden al dato de temperatura, haciendo uso del sistema de numeración de la Figura 3.5 obtenemos que el dato a enviar será: “12.1 24.5” en el cual 12.1 será el dato de humedad y 24.5 será el dato de temperatura.

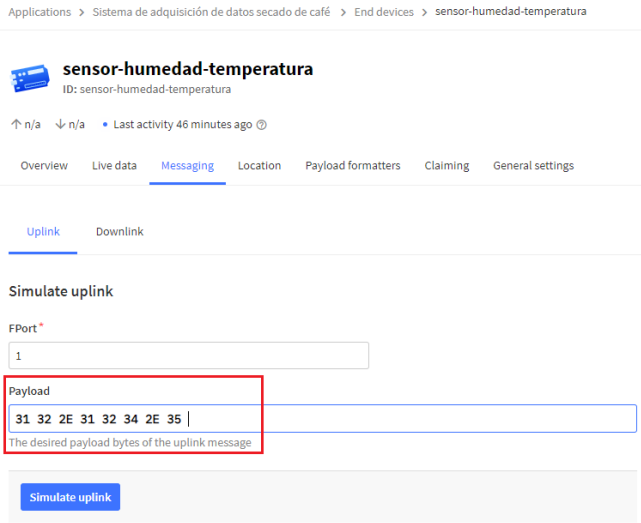

Figura 3.6 Simulación envío de datos del nodo final a la red TTN.

El mensaje se envió correctamente y podrá ser visto en la pestaña de datos en vivo.

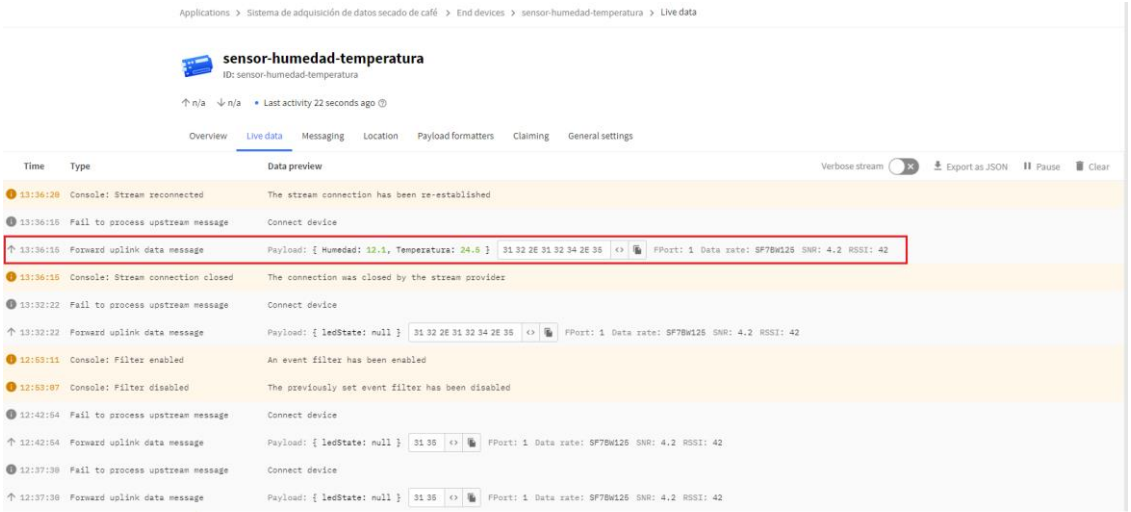

Figura 3.7 Visualización en vivo de los datos enviados.

Como podemos observar en la Figura 3.7 se observa todos los mensajes previamente enviados y resaltado se encuentra los datos de humedad y temperatura que se enviaron en la Figura 3.6.

Podemos observar que hay un texto que dice “Payload” y en llaves encierra el mensaje decodificado el cual es “ Humedad: 12.1, Temperatura: 24.5” y comprobamos que la decodificación es correcta, seguido del mensaje decodificado tenemos el mensaje en hexadecimal.

Para comprobar el funcionamiento de la comunicación entre el nodo final y la red TTN se envió una serie de datos, simulando el comportamiento del proceso de secado de café al sol, en lapsos de tiempo de un minuto, obteniendo los siguientes datos en la red.

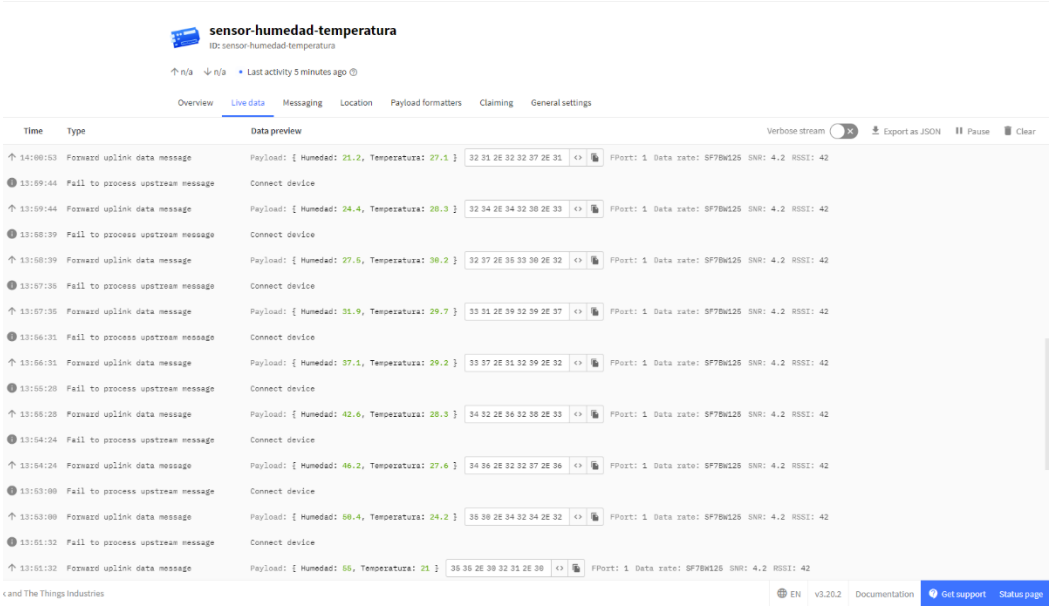

Figura 3.8 Simulación envío de datos de humedad y temperatura a la red.

En la Figura 3.8 se puede observar que los datos de humedad y temperatura fueron enviados aproximadamente cada minuto, todos los datos fueron recibidos por la red TTN y decodificados para su visualización

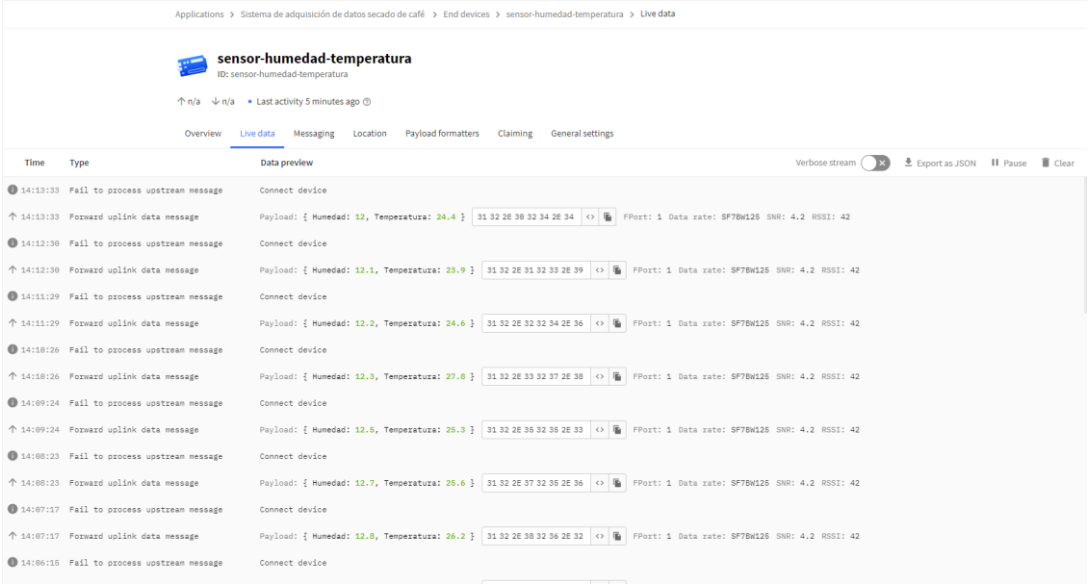

Figura 3.9 Simulación envío de datos de humedad y temperatura a la red.

Se continúo enviado datos de temperatura y humedad hasta alcanzar una humedad relativa del 12% que es el valor optimo que debe contener el grano de café para continuar con la etapa de tostado.

3.1.2. Comunicación entre la red TTN y aplicaciones IoT

Con la comunicación establecida entre el nodo final y la red el siguiente paso es verificar el funcionamiento de la comunicación con las distintas aplicaciones IoT con las The Things Network puede vincularse.

Existen numerosas aplicaciones que podemos integras a nuestra aplicación TTN, existe la posibilidad de integrar más de un Webhook a nuestra aplicación, en nuestro caso integramos a nuestra aplicación el Webhook de AnyViz y Datacake, ambas aplicaciones IoT solo requieren una cuenta en sus respectivas páginas web para poder integrarse a la red TTN.

3.1.2.1. Funcionamiento con AnyViz

AnyViz es una interfaz IoT que permite visualizar datos adquiridos de la red TTN de una forma mucho más dinámica, los datos pueden ser importados y exportados en archivos Excel para su posterior análisis.

Su tablero principal es muy intuitivo y fácil de manejar, simplemente se arrastra el elemento que se desee integrar en la pantalla y se lo configura al gusto del usuario.

Para verificar el funcionamiento de AnyViz, se creó un tablero principal y se agregó el valor en tiempo real de la humedad y de la temperatura, también una interfaz gráfica para observar su comportamiento con el paso del tiempo y finalmente un mapa con la ubicación del nodo final.

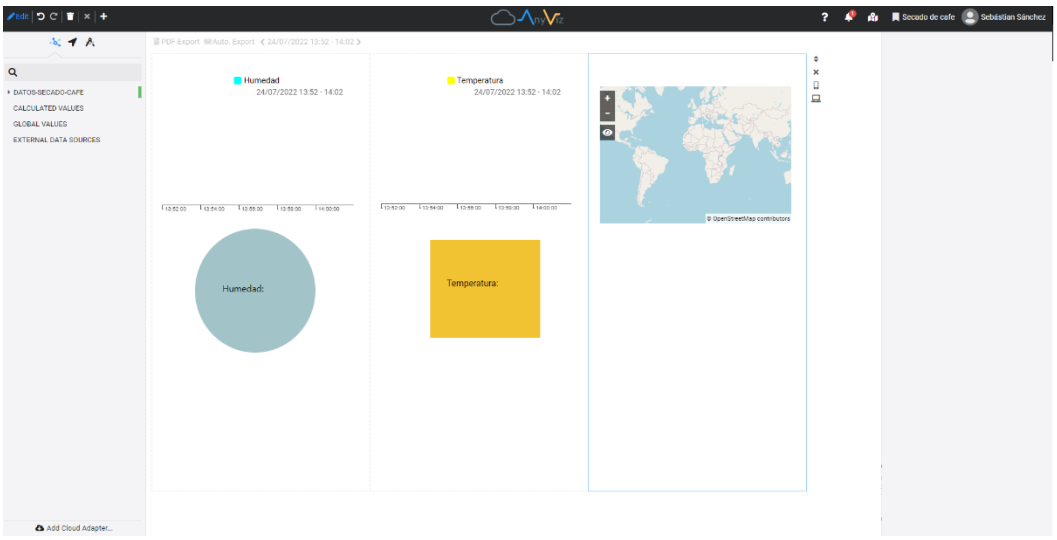

Figura 3.10 Tablero principal AnyViz.

La comunicación de AnyViz con la red TTN es instantánea y no requiere de decodificación previa, simplemente el dato que llega del nodo final a la red es enviado a AnyViz ya decodificado por la misma red TTN. Para la simulación se envió una seria de datos simulando el comportamiento del proceso de secado de café al sol.

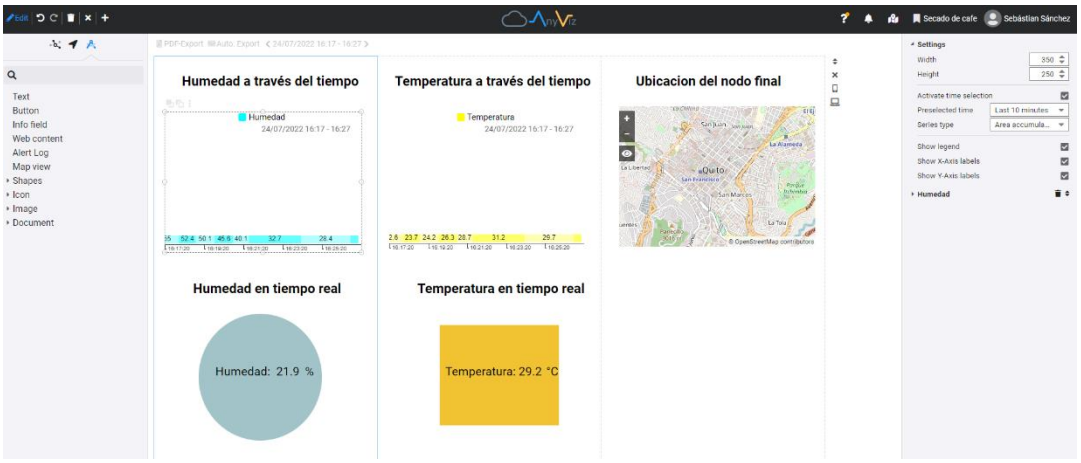

Figura 3.11 Simulación recepción de datos desde la red TTN a AnyViz.

En la Figura 3.11 se pueden observar los datos de humedad y temperatura en ese instante de tiempo, también una interfaz gráfica donde se observa el comportamiento de estos datos a medida que son adquiridos, además de la ubicación del nodo final.

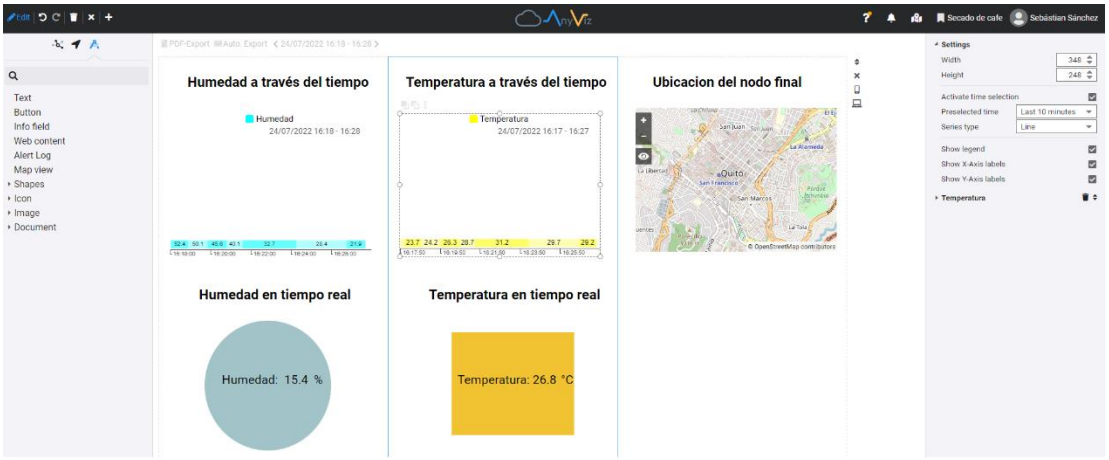

Figura 3.12 Simulación recepción de datos desde la red TTN a AnyViz.

Se continúo enviando varios datos desde el nodo principal y en la Figura 3.12 se ven reflejados, para ese instante se obtuvo una humedad del 15.4% y una temperatura del 26.8°C, los datos posteriores a los mencionados se pueden ver en la interfaz gráfica, esta interfaz puede ser configurada para observar datos anteriores hasta un máximo de 1 año y un mínimo de 1 minuto.

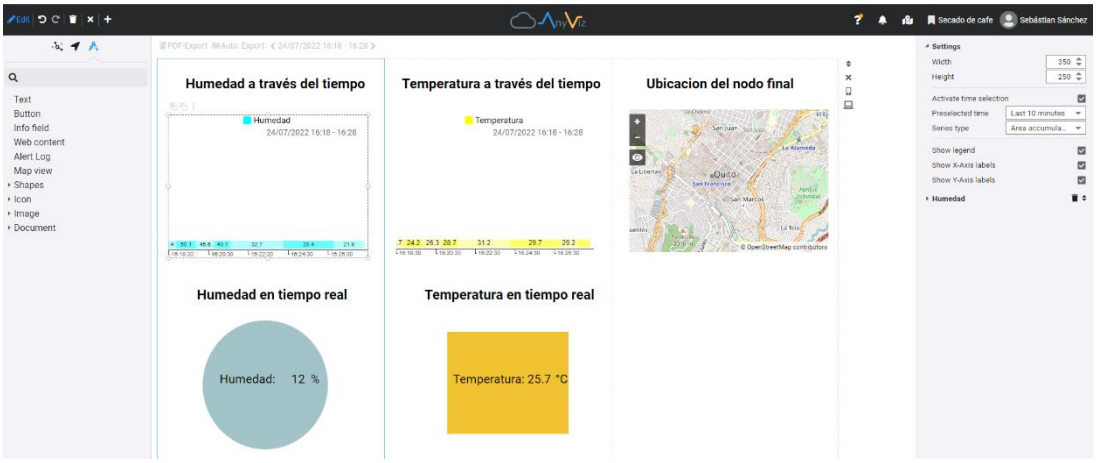

Figura 3.13 Simulación recepción de datos desde la red TTN a AnyViz.

La simulación de envío de datos termina cuando se alcanza un valor de humedad aproximado del 12%. Se puede ver que no hubo algún error en la comunicación entre la red TTN y la aplicación de AnyViz, los datos llegan a la aplicación de forma inmediata una vez hayan sido decodificados en la aplicación de TTN. AnyViz posee la característica de configurar alarmas de aviso cuando algún dato llegue a cierto tipo de valores, esta alarma se notifica en el tablero principal y también se puede enviar por correo electrónico o mensaje a cualquier otro usuario.

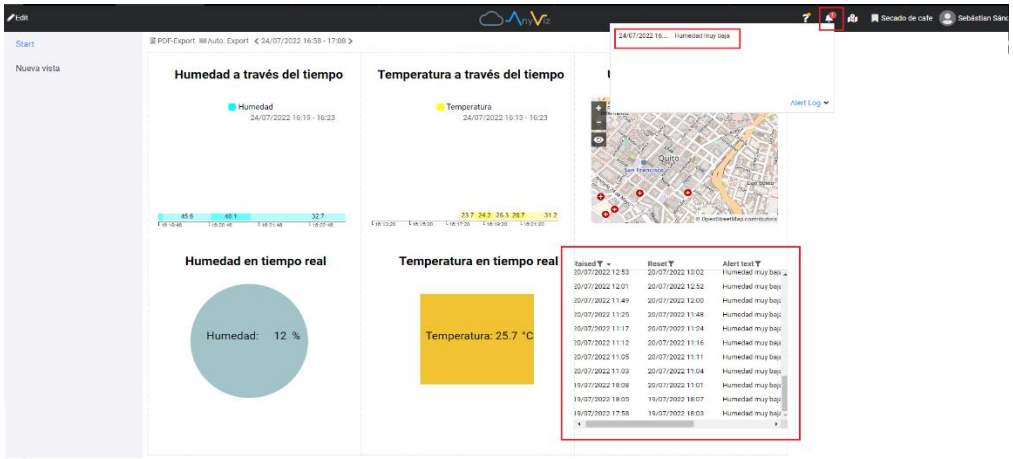

Figura 3.14 Panel de alarmas de AnyViz.

3.1.2.2. Funcionamiento con Datacake

La interfaz de Datacake es muy intuitiva y dinámica, donde en el tablero principal se pueden añadir una serie de widgets.

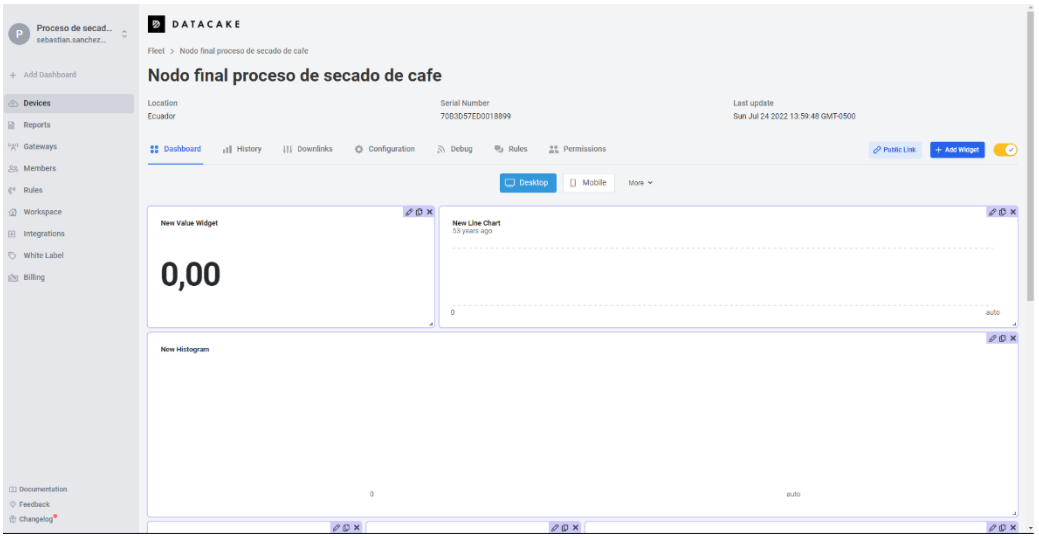

Figura 3.15 Tablero principal Datacake.

Los widgets que más resaltan son el de mostrar un valor, mostrar un gráfico y mostrar un histograma, cada uno de estos widgets se puede modificar en tamaño y posición, para ubicarlos en el lugar donde sea más cómodo de visualizar para el usuario.

Una vez añadimos los widgets que más útiles sean para la visualización y análisis de datos, se los configura de manera individual, asignándole a cada uno un dato y personalizándolo al gusto.

Cuando se logra vincular Datacake con TTN como se mencionó en la sección 2.2.2.6.2 lo primero que se debe hacer es decodificar los datos nuevamente, ya que algunas aplicaciones IoT reciben los datos en formato hexadecimal, el código de decodificación de Datacake se encuentra en el ANEXO V.

Con los datos decodificados, se procede a añadir a cada uno de estos un widget para su mejor visualización.

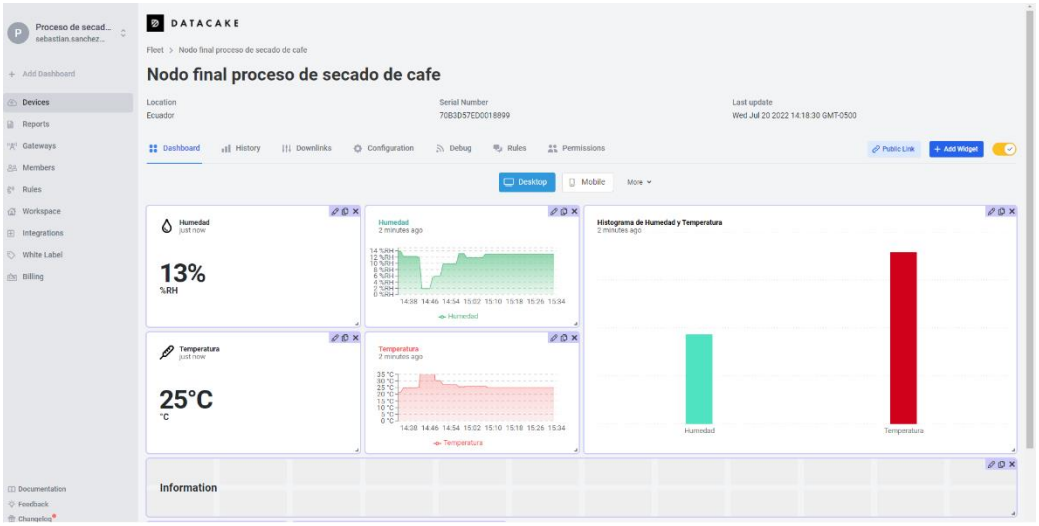

Figura 3.16 Visualización de datos en Datacake.

Como se puede observar en la Figura 3.16 se añadieron varios widgets personalizados al panel principal y se les asigno los datos de humedad y temperatura recibidos de la red TTN.

Para comprobar el funcionamiento de la comunicación entre la red TTN y Datacake se envió una serie de datos simulando el proceso de secado al sol del grano de café. Cada dato fue enviado en lapsos de un minuto.

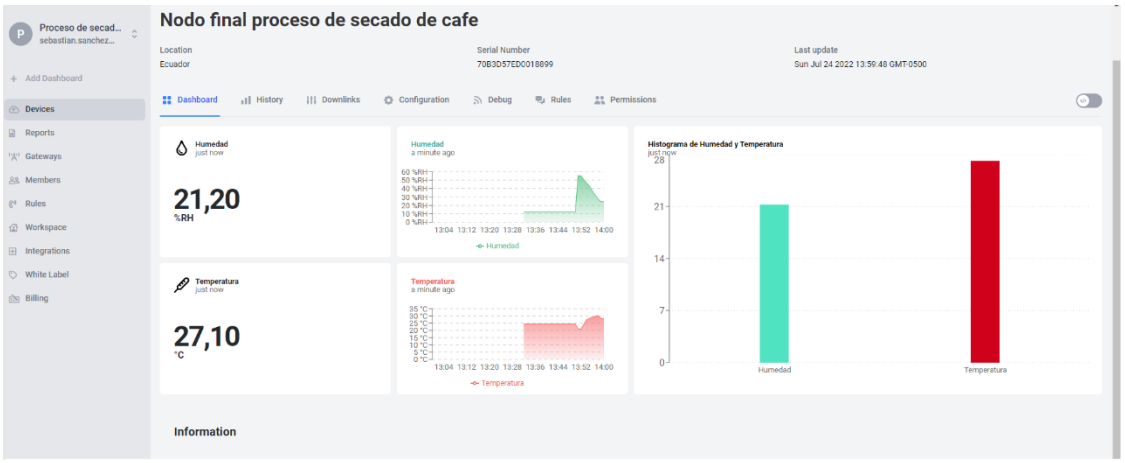

Figura 3.17 Simulación recepción de datos desde la red TTN a Datacake.

En la Figura 3.17 se puede visualizar el valor exacto de la humedad y temperatura en ese instante, así como una gráfica con los valores en el paso del tiempo pudiendo así observar su variación en el día y finalmente un histograma donde se compara el porcentaje de humedad relativa con la temperatura del ambiente, ambos valores en ese instante.

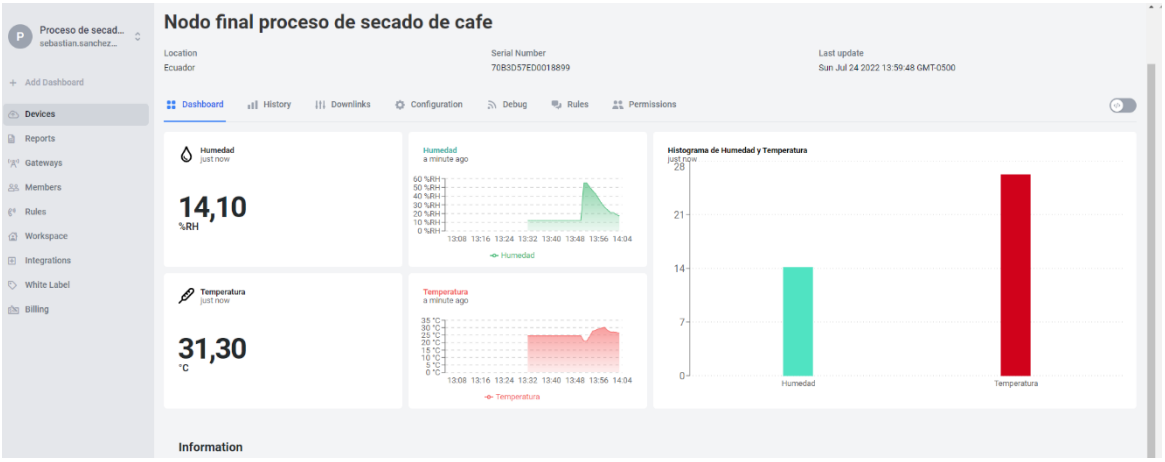

Figura 3.18 Simulación recepción de datos desde la red TTN a Datacake

Se continúo enviando diferentes tipos de datos para comprobar la comunicación, cada uno de los datos fue recibido en Datacake al instante que el dato llego a la red, la comunicación es instantánea tanto en el nodo final con la red TTN y la red TTN con Datacake.

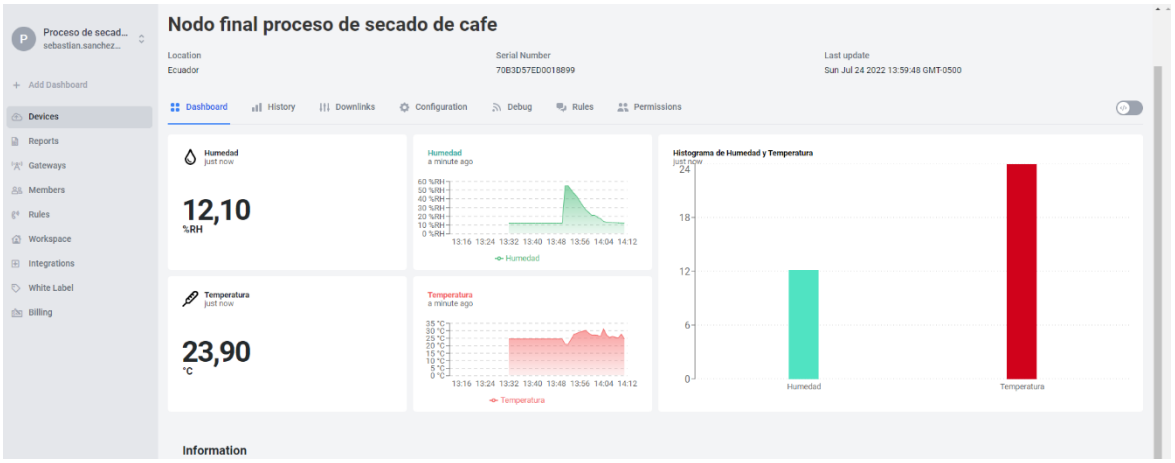

Figura 3.19 Simulación recepción de datos desde la red TTN a Datacake

Finalmente, el envío de datos finalizó al aproximarse a un valor de humedad relativa del 12%, la ventaja que proporciona Datacake es su fácil visualización de los datos, en la gráfica de humedad se puede observar un comportamiento similar al de la Figura 2.46, donde su humedad relativa disminuyo del 55% al 12% aproximadamente.

Con los datos dentro de la interfaz de Datacake se puede hacer un análisis minucioso de estos y así determinar qué factores influyen en la etapa de secado para obtener un café de alta calidad.

Datacake también tiene la posibilidad de visualizar los datos desde cualquier dispositivo móvil con conectividad a internet, el proceso de personalización de su tablero principal es el mismo que se realizó para ordenadores.

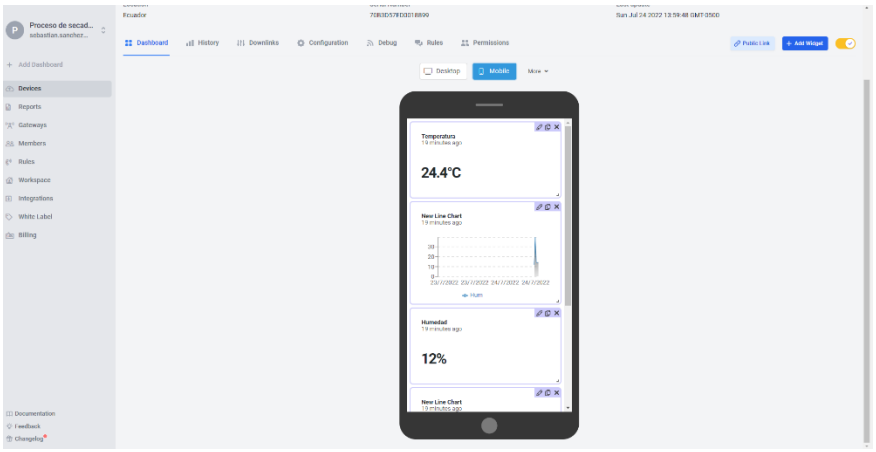

Figura 3.20 Interfaz de Datacake para móviles.

3.1.3. Simulación del controlador PID

Como vimos en la sección 2.3.2 se obtuvo el diseño de la planta, representada por la **Ecuación *4***, se usó el software de simulación Simulink para obtener la gráfica de la planta modelada.

En Simulink, se colocó la función de transferencia, el retardo correspondiente y una función “Step” para poder observar su respuesta ante una entrada escalón.

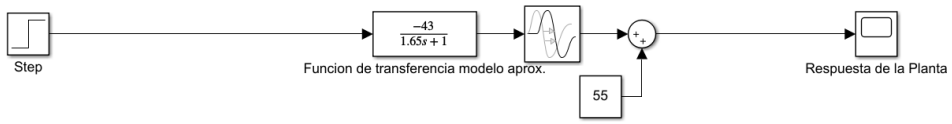

Figura 3.21 Planta simulada en Matlab Simulink.

La respuesta obtenida del modelo fue la siguiente:

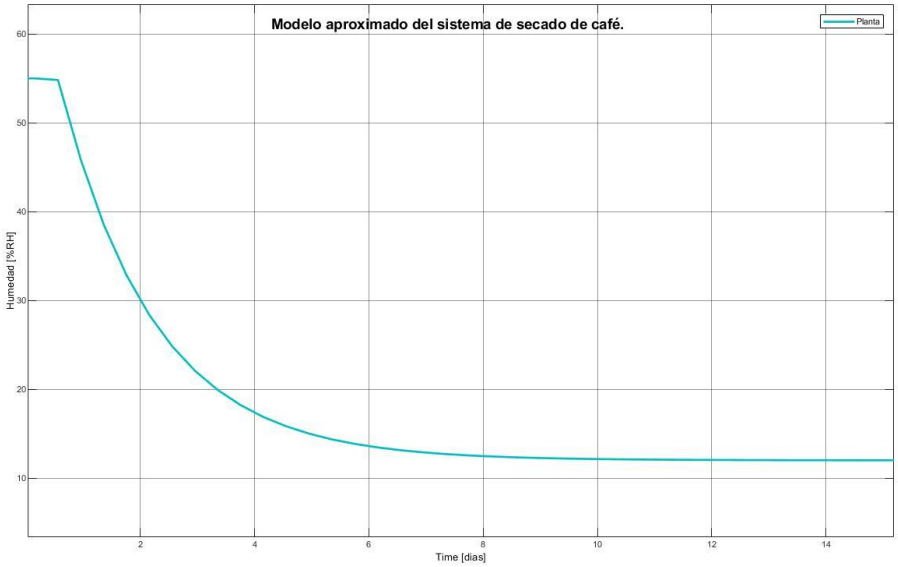

Figura 3.22 Simulación modelo aproximado de la planta.

El comportamiento del modelo aproximado refleja muy bien las características vistas en la Figura 2.46, se puede observar que la humedad decrementa de un valor del 55% hasta un valor aproximado del 12%, simulando el proceso de secado de café al sol, el periodo que le toma alcanzar este valor depende de factores ambientales, pero suele estar entre los 7 y 15 días, como se muestra en la respuesta del modelo aproximado.

Con el diseño exitoso de la planta, se procede a verificar el funcionamiento del controlador PID, este fue sintonizado por el método de Dahlin como se vio en la sección 2.3.3.1.

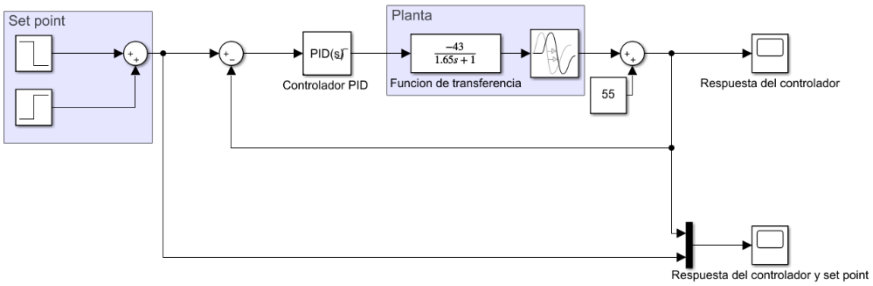

Figura 3.23 Controlador PID simulado en Matlab Simulink.

En Simulink se colocó el bloque “PID Controller” y se configuró con los parámetros

encontrados, como se muestra en la Figura 2.48, obteniendo:

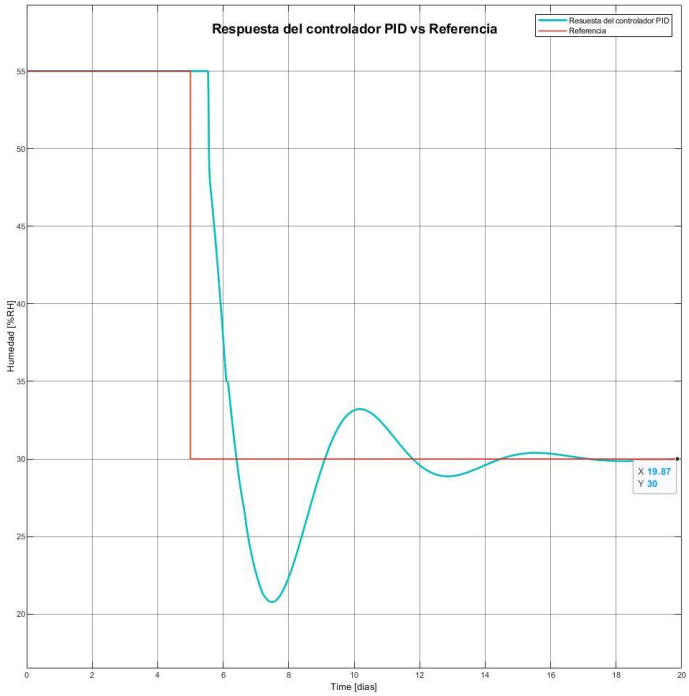

Figura 3.24 Respuesta del controlado PID.

En la gráfica de la Figura 3.24 se puede observar que la respuesta del controlador PID pose un sobre-pico muy alto, aproximadamente del 36.7%, además presenta oscilaciones y su tiempo de establecimiento es muy largo para el proceso para el que está hecho.

Por este motivo se realizó un pequeño ajuste en los parámetros de controlador, como se vio en la Figura 2.49, obteniendo una nueva respuesta:

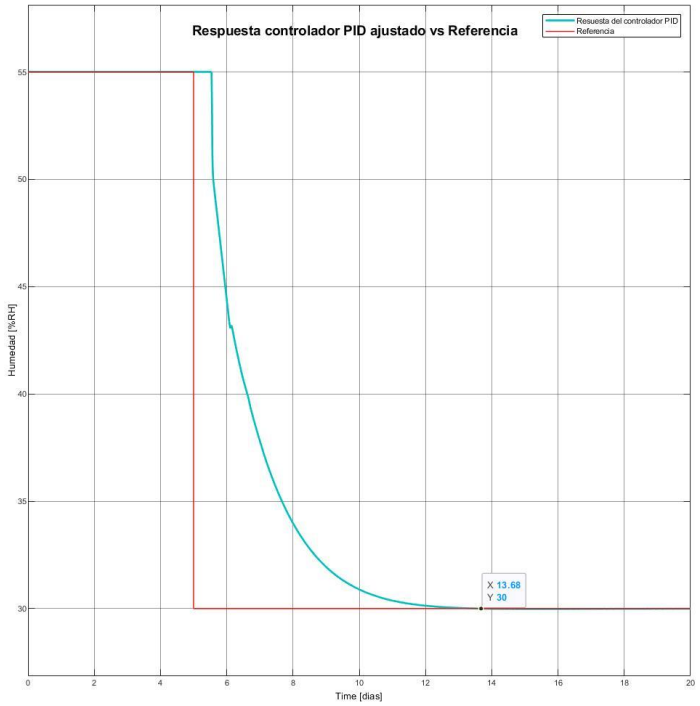

Figura 3.25 Respuesta del controlado PID ajustado.

Con los ajustes realizados en los parámetros proporcional e integral del controlador PID, se obtiene una respuesta mucho más acorde a la necesitada para el proceso, ya que no presenta sobre-picos ni oscilaciones, y su tiempo de establecimiento está dentro del esperado por el proceso.

Para realizar pruebas de funcionamiento, se modificó los parámetros de set point, para que el controlador siga varias referencias.

Se realizó dos cambios de referencias de reducción de humedad, primero se redujo la humedad del 55% al 42% y después se redujo del 42% al 12%

En la Figura 3.26 se puede observar que el controlador funciona de manera correcta ante cambios de referencia.

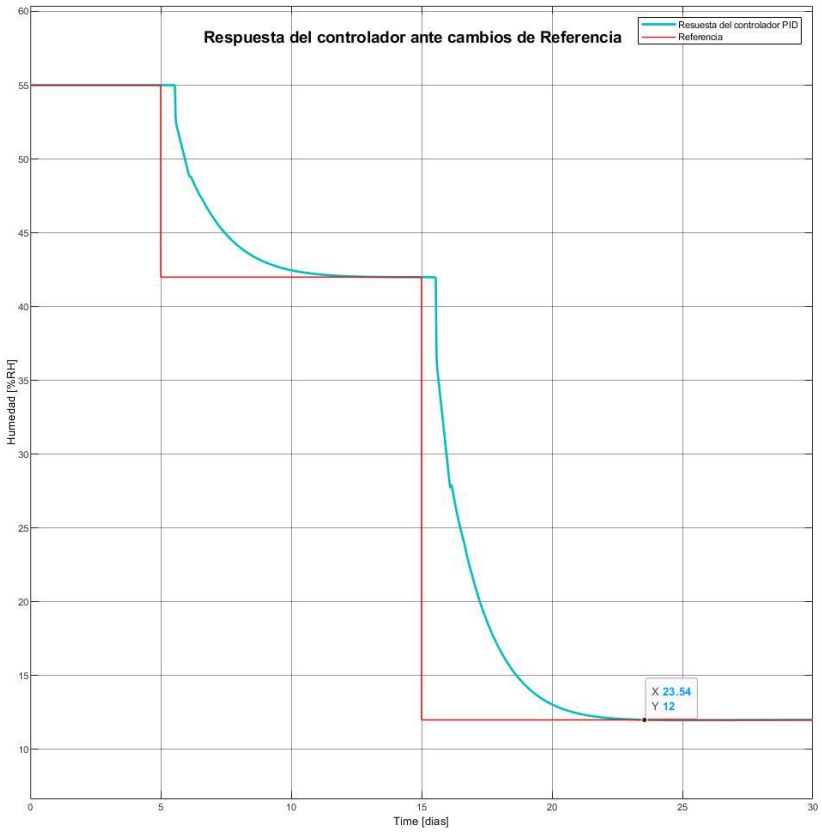

Figura 3.26 Respuesta del controlador ante cambios de referencia.

Después se realizó cambios de referencia donde se redujo la humedad del 55% al 19% para posteriormente aumentarla al 49%.

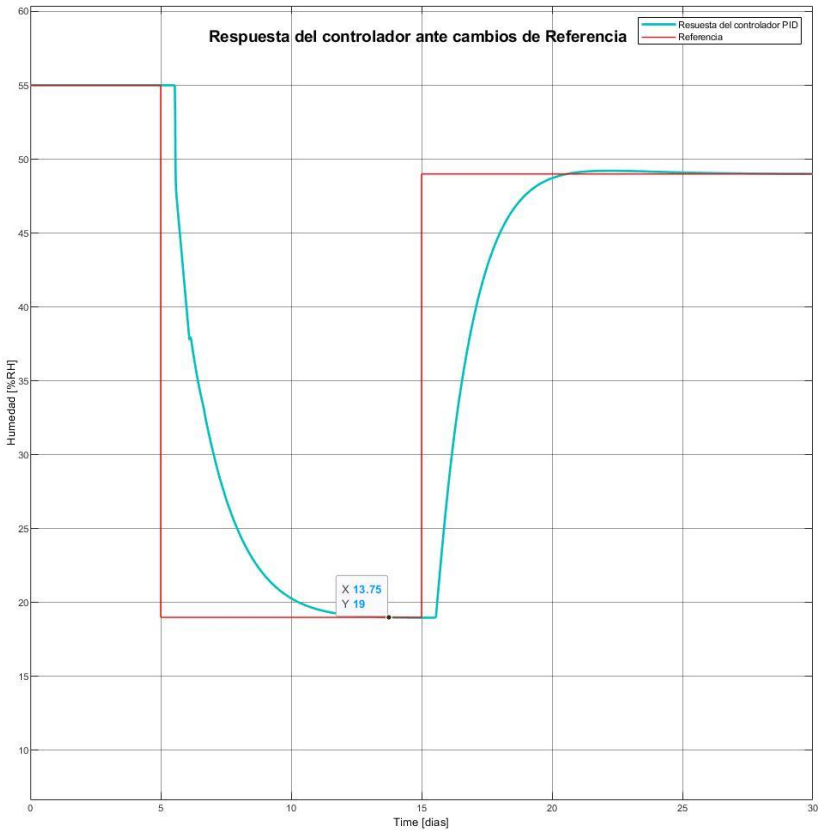

Figura 3.27 Respuesta del controlador ante cambios de referencia.

En la Figura 3.27 se puede observar que el controlador sigue la referencia correctamente cuando la humedad se reduce, pero cuando la humedad aumenta el controlador tarda un poco más en establecerse en la referencia, sin embargo no representa problema alguno, ya que el proceso de secado trata de reducir la humedad relativa de la semilla de café, y casi nunca existirán casos en los que se desee aumentar esta humedad, además expertos caficultores afirma que tratar de aumentar la humedad del grano de café en el proceso de secado puede llegar a alterar muchas de sus propiedades, afectando al olor y sabor del café.

3.2. Conclusiones

- La caficultura en el Ecuador tiene suma importancia a nivel económico, social y ambiental, siendo una de las principales actividades que generan ingresos a los sectores rurales del país, por este motivo el proyecto ayudará a identificar las condiciones óptimas en el proceso de secado para obtener un café de alta calidad, a través de la adquisición de datos como la humedad y temperatura, para su posterior análisis en las diferentes aplicaciones IoT vinculadas al sistema de adquisición de datos.

- Debido a que la gran mayoría de productoras de café se encuentran en zonas rurales alejadas donde es complicado obtener energía eléctrica de manera constante, es necesario de un modelo de adquisición de datos que no requiera alimentación continua, el sistema de adquisición de datos desarrollado en este proyecto cumple al 100% esa necesidad, ya que incorpora una batería recargable de alta duración con la cual el sistema podrá operar sin problemas por un largo periodo de tiempo antes de necesitar ser recargado.

- Las variables de temperatura y humedad son las más importantes en el proceso de secado del café, principalmente esta última, ya que es en la fase de secado donde se definen muchas propiedades de aroma y sabor que obtendrá el grano de café y que serán intensificados en la fase de tostado, y son estas características las que definen el costo de este producto, pudiendo elevarse si cumplen con un excelente aroma y sabor.

- El sensor AM2302/DHT22 es altamente usado en la agricultura, tiene un costo accesible y facilidad de conexión con microcontroladores Arduino gracias a sus librerías de programación. Permite adquirir datos de humedad y temperatura en un lapso de 2 segundos, ya que es el tiempo que demora toda la trama de comunicación del sensor.

- La tarjeta Arduino MKR WAN 1310 tiene la posibilidad de funcionar con baterías Li-Ion y Li-Po recargables, lo que las hace sumamente prácticas para aplicaciones IoT con conectividad LoRa ya que esta optimizadas para consumir la menos cantidad de energía posible.

- El protocolo LoRaWAN ha ganado mucha popularidad en los últimos años debido a su uso en aplicaciones IoT, principalmente en sectores donde se requiere un largo alcance en la comunicación y poco consumo de energía, como en los sectores agrícolas, el sistema de adquisición de datos desarrollado al usar tecnología LoRaWAN garantiza que los datos pueden ser enviados en un rango de hasta 15 Km a la redonda para posteriormente ser monitoreados por alguna aplicación IoT desde el ordenador o el móvil

en cualquier parte del mundo y a tiempo real.

- El nodo final desarrollado es de clase A según la capa MAC de la tecnología LoRaWAN ya que el dispositivo envía los datos del sensor cada cierto periodo de tiempo (programable) y no de forma continua, optimizando el consumo de energía, y solo recibe un mensaje de enlace descendente después de haber enviado un mensaje de enlace ascendente, donde se encuentran los datos adquiridos por el sensor (cifrados) y distintos metadatos donde está la información del dispositivo de nodo final. Esta última característica es importante ya que el sistema es de adquisición de datos, pero puede ser usado para el control del proceso.

- La red a la que se enlaza el sistema de adquisición de datos es The Things Network, debido a que posee una gran cantidad de usuarios a nivel mundial y se pueden usar los dispositivos gateways de estos usuarios para comunicar el nodo final a la red de forma gratuita.

- El protocolo LoRaWAN ofrece dos niveles de seguridad al momento de transmitir o recibir información, el primer nivel es la seguridad de red, donde se garantiza la autenticidad del nodo final y el segundo nivel es la seguridad de aplicación, donde se garantiza que ningún operador de red tenga acceso a los datos dentro de la aplicación del usuario del nodo final.

- Los datos que se envían del nodo final a la red TTN son en formato hexadecimal, para el sistema de adquisición de datos desarrollado los datos de humedad y temperatura se optó por enviar de forma concatenada como una cadena de 8 bytes, donde los cuatro primeros bytes son la humedad y los cuatro restantes son la temperatura, para optimizar la comunicación por el ancho de banda limitado de LoRaWAN.

- La implementación de un decodificador en la red TTN se ajusta a las necesidades del usuario, ya que es este el que lo programa en función a la forma en la que son enviados los datos del nodo final a la red.

- La plataforma IoT de AnyViz tiene la característica de poder exportar los datos adquiridos en formato de Excel, esto ayuda mucho para un posterior análisis riguroso de los datos y conseguir las características que influyen para obtener un café de alta calidad.

- La plataforma IoT de Datacake permite vincular solo una aplicación con TTN de forma gratuita, para poder vincular más aplicaciones es necesario pagar una membresía, la

cual no es costosa, la membresía cuesta $1 al mes o $5 de manera anual y ofrece una retención de datos más duradera y lectura de mayor cantidad de datos por día.

- Para el proceso de secado del café concretamente el método de secado al sol que es el más usado en el país, es muy complicado lograr un control automático del mismo, debido a que el proceso en si es muy artesanal y depende de factores ambientales como la cantidad de luz solar o lluvias, imposibles de controlar, por ese motivo un sistema de adquisición de datos es mucho más eficiente ya que se puede estudiar los datos recolectados y tratar de replicar las condiciones en las que el café alcanza su máxima calidad posible.

3.3. Recomendaciones

- Para procesos de agricultura, donde no exista la posibilidad de tener energía eléctrica de forma constante se recomienda el uso de tarjetas Arduino MRK debido a su alta eficiencia energética.

- Se recomienda el uso de la red The Things Network debido a que es una red gratuita que se crece a base de sus usuarios y es posible compartir algún gateway vinculado a la red TTN.

- Es mucho más sencillo comprar un dispositivo gateway en el mercado y enlazarlo a la red TTN, tratar de construir un dispositivo gateway puede llegar a complicarse por la programación o la poca disponibilidad de la electrónica en el país.

- Al momento de extraer las credencias DevEUI y AppKey es recomendable no compartir estos parámetros ya que son únicos de cada dispositivo y pueden ser registrados y bloqueados por otras personas si tiene acceso a estas.

- Antes de vincular la red TTN con alguna aplicación IoT disponible se recomienda revisar si esta aplicación necesita de una decodificación diferente a la implementada en la red, para esto es necesario visitar la página web de la aplicación IoT e informarse de manera previa.

- La batería del sistema de adquisición de datos se puede cambiar por una de mayor capacidad, revisando el datasheet de la tarjeta Arduino, para así obtener mayor tiempo de uso sin necesidad de ser recargado.

4. REFERENCIAS BIBLIOGRÁFICAS

[1] Fórum Cultural del Café, «El Café en Ecuador,» *fórumcafé,* pp. 6-8, 2020.

[2] D. Sehrawat y N. S. Gill, «Smart Sensors: Analysis of Different Types of IoT Sensors,» *International Conference on Trends in Electronics and Informatics,* vol. 3rd, nº doi: 10.1109/ICOEI.2019.8862778, pp. 523-528, 2019.

[3] Aosong Electronics Co.,Ltd, «Digital-output relative humidity & temperature sensor/module AM2303,» Aosong Electronics Co.,Ltd, Guangzhou, 2010.

[4] A. M. Sánchez, T. Vayas, F. Mayorga y C. Freire , «Analisis del sector cafetero ecuatoriano,» 7 Octubre 2020. [En línea]. Available: https://blogs.cedia.org.ec/obest/wp-content/uploads/sites/7/2020/10/Analisis-del-sector-cafetero-ecuatoriano-final-tres.pdf. [Último acceso: 17 Marzo 2022].

[5] A. Prada, C. Vela, G. Bardález y J. Saavedra, «Efectividad de un Proceso de Secado de Café usando Secadores Solares con Sistema de Flujo de Aire Continuo Impulsado por Energía Fotovoltaica, en la Región San Martín, Perú,» *SciELO,* vol. 30 , nº 6, pp. 85-92, 2019.

[6] D. Cruz, E. López de León, L. F. Pascual y M. Battaglia, «Guía técnica de construcción y funcionamiento de secadoras solares tipo domo,» *Journal of Agriculture and Environment for International Development,* vol. 104, nº 3 - 4, pp. 125 - 138, 210.

[7] Cultura Cafeina, «Cultura Cafeina,» 18 Junio 2020. [En línea]. Available: https://culturacafeina.com/produccion-del-cafe/tipos-de-secado-de-cafe/. [Último acceso: 21 Marzo 2022].

[8] FEDERACIÓN NACIONAL DE CAFETEROS; CENTRO NACIONAL DE INVESTIGACIONES DE CAFÉ, «Cenicafe.org,» 2004. [En línea]. Available: https://www.cenicafe.org/es/publications/cartilla_21._Secado_del_cafe.pdf. [Último acceso: 25 Marzo 2022].

[9] Apacionados por el café, «Porque la importancia del secado en el café,» 23 Agosto 2019. [En línea]. Available: https://www.apasionadosporelcafe.com/blog/la-importancia-del-secado-cafe/. [Último acceso: 26 Mayo 2022].

[10] D-Robotics UK, «DHT11 Humidity & Temperature Sensor,» D-Robotics, Londres, 2010.

[11] Robótica Chile, «roboticatecnociencia.cl,» 2021. [En línea]. Available: https://roboticatecnociencia.cl/producto/sensor-temperatura-dht22-am2302-cod-it20008-cn/?gclid=CjwKCAjwoMSWBhAdEiwAVJ2ndgw6LZjyzzEqrlGUsiPS_DX_YzF5FqIwO7exyu89NGEJjDBvXu5xTBoC2kIQAvD_BwE. [Último acceso: 28 Mayo 2022].

[12] Arduino, «What is Arduino?,» 05 Febrero 2018. [En línea]. Available: https://www.arduino.cc/en/Guide/Introduction. [Último acceso: 29 Mayo 2022].

[13] Arduino, «New MKR WAN 1310 for LoRa connectivity comes with 2MByte Flash and extended battery life,» 10 Octubre 2019. [En línea]. Available: https://blog.arduino.cc/2019/10/10/new-mkr-wan-1310-for-lora-connectivity-comes-with-2mbyte-flash-and-extended-battery-life/?queryID=undefined. [Último acceso: 29 Mayo 2022].

[14] Arduino, «Arduino MKR WAN 1310,» 10 Octubre 2019. [En línea]. Available: https://store-usa.arduino.cc/products/arduino-mkr-wan-1310. [Último acceso: 29 Mayo 2022].

[15] P. Gokhale, O. Bhat y S. Bhat, «Introduction to IOT,» *International Advanced Research Journal in Science, Engineering and Technology,* vol. 5, nº 1, pp. 41-44, 2018.

[16] M. U. Farooq, M. Waseem, S. Mazhar, A. Khairi y T. Kamal, «A Review on Internet of Things,» *International Journal of Computer Applications,* vol. 113, nº 1, pp. 1-7, 2015.

[17] D.-H. Kim, E.-K. Lee y J. Kim, «Experiencing LoRa Network Establishment on a Smart Energy Campus Testbed,» *Sustainability,* vol. 11, nº 7, 30 Marzo 2019.

[18] S. Saade, M. Lopez, L. Penalver, E. Volentini, C. Albaca, F. Lutz y J. Bilbao, «Prototipo de Arquitectura de IoT,» Tucumán, 2018.

[19] LoRa Alliance®, «LoRa Alliance,» 19 Febrero 2021. [En línea]. Available: https://lora-alliance.org/about-lorawan/. [Último acceso: 2 Junio 2022].

[20] Semtech Corporation, «LoRa® and LoRaWAN®:A Technical Overview,» Camarillo, 2019.

[21] Murata Investment Co., Ltd., «Sub-G Module Data Sheet,» Murata (China), China, 2018.

[22] Microchip Technology Inc., «ATECC508A CryptoAuthentication Device Complete Data Sheet,» Microchip (EU), California, 2017.

[23] Aosong Electronics Co.,Ltd., «Temperature and humidity module AM2302 Product Manual,» Aosong (Guangzhou), Guangzhou, Ch, 2019.

[24] Aosong Electronics Co.,Ltd, «Aosong,» 2019. [En línea]. Available: http://www.aosong.com/userfiles/images/product/AM2302C/AM2302C%20%E4%B8%BB%E5%9B%BE%20(3).jpg. [Último acceso: 5 Junio 2022].

[25] Ultralife Corporation, «UBP103450/PCM Technical Datasheet,» Ultralife EU, Newark, New York, 2016.

[26] Aprendiendo Arduino, «Aprendiendo Arduino,» 7 Marzo 2018. [En línea]. Available: https://www.aprendiendoarduino.com/tag/lora/. [Último acceso: 6 Junio 2022].

[27] Arduino, «Arduino downloads software,» [En línea]. Available: https://www.arduino.cc/en/software. [Último acceso: 8 Junio 2022].

[28] The Things Network, «The Things Network,» 17 Mayo 2021. [En línea]. Available: https://www.thethingsnetwork.org/#workbench. [Último acceso: 9 Junio 2022].

[29] The Things Network, «The Things Network Register,» [En línea]. Available: https://account.thethingsnetwork.org/register. [Último acceso: 9 Junio 2022].

[30] The Things Stack, «Cloud Integrations,» [En línea]. Available: https://www.thethingsindustries.com/docs/integrations/cloud-integrations/. [Último acceso: 19 Julio 2022].

[31] E. P. Moncayo Solarte y S. A. Sánchez Reyes, «Diseño de un controlador de humedad y temperatura para un secador de cafe,» Universidad del Cauca, Popayan, 2010.

[32] M. A. Mejia Orozco y J. J. Bedoya Yepes, «Estudio de mercado para comercialización de secador solar parabólico optimizado usado en la producción de café pergamino seco en el departamento de Risaralda.,» Universidad tecnológica de Pereira, Pereira, 2016.

[33] 2CI Group, «Conceptos de actualidad: LoRa y LoRaWan,» 22 Marzo 2021. [En línea]. Available: https://www.2cigroup.com/es/conceptos-de-actualidad-lora-y-lorawan/#:~:text=Por%20lo%20tanto%2C%20LoRaWAN%C2%AE,FSK%20fijo%20de%2050%20Kbps.. [Último acceso: 15 Junio 2022].

5. ANEXOS

ANEXO I. Código de adquisición de datos.

ANEXO II. Código de obtención DevEUI.

ANEXO III. Código comunicación del nodo final con la red TTN.

ANEXO IV. Código de decodificación en JavaScript.

ANEXO V. Código de decodificación para Datacake.

ANEXO I

```
//                    ESCUELA POLITÉCNICA NACIONAL                    //
//                  Facultad de Ingeniería Eléctrica y Electrónica    //
//                                                                    //
//    Autor: Sebástian Alexander Sánchez Macas                        //
//    Director: Dr. Geovanny Danilo Chávez                            //
//    Tema: Diseño y simulación del sistema de adquisición de datos en IoT con  //
//          LoRaWAN para el proceso de secado del café.               //
//                                                                    //
//    Adquisición de datos del sensor DHT22                           //


#include "DHT.h"

#define DHTTYPE DHT22   // DHT 22  (AM2302), AM2321

const int DHTPin = 7;     // definimos el pin digital al que va conectado el sensor

DHT dht(DHTPin, DHTTYPE);

void setup() {
  Serial.begin(9600); // Puerto serial a 9600 bps
  Serial.println("Iniciando...."); //Mensaje de Inicio

  dht.begin(); // Iniciamos el sensor
}

void loop() {

  delay(2000); // Esperamos 2 segundos para la medicion de datos.

  float h = dht.readHumidity(); // Obtenemos el dato de humedad
  float t = dht.readTemperature(); // Obtenemos el dato de temperatura

  if (isnan(h) || isnan(t)) {
    Serial.println("Error de lectura en el sensor DHT!"); // Mensaje de error de lectura
    return;
  }
  Serial.print("Humedad: "); //Imprimimos el dato de humedad
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperatura: "); //Imprimimos el dato de temperatura
  Serial.print(t);
  Serial.print(" *C ");
}
```

ANEXO II

```
//                    ESCUELA POLITÉCNICA NACIONAL                    //
//                  Facultad de Ingeniería Eléctrica y Electrónica    //
//                                                                    //
//    Autor: Sebástian Alexander Sánchez Macas                        //
//    Director: Dr. Geovanny Danilo Chávez                            //
//    Tema: Diseño y simulación del sistema de adquisición de datos en IoT con  //
//          LoRaWAN para el proceso de secado del café.               //
//                                                                    //
//    Obtención número DevEUI                                         //


#include <MKRWAN.h>
// Seleccionamos la región (AS923, AU915, EU868, KR920, IN865, US915, US915_HYBRID)
_lora_band region = US915;

LoRaModem modem(Serial1); // Se habilita un puerto serial para el modem LoRa

void setup() {
  Serial.begin(115200);
  while (!Serial);
  if (!modem.begin(region)) {  // Inicializamos el modulo
    Serial.println("Failed to start module"); // Mensaje de error al iniciar el modulo
    while (1) {}
  };
  Serial.print("Your device EUI is: ");
  Serial.println(modem.deviceEUI()); // Imprimimos el numero EUI de la tarjeta Arduino
}

void loop() {
}
```

ANEXO III

```
//                    ESCUELA POLITÉCNICA NACIONAL                    //
//                  Facultad de Ingeniería Eléctrica y Electrónica    //
//                                                                    //
//    Autor: Sebástian Alexander Sánchez Macas                        //
//    Director: Dr. Geovanny Danilo Chávez                            //
//    Tema: Diseño y simulación del sistema de adquisición de datos en IoT con  //
//          LoRaWAN para el proceso de secado del café.               //
//                                                                    //
//    Comunicación del nodo final con la red TTN.                     //


#include <MKRWAN.h>       // Libreria MKR de Arduino
#include <avr/dtostrf.h>  // Libreria para la conversion float a cadena de caracteres
#include <stdlib.h>       // Libreria para la conversion float a cadena de caracteres
#include "DHT.h"          // Libreria del sensor DHT22
#define DHTTYPE DHT22     // Definimos el tipo de sensor / DHT22  (AM2302), AM2321
#include "credenciales_secretas.h" // Archivo de cabecera con las credenciales de TTN
String appEui = DEV_EUI_SECRETO;  // Asignamos las claves a dos objetos de tipo String
String appKey = APP_KEY_SECRETO;  // que se encuentran en el archivo de cabecera

const int DHTPin = 7;     // Definimos el pin digital al que va conectado el sensor

DHT dht(DHTPin, DHTTYPE); // Instancia a la clase DHT, creando el objeto dht
LoRaModem modem; // Instancia a la clase LoRaModem, creando el objeto modem


void setup() {

  Serial.begin(9600); // Inicializa el puerto serie a 9600 baudios:

  if (!modem.begin(US915)) { // Cambiamos la banda a nuestra region(AS923, AU915, EU868,
KR920, IN865, US915, US915_HYBRID)
  Serial.println("Error al inicializar el modulo"); // Mensaje de error con el modulo
  while (1) {}
  };
  Serial.print("Version de tu modulo: "); // Imprimimos la vercion y el DevEUI del nodo final
  Serial.println(modem.version());        // si no existe problemas con la banda de la region
  Serial.print("Tu device EUI es: ");
  Serial.println(modem.deviceEUI());

  int connected = modem.joinOTAA(appEui, appKey); // Nos conectamos con las credenciales a
TTN
  if (!connected) {
    Serial.println("Ocurrio un error en la conexion a TTN"); // Mensaje de error cuando las
credenciales fallan
    while (1) {}
  }
  dht.begin(); // Iniciamos el sensor
  modem.minPollInterval(60); // Establece el intervalo de sondeo a 60 secs.

}

void loop() {
```

```
  delay(60000); // Esperamos 60 segundos para la medicion de datos.

  float h = dht.readHumidity(); // Obtenemos el dato de humedad
  float t = dht.readTemperature(); // Obtenemos el dato de temperatura

  if (isnan(h) || isnan(t)) {
     Serial.println("Error de lectura en el sensor DHT!"); // Mensaje de error de lectura
     return;
   }

  Serial.print("Humedad: "); //Imprimimos el dato de humedad
  Serial.print(h);
  Serial.print(" %\t");
  Serial.println();
  Serial.print("Temperatura: "); //Imprimimos el dato de temperatura
  Serial.print(t);
  Serial.print(" *C ");
  Serial.println();

  char humistring[20]; //Convierte las variable t y h de tipo float a tipo char array de caracteres
  dtostrf(h,3,1,humistring); //el 3 es igual al número de digitos totales y el 1 el numero de
decimales
  char tempstring[20];
  dtostrf(t,3,1,tempstring);

  //Creamos el objeto humedad y temperatura de la clase String y la pasamos
  //al array humistring y tempstring como constructor.
  String humedad(humistring);
  String temperatura(tempstring);
  //String datos_unidos(humistring); // String usado para la concatenacion de datos
  Serial.print("Convirtiendo Humedad y Temperatura a hexadecimal... ");
  Serial.println();
  //Muestra en la terminal los bytes de la humedad en hexadecimal que se van a enviar
  Serial.print("Enviando Humedad: " + humedad + " - ");
  for (unsigned int i = 0; i < humedad.length(); i++) {
    Serial.print(humedad[i] >> 4, HEX);
    Serial.print(humedad[i] & 0xF, HEX);
    Serial.print(" ");
  }
    Serial.println();

  //Muestra en la terminal los bytes de la temperatura en hexadecimal que se van a enviar
  Serial.print("Enviando Temperatura: " + temperatura + " - ");
  for (unsigned int j = 0; j < temperatura.length(); j++) {
    Serial.print(temperatura[j] >> 4, HEX);
    Serial.print(temperatura[j] & 0xF, HEX);
    Serial.print(" ");
  }

  Serial.println();

  int err; // Variable declarada para errores producidos en la transmisión

  modem.beginPacket(); //Inicia el envío de paquetes

  // Paquete de datos individuales:
```

```
  modem.print(humedad); // Envío la humedad a TTN
  modem.print(temperatura); // Envío la temperatura a TTN
  ////////////////////////////////////////////////////

  //  Paquete de datos unidos:
  //datos_unidos.concat(temperatura); // Concateno los datos de humedad y temperatura
  //modem.print(datos_unidos);  // Envio el dato unificado
  ////////////////////////////////////////////////////////////////////////////

  err = modem.endPacket(true);  //Finaliza el envío de paquetes
  if (err > 0) {
    Serial.println("Mensaje enviado correctamente!"); // Mensaje de correcto envio
  } else {
    Serial.println("Error al enviar el mensaje :(");  // Mensaje de error en el envio
    Serial.println("(puede enviar una cantidad limitada de mensajes por minuto, dependiendo de
la intensidad de la señal");
    Serial.println("Ese tiempo puede variar de 1 mensaje cada dos segundos a 1 mensaje cada
minuto)");
  }
  delay(1000); // Espera de 1 segundo para recibir un mensaje desde la pagina de TTN al nodo
final
  if (!modem.available()) {
    Serial.println("No se recibió ningún mensaje de enlace descendente (downlink) en este
momento.");
    return;
  }
  char rcv[64]; // Variable para almecenar el mensaje de TTN
  int i = 0;
  while (modem.available()) {
    rcv[i++] = (char)modem.read();
  }
  Serial.print("Recibido: "); // Imprimo en la terminal el mensaje de TTN
  for (unsigned int j = 0; j < i; j++) {
    Serial.print(rcv[j] >> 4, HEX);
    Serial.print(rcv[j] & 0xF, HEX);
    Serial.print(" ");
  }
  Serial.println();
}
```

ANEXO IV

```
//                    ESCUELA POLITÉCNICA NACIONAL                    //
//                Facultad de Ingeniería Eléctrica y Electrónica      //
//                                                                    //
//    Autor: Sebástian Alexander Sánchez Macas                        //
//    Director: Dr. Geovanny Danilo Chávez                            //
//    Tema: Diseño y simulación del sistema de adquisición de datos en IoT con //
//          LoRaWAN para el proceso de secado del café.               //
//                                                                    //
//    Decodificación de datos.                                        //


function Decoder(bytes, port) {
  var dato={};

  var hum_centenas=(bytes[0]-0x30)*100;
  var hum_decenas=(bytes[1]-0x30)*10;
  var hum_unidad=(bytes)[3]-0x30;

  var temp_centenas=(bytes[4]-0x30)*100;
  var temp_decenas=(bytes[5]-0x30)*10;
  var temp_unidad=(bytes)[7]-0x30;

  dato.Humedad=(hum_centenas+hum_decenas+hum_unidad)/10;
  dato.Temperatura=(temp_centenas+temp_decenas+temp_unidad)/10;


    return dato;

}
```

ANEXO V

```
//                    ESCUELA POLITÉCNICA NACIONAL                    //
//                Facultad de Ingeniería Eléctrica y Electrónica      //
//                                                                    //
//    Autor: Sebástian Alexander Sánchez Macas                        //
//    Director: Dr. Geovanny Danilo Chávez                            //
//    Tema: Diseño y simulación del sistema de adquisición de datos en IoT con //
//            LoRaWAN para el proceso de secado del café.             //
//                                                                    //
//    Decodificación de datos DataCake.                               //


function Decoder(bytes, port) {
  var dato={};

  var hum_centenas=(bytes[0]-0x30)*100;
  var hum_decenas=(bytes[1]-0x30)*10;
  var hum_unidad=(bytes)[3]-0x30;

  var temp_centenas=(bytes[4]-0x30)*100;
  var temp_decenas=(bytes[5]-0x30)*10;
  var temp_unidad=(bytes)[7]-0x30;

  dato.Humedad=(hum_centenas+hum_decenas+hum_unidad)/10+"%";
  dato.Temperatura=(temp_centenas+temp_decenas+temp_unidad)/10+"°C";
  dato.Hum=(hum_centenas+hum_decenas+hum_unidad)/10;
  dato.Temp=(temp_centenas+temp_decenas+temp_unidad)/10;


    return dato;
```