

UNIVERSIDAD DE OVIEDO

CENTRO INTERNACIONAL DE POSTGRADO

MASTER EN INGENIERÍA MECATRÓNICA

TRABAJO FIN DE MÁSTER

**DESARROLLO DE UN NODO SENSOR PARA APLICACIONES IOT
(INTERNET OF THINGS)**

JUNIO DE 2019

(Paula Estrada)

(Miguel Ángel José Prieto)

[Firma]

AGRADECIMIENTOS

Agradezco a Ikerlan por brindarme la oportunidad de participar en este proyecto tan interesante, gracias a esta oportunidad pude conocer una nueva cultura de trabajo e investigación que hasta el momento ha sido una de mis metas.

Agradezco también a los coordinadores del máster quienes han hecho posible este trabajo y por supuesto a mis amigos y familiares que siempre estuvieron ahí para brindarme todo su apoyo.

RESUMEN

El presente trabajo de fin de Máster realizado durante el periodo de seis meses en IKERLAN, en conjunto con la Universidad de Oviedo, tiene como objetivo presentar el diseño de un nodo sensor inalámbrico que envía información utilizando la tecnología LoRa (por sus cifras en inglés Long Range). El nodo sensor diseñado para este proyecto monitorea parámetros de humedad y temperatura utilizando un sensor para la medición de éstos y un acelerómetro para determinar la posición del nodo, estos datos son enviados a través de un transceptor que trabaja con la tecnología LoRa en la banda de 868 MHz. El procesamiento de estos datos y el control de las funcionalidades de todos los componentes del nodo sensor son realizadas por un microcontrolador ARM de 32 bits.

A grandes rasgos LoRa es una tecnología surgida con el auge que ha tenido la monitorización de diferentes parámetros en el IoT (Internet de las cosas) y la necesidad de almacenamiento y gestión adecuados de estos datos.

Ahora bien, LoRaWAN es un protocolo de comunicación de red que usa la tecnología LoRa para comunicar y administrar dispositivos, esta tecnología es de las más utilizadas para la monitorización de sistemas embebidos debido a que es una tecnología “open source” y además cuenta con características de bajo consumo de energía. Una red LoRaWAN consiste principalmente en tres componentes: dispositivos finales (nodos), gateways (puerta de enlace) y un servidor de red.

El dispositivo final o nodo se comunica con el Gateway utilizando el protocolo de comunicación LoRaWAN y el Gateway reenvía la información enviada por el dispositivo final al servidor de red. El servidor de red utilizado para este proyecto es una plataforma de uso libre llamada “The Things Network” dedicada al fomento y desarrollo de la tecnología LoRa.

El nodo diseñado cuenta con una carcasa de capacidad IP67, que es el grado de protección que se utiliza para clasificar los diferentes contenedores que resguardan los componentes que constituyen un equipo expuesto a condiciones no ideales en el exterior, el diseño mecánico de esta carcasa se llevó a cabo utilizando el software Solid Works y su impresión 3D se realizó utilizando el método de estereolitografía.

Por último, un aspecto importante en el desarrollo de este proyecto fue asegurarse de tener el nivel de energía óptimo para controlar el consumo de energía del nodo. Uno de los mayores retos de los proyectos desarrollados para IoT es la energía consumida por los dispositivos, durante el desarrollo de este proyecto se buscó la solución más óptima para la alimentación del nodo, esto se logró en conjunto con la programación de un código que cuenta con las características necesarias para optimizar el consumo de batería del dispositivo, y utilizando por lo general modo “sleep” la descarga de la batería del nodo puede prolongarse hasta por 10 años.

PALABRAS CLAVE

Nodo - LoRa- Gateway - Servidor – IP67 – Frecuencia –LoRaWAN-Latencia-RSSI-SNR

ÍNDICE GENERAL

1.	INTRODUCCIÓN	1
1.1	INTERNET DE LAS COSAS (IoT)	1
1.2	LPWANs	2
2.	ESTADO DEL ARTE	5
2.1	BREVE HISTORIA DE LOS NODOS SENSORES	5
2.2	NODO SENSOR	5
2.2.1	ARQUITECTURA DE UN NODO SENSOR	6
2.2.1.1	MICROCONTROLADOR	6
2.2.1.2	TRANCEPTOR	6
2.2.1.3	FUENTE DE ALIMENTACIÓN	6
2.2.1.4	SENSORES	7
2.2.1.5	ANTENA	7
2.2.1.5.1	PARÁMETROS DE LAS ANTENAS	8
2.2.1.5.1.1	DIAGRAMA DE PARÁMETROS DE RADIACIÓN DE LAS ANTENAS	8
2.3	TECNOLOGÍAS DE COMUNICACIÓN DE LOS NODOS SENSORES	9
2.3.1.	TECNOLOGÍA DE COMUNICACIÓN LORA	9
2.3.1.1	PROPIEDADES IMPORTANTES DE LA MODULACIÓN DE LA TECNOLOGÍA LORA	10
2.3.2	PROTOCOLO DE COMUNICACIÓN LORAWAN	11
2.3.2.1	ARQUITECTURA DE UNA RED LORAWAN	13
2.3.3	CLASIFICACIÓN DE LOS DISPOSITIVOS LORA	14
2.3.3.1	CLASE A	14
2.3.3.2	CLASE B	15
2.3.3.3	CLASE C	15
2.4	SEGURIDAD EN LORAWAN	15
2.5	APLICACIONES LORA	17
2.5.1	DESARROLLOS ACTUALES CON LORA	18
2.5.1.1	MONITOREO INDUSTRIAL DE TEMPERATURA	18
2.5.1.2	MONITOREO INALÁMBRICO DE NIVELES DE GAS	19
2.5.1.2.1	CASO DE USO REAL MONITOREO INALÁMBRICO DE GAS	20
2.5.1.3	IoT-LORA ESTACIONAMIENTO INTELIGENTE	21
2.5.1.3.1	CASO DE USO REAL IoT-LORA ESTACIONAMIENTO INTELIGENTE	22

2.6	GATEWAY	23
2.7	SERVIDOR THE THINGS NETWORK	23
2.8	DIAGRAMA DE GANTT DE PLANEACIÓN DE ACTIVIDADES DEL PROYECTO	24
3	ESTRUCTURA Y DESCRIPCIÓN DEL NODO SENSOR (HARDWARE)	25
3.1	TARJETA MINI ULTRA PRO	25
3.1.1	MICROCONTROLADOR ATSAM21G18A-AU	26
3.1.2	TRANCEPTOR MÓDULO LORA RFM95-W	26
3.2	SENSORES	27
3.2.1	SENSOR DE TEMPERATURA DHT22	27
3.2.1.1	SECUENCIA Y CLASIFICACIÓN DE PINES (ORDEN ASCENDENTE)	27
3.2.1.2	RANGO DE MEDICIONES DE TEMPERATURA	28
3.2.2	ACELERÓMETRO ADXL355	28
3.3	ANTENAS PARA EL NODO SENSOR	28
3.4	BATERÍA	29
3.5	DESCRIPCIÓN DE GATEWAYS	30
3.5.1	GATEWAY MULTITECH (MULTICANAL)	30
3.5.2	NANO GATEWAY LOPY (MONO CANAL)	32
3.5.2.1	ANTENA DE GATEWAYS	32
4	CONFIGURACIÓN DEL NOD SENSOR (SOFTWARE)	33
4.1	CONFIGURACIÓN DEL NODO EN LA NUBE THE THINGS NETWORK	33
4.2	ACTIVACIÓN DEL NODO EN LA NUBE	35
4.3	CONFIGURACIÓN GENERAL DEL MULTITECH CONDUIT	35
4.4	CONFIGURACIÓN DEL PACKET FORWARDER PARA EL GATEWAY MULTITECH	40
4.5	CONFIGURACIÓN DEL NANO GATEWAY LOPY	42
4.6	CONFIGURACIÓN DEL ORTAL DWEET.IO	43
4.7	CONFIGURACIÓN DE LA APLICACIÓN FREEBOARD	44
4.8	PRUEBAS DE ALCANCE DENTRO DE IKERLAN	44
4.8.1	PRUEBAS DE ALCANCE CON ANTENA DE HILO	46
4.8.2	PRUEBAS DE ALCANCE CON ANTENA HELICOIDAL	48
5	DISEÑO MECÁNICO DE LA CARCASA	49
5.1	GRADO DE PROTECCIÓN IP67	49
5.2	SELECCIÓN DE LA JUNTA TÓRICA (TOROIDE)	51
5.3	DISEÑO DEL MOLDE	52
5.3.1	ESTANQUEIZACION ESTÁTICA, COMPRESIÓN RADIAL	52
5.4	IMPRESIÓN 3D DE LA CARCASA	52
5.4.1	PROCESO DE ESTEREOLITOGRAFÍA	52
5.4.2	POSPROCESAMIENTO DE LA CARCASA	54

DESARROLLO DE UN NODO SENSOR PARA APLICACIONES IOT (INTERNET OF THINGS)

LINEAS FUTURAS Y MEJORAS	57
CONCLUSIONES	57
BIBLIOGRAFÍA	59
ANEXO I	61
ANEXO II	65
ANEXO III	81
ANEXO IV	89

1. INTRODUCCIÓN

La finalidad de este trabajo de fin de master es diseñar y construir un nodo sensor inalámbrico orientado a aplicaciones IoT que envíe parámetros de humedad y temperatura mediante el protocolo de comunicación LoRaWAN a un servidor o nube de uso público.

Ikerlan ya cuenta con una red LPWAN que utiliza las tecnologías LoRa, Sigfox y NB-IoT que utilizan para la monitorización de distintos parámetros, principalmente temperatura, humedad y luz de las instalaciones y laboratorios. Los nodos sensores con los que ya se cuenta son conocidos como mDot de la marca Multitech que están compuestos de una antena omnidireccional de 15 cm de longitud, que envían datos a un Gateway encargado de enviar datos a un servidor o nube privado. Estos datos son gestionados por una interfaz que permite visualizar y almacenar los datos enviados por el sensor.

Mantener una red con estas tecnologías representa mayor costo, además de que al ser un servidor de uso privado la red no puede crecer o mejorarse de una forma más sencilla, es por esto que se propuso elaborar una nueva red LPWAN que basándose únicamente en la tecnología LoRa, pueda enviar datos a un servidor público. Para la consecución de estos requerimientos se han marcado los siguientes objetivos:

- Diseño del sistema embebido (Hardware y software). Esto incluye la integración de tres sensores: Sensor de temperatura, sensor de humedad y sensor de posición.
- Diseño y fabricación 3D de una carcasa para el nodo con grado de protección IP-67.
- Diseño de una antena de largo alcance que permita la transmisión de datos con las dimensiones adecuadas para que pueda transmitir con los alcances establecidos por LoRa (15 km a la redonda).
- Validación sobre la red real (Disponible en las instalaciones de Ikerlan).

1.1 Internet de las Cosas (IoT).

Internet of Things (IoT) es un sistema de dispositivos, máquinas, objetos, animales o personas interconectadas que son capaces de transmitir datos con la mínima interacción humana.

El objetivo de IoT es aportar grandes beneficios en diversas áreas como son: Smart Cities, Smart Buildings, medicina, agricultura, industria, domótica, entre otras.

Hay varios factores que se deben tener en cuenta al desarrollar una aplicación IoT, como consumo, tamaño y coste de los dispositivos, latencia escalabilidad y seguridad. Dependiendo de la aplicación que se realice puede haber requerimientos en el tipo de comunicación utilizado para la conexión entre los dispositivos. Puede haber dispositivos que requieran de un gran ancho de banda, otros que necesiten un alto nivel de protección contra interferencias y otros que garanticen la fiabilidad de las transmisiones

Para dar respuesta a los diferentes problemas que pueden suponer las diferentes aplicaciones de IoT, se han desarrollado varias tecnologías inalámbricas que se agrupan de la siguiente manera:

-NFC/RFID: estas dos tecnologías son utilizadas para comunicaciones de muy corto alcance (centímetros) y se utilizan sobre todo para la identificación de objetos.

-LAN: estos protocolos se utilizan para comunicaciones de muy corto alcance (centímetros) y se utilizan sobre todo para la identificación de objetos.

-**Cellular:** estas tecnologías se diferencian por usar una banda de frecuencias con licencia, por lo que las interferencias son menores en comparación con otras tecnologías. La cobertura en estas tecnologías está muy extendida, pero es necesario pagar por los datos enviados. Dentro de estas tecnologías se encuentran diferentes opciones como son 2G, 3G o LTE.

-**LPWAN:** este grupo de tecnologías se diferencian por tener un consumo muy bajo de los dispositivos y transmitir con una velocidad muy pequeña, para aumentar el alcance de las comunicaciones. Dentro de este grupo se incluyen LoRa, Sigfox y NB-IoT.

1.2 LPWANs.

Por sus siglas en inglés Low Power Wide Area Network es una tecnología para la interconexión de dispositivos de bajo data rate (tasa de transmisión de datos), para la optimización del consumo energético y el alcance de las comunicaciones.

Esta tecnología es adecuada para conectar dispositivos con necesidad de transmitir pocos datos a una larga distancia, manteniendo una larga duración de batería.

Las principales características de las redes LPWAN son las siguientes:

- Bajo consumo: los dispositivos desarrollados para las redes LPWAN deben de tener un consumo de energía muy bajo, para alargar lo máximo posible la duración de las baterías.
- Largo alcance: el alcance de estas tecnologías debe poder cubrir amplias áreas de hasta 30 km en promedio.
- Bajo coste: el bajo coste de los dispositivos es esencial, ya que al abarcar áreas tan extensas pueden colocarse miles de dispositivos, que concierte en crítico el precio de cada uno de éstos.

Varias tecnologías LPWAN que han surgido para dar solución a los diferentes problemas propuestos. Entre ellas DASH7, Ingenu, Sigfox, NB-IoT y LoRa. De todas estas tecnologías se toman en cuenta tres principales que son las que hasta ahora se han utilizado en Ikerlan:

Tecnología	LoRa	Sigfox	NB-IoT
Frecuencia	Banda no licenciada de rango (863-870 MHz)	Banda no licenciada (868 MHz)	Banda licenciada LTE
Ancho de banda de canal	250 kHz	200 Hz	180 kHz
Tasa de transferencia (data rate)	50 kbps	100 bps	170 kbps
Potencia de transmisión	14 dBm	14 dBm	23 dBm
Alcance	15 km	30 km	15 km

Tabla 1 Comparativa de tecnologías LPWANs

Las tecnologías Sigfox y NB-IoT como se ha mencionado antes, no permiten la creación de redes públicas, ya que las estaciones base cuestan varios miles de euros y pertenecen a compañías telefónicas, en el caso de NB-IoT, y a la propia compañía Sigfox, que tiene control sobre la red y cobran por su uso. En cambio, la tecnología LoRa permite obtener un Gateway por un precio económico (entre 100 y 200

euros), con esto se puede ser el propietario de la red, dar acceso público a ella, agregar la cantidad de dispositivos o nodos de una manera libre y económica.

Teniendo todos estos factores en cuenta, y los objetivos de este trabajo se utilizará la tecnología LoRa.

Como ventajas al utilizar esta tecnología está el no necesitar estaciones base de terceros, por lo que Ikerlan podrá gestionar su propia red con nodos sensores diseñados para trabajar bajo las condiciones establecidas por este protocolo de comunicación.

2. ESTADO DEL ARTE

2.1. Breve historia de los nodos sensores

Hoy en día los nodos sensores son dispositivos que se utilizan en el sector industrial, la automatización, la salud, el tráfico y en muchas áreas de consumo.

Su estudio y desarrollo se remonta a los años 80 en aplicaciones de carácter militar por la Agencia de Proyectos de Investigación Avanzada de la Defensa de los Estados Unidos (DARPA), quien llevó a cabo un programa de redes de nodos sensores distribuidos para el ejército especialmente en las zonas de conflicto. Los nodos sensores que la DARPA desarrolló eran bastante grandes (del tamaño de una caja de zapatos o más grande) por lo que el número de aplicaciones era bastante limitada. Además de que los primeros nodos no estaban asociados a la conectividad inalámbrica en absoluto. Estas primeras redes de sensores tan sólo destacaban por sus fines militares y aún no satisfacían algunos requisitos de gran importancia en este tipo de redes tales como la autonomía y el tamaño.

Para la década de los 90 nuevamente la DARPA genera un nuevo proyecto enfocado a los nodos sensores llamado SenseIt, cuyo objetivo era mejorar aspectos relacionados con la velocidad de adaptación de los sensores en ambientes cambiantes y en cómo hacer que la información que era recopilada por los sensores fuera de alguna forma más confiable.

Para finales de los años 90 y principios de nuestro siglo los nodos sensores comenzaron a tomar mayor relevancia en el ámbito civil, haciéndose cada vez más pequeños, su tamaño ha evolucionado de ser comparable a una caja de zapatos, un paquete de tarjetas hasta llegar al polvo partícula (dust particule). Además de la reducción de su tamaño en la actualidad son mucho más baratos en precio.

Los avances recientes en las comunicaciones, la micro-electrónica y la computación han resultado en la generación de nuevas investigaciones de nodos sensores, la cual se centra en tecnología y procesamiento de la información en la red adecuada para entornos altamente dinámicos y nodos sensores de recursos limitados.

Existen dos grandes compañías como Crossbow, quienes han desarrollado nodos sensores del tamaño de una moneda que funcionan con baterías que les permiten tener una mayor autonomía. Así mismo Dust Inc., que a su vez han creado nodos de un tamaño inferior a un guisante. Como resultado de este desarrollo la comercialización y aplicación de nodos sensores se ha acelerado, y muchas nuevas compañías de tecnología encargadas en su desarrollo están surgiendo. Al mismo tiempo del surgimiento de estas compañías nuevos retos para el desarrollo de estos nodos comienzan a presentarse día a día puesto que una de las claves de su buen funcionamiento y autonomía es la energía, ya que los nodos sensores tienen energía limitada, trabajan en largos periodos al exterior, en zonas de difícil acceso y con frecuencia en condiciones ambientales no ideales.

2.2. Nodo Sensor

Un **nodo sensor** es un dispositivo con la capacidad de procesar y reunir información por medio de un sensor, comunicarse con otros nodos o gateways en una red para desempeñar una tarea determinada.

Los nodos sensores son dispositivos autónomos que funcionan con baterías muy similares a las de los teléfonos celulares, además del hecho de que sus comunicaciones se basan en protocolos de bajo consumo, gracias a los cuales pueden ir de un estado de inactividad a realizar una transmisión y luego regresar a su estado inicial evitando el desgaste de energía.

2.2.1 Arquitectura de un nodo sensor.

Los principales componentes de un nodo sensor son: microcontrolador, transceptor, fuente de alimentación y uno o más sensores

2.2.1.1 Microcontrolador

El micro-controlador es utilizado en muchos sistemas como los nodos sensores debido a su bajo coste, flexibilidad para conectar a otros dispositivos, facilidad de programar, y bajo consumo de energía, realiza tareas, procesa datos y controla la funcionalidad de los componentes en el nodo sensor.

2.2.1.2 Transceptor

El transceptor se encarga de generar una señal de radiofrecuencia a partir de una de banda base digital y viceversa. Tiene las funcionalidades de un auricular y un transmisor combinadas en un solo dispositivo que es el transceptor, este dispositivo convierte las señales digitales de los sensores en una señal de radio analógica modulándola en una portadora de RF y enviándolas al microcontrolador.

La placa del nodo debe integrar los transceptores necesarios de acuerdo a las bandas de frecuencia donde se vaya a trabajar en este caso la de 868 MHz (Europa). Las bandas de frecuencia usadas por LoRa corresponden a las ISM, el cual otorga señales de radio gratuita, asignando un espectro y disponibilidad global.

2.2.1.3 Fuente de alimentación

Para alimentar a un nodo se puede usar una batería interna, una fuente de alimentación externa o una combinación. Por ejemplo, en los nodos sensores más comunes, el diseño puede tener una pequeña celda solar conectada a una batería recargable. Dado que uno de los objetivos clave de la tecnología LPWAN es utilizar sensores que puedan funcionar con la misma batería durante años, es crucial estimar la duración de la batería para el diseño de un nodo sensor en particular.

Hay cinco modos principales de consumo de energía del HW que deben determinarse:

- **Apagado/sleep:** todos los componentes del nodo se encuentran apagados o en modo sleep.
- **IDLE:** los sensores, y otros elementos electrónicos están apagados a excepción del microcontrolador.
- **Trabajo:** el nodo sensor está en modo operacional (no hay transmisión RF)
- **LoRa TX:** el nodo sensor transmite datos a través del transceptor.
- **LoRa RX:** el nodo sensor recibe datos a través del transceptor (para este proyecto únicamente se usa el modo de transmisión).

De acuerdo al modo en el que se encuentre el nodo sensor, será el nivel de consumo de energía que tenga, para esto es necesario calcular cuánta corriente consumirá el nodo en cada uno de estos modos. Para lograr esto es necesario estimar cuánto tiempo estará el dispositivo en cada uno de estos modos por hora, por día o por semana.

2.2.1.4 Sensores

Los sensores son dispositivos de hardware que producen una respuesta medible a un cambio en una condición física como temperatura, presión o posición. La continua señal analógica producida por los sensores es digitalizada por un convertidor digital a analógico y enviada a controladores para procesamiento adicional.

Entre sus características técnicas principales para el uso de sensores en este proyecto se encuentran:

- Rango de medida: dominio en la magnitud medida en el que puede aplicarse el sensor.
- Precisión: es el error de medida máximo esperado.
- Offset o desviación de cero: valor de la variable de salida cuando la variable de entrada es nula. Si el rango de medida no llega a los valores nulos de la variable de entrada, habitualmente se establece otro punto de referencia para definir el offset.
- Sensibilidad: suponiendo que es de entrada a salida y la variación de la magnitud de entrada.
- Resolución: mínima variación de la magnitud de entrada que puede detectarse a la salida.
- Rapidez de respuesta: puede ser un tiempo fijo o depender de cuánto varíe la magnitud a medir. Depende de la capacidad del sistema para seguir las variaciones de la magnitud de entrada.
- Derivas: son otras magnitudes, aparte de la medida como magnitud de entrada, que influyen en la variable de salida. Por ejemplo, pueden ser condiciones ambientales, como la humedad, la temperatura u otras como el envejecimiento (oxidación, desgaste, etc.) del sensor.
- Repetitividad: error esperado al repetir varias veces la misma medida.

2.2.1.5 Antena

Este es un componente clave para alcanzar la distancia máxima en el enlace de comunicación inalámbrica entre el dispositivo final y la puerta de enlace o Gateway LoRaWAN con el que se conectará el dispositivo. El objetivo de una antena es transformar las señales eléctricas en ondas electromagnéticas de RF, propagándose al espacio libre (modo de transmisión (TX)) y transformar ondas electromagnéticas de RF en señales eléctricas (modo de recepción (RX)). Su tipo, diseño, orientación y posicionamiento puede enlazar o romper el enlace de comunicación RF.

Dentro de los cinco tipos de antenas más comunes para los dispositivos IoT se encuentran:

- Antena de alambre de cuarto de onda simple
- Antena en PBC (ya impresa en la placa del circuito)
- Antena de chip
- Antena helicoidal simple de cuarto de onda
- Antena de látigo de cuarto de onda externa.

El cambio a LoRaWAN requerirá una antena con un cuarto de longitud de onda sintonizada específicamente para el espectro de frecuencia de portadora LoRa de 868 MHz. La Tabla 2 muestra las

ventajas y las desventajas de cada uno de estos tipos de antenas. La elección de la antena finalmente se convertirá en una compensación entre costo, tamaño y rendimiento.

Tipo de antena	Ventajas	Desventajas
Antena de cable	-Extremadamente barata -Buen rendimiento	- Repetitividad del rendimiento de la antena en la fabricación en serie.
Antena en PBC	-Muy bajo costo -Buen rendimiento -Tamaño relativamente pequeño -Antenas de diseño estándar ampliamente disponibles	-Sensible a la calidad del diseño del plano de tierra y la colocación de componentes cercanos. -Requiere un área de PCB relativamente grande.
Antena de chip	-Tamaño pequeño -Muchas opciones diferentes en el mercado	-Costo y rendimiento medios. -Necesitan de circuitos adicionales para su funcionamiento que vayan de acuerdo a las especificaciones.
Antena helicoidal	-Baratas -Buen rendimiento	- Desempeño medio. -Cuidado especial en el montaje en la PCB.
Antena de látigo	-El mejor rendimiento de las antenas. -Ciclo de diseño más corto.	-Precio más elevado. -La ubicación puede ser problemática, especialmente en dispositivos IoT pequeños -Requiere de pruebas de emisiones conducidas.

Tabla 2 Ventajas y desventajas de los tipos de antena analizados

2.2.1.5.1 Parámetros de las antenas

Las antenas se caracterizan por una serie de parámetros, estando los más habituales descritos a continuación:

2.2.1.5.1.1 Diagrama de parámetros de radiación de las antenas

Es la representación gráfica de las características de radiación de una antena en función de la dirección angular. Habitualmente se define el sistema de radiación en coordenadas esféricas, que se utiliza para determinar la posición espacial de un punto P mediante una distancia r y dos ángulos θ , φ . Por lo tanto, el punto P (r , θ , φ) queda representado por un conjunto de tres magnitudes: el radio r, la longitud θ y el azimut φ .

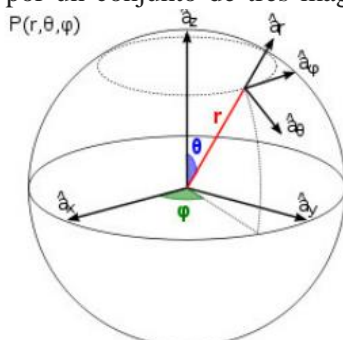


Figura 1. Descripción de las coordenadas esféricas del patrón de radiación de las antenas.

La coordenada r representa la distancia desde el centro de fase del sistema radiante hasta un punto cualquiera, con lo que su dominio varía desde el cero hasta el infinito. La coordenada θ es el ángulo recorrido en el eje z de las coordenadas cartesianas, con lo que su dominio comprende desde $-\pi/2$ hasta $\pi/2$ radianes. La coordenada φ del sistema recorre el ecuador de la diferencia comenzado en el eje X de las coordenadas cartesianas, y por tanto varían desde 0 a 2π radianes.

El diagrama de coordenadas esféricas sirve para representar cualquier magnitud vectorial y son especialmente adecuados para el estudio de los campos radiados por las antenas. Lo más habitual es representar la densidad de potencia radiada, aunque también se pueden encontrar diagramas de polarización o de fase. Dependiendo del tipo de antena utilizada, la energía electromagnética puede o no emitirse, o recibirse, con igual intensidad en todas direcciones.

El tipo de antena a utilizar es una antena de hilo o cable, esta antena se caracteriza por ser una antena isotrópica lo que significa que su emisión y recepción no depende de la dirección.

2.3 Tecnologías de comunicación de los nodos sensores.

Como se mencionó en el capítulo uno, el internet de las cosas utiliza diferentes tecnologías para la comunicación de sus dispositivos, dentro de estas tecnologías las opciones comerciales más conocidas son ZigBee, SigFox, NB-IoT, LTE y LoRa entre otras.

Para este proyecto se seleccionó la tecnología de comunicación inalámbrica LoRa ya que es una solución que cumple con diferentes requerimientos propios de la red inalámbrica que se tiene en Ikerlan.

2.3.1 Tecnología de comunicación LoRa.

LoRa es una tecnología inalámbrica desarrollada y patentada por Semtech que puede definirse como la capa física o modulación inalámbrica utilizada para crear enlaces de comunicación de largo alcance, emplea bandas de radio frecuencia de uso público sin licencia ISM (Industrial, Scientific and Medical) como los utilizados por Bluetooth y Wi-Fi, reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica. Esta tecnología se basa en la modulación de espectro ensanchado la cual aumenta la capacidad del canal de comunicación, aumentando el ancho de banda de la señal a transmitir lo que le permite aumentar significativamente el rango de potencia de comunicación, el alcance de LoRa permite rangos de cobertura de hasta 15 kilómetros.

LoRa mejora significativamente la sensibilidad del receptor, pues al igual que otras técnicas de modulación de espectro ensanchado, utiliza toda la anchura de banda del canal para transmitir una señal, haciéndola robusta al ruido.

Entre sus principales ventajas se encuentra:

- Alta tolerancia a las interferencias
- Alta sensibilidad para recibir datos (-168dB)
- Bajo Consumo de energía (hasta 10 años con una batería)
- Largo alcance 10 a 15km
- Baja transferencia de datos (hasta 255 bytes)

- Conexión punto a punto
- Seguridad en la red

Todo esto hace a la tecnología ideal para conexiones a grandes distancias y para redes de IoT que se pueden utilizar en ciudades inteligentes, lugares con poca cobertura celular o redes privadas de sensores o actuadores. Gracias a su largo alcance y su bajo consumo Lora es una tecnología barata y fácil de implementar. Es por esto que esta tecnología será utilizada para este proyecto.



Figura 2 Tecnología LoRa 1

2.3.1.1 Propiedades de importantes de la modulación de la tecnología LoRa.

La ganancia de procesamiento de LoRa se introduce en el canal de RF al multiplicar la señal de datos por un código de expansión o una secuencia de chips (Bytes). Al aumentar la velocidad de chip, aumentan los componentes de frecuencia del espectro de la señal total. En otras palabras, la energía de la señal total se extiende entre un rango más amplio de frecuencias, lo que permite al receptor discernir una señal con un SNR (Signal Noise Ratio) peor. En términos de LoRa, la cantidad de código de expansión aplicado a la señal de datos original se denomina factor de dispersión (Spreading Factor SF). Hay un total de seis factores de dispersión definidos en la Modulación LoRa: [SF7-SF12].

Una señal modulada con un factor de propagación más grande puede viajar una distancia más larga y aun así ser recibida sin errores por el receptor de RF en comparación con una señal con un factor de dispersión menor.

Spreading Factor (for UL at 125KHz)	Bit Rate	Range (dependent on terrain conditions)	Time on Air for an 11-byte payload
SF10	980 bps	8 km	371 ms
SF9	1760 bps	6 km	185 ms
SF8	3125 bps	4 km	103 ms
SF7	5470 bps	2 km	61 ms

Tabla 3 Factores de dispersión para LoRa

La Tabla 3 muestra los cuatro factores de dispersión diferentes [SF7-SF10] que se pueden utilizar para los mensajes de enlace ascendente (UL) en un canal de 125 MHz (los mensajes de enlace descendente utilizan canales de 500 KHz que pueden usar los seis factores de dispersión disponibles). Así como el rango estimado (esto depende del terreno; se lograrán distancias más largas en un entorno rural en comparación con un entorno urbano). También muestra los valores de tiempo en el aire (TOA) para un payload (llamando así a los datos transmitidos o carga útil, que es en realidad el mensaje enviado) para cada uno de los cuatro factores de dispersión.

Una propiedad inherente de los factores de dispersión de la modulación LoRa es que son ortogonales. Esto significa que las señales moduladas con diferentes factores de dispersión y transmitidas en el mismo canal de frecuencia al mismo tiempo no interfieren entre sí. Las señales LoRa son muy resistentes a los mecanismos de interferencia tanto en banda como fuera de banda.

La modulación LoRa también ofrece inmunidad a multitrayecto y desvanecimiento, por lo que es ideal para su uso en entornos urbanos y suburbanos, donde dominan ambos mecanismos. Los desplazamientos Doppler causan un pequeño cambio de frecuencia en el pulso de LoRa que introduce un cambio relativamente despreciable en el eje de tiempo de la señal de banda de base. Esta tolerancia de desplazamiento de frecuencia mitiga el requisito de fuentes de reloj de referencia de tolerancia ajustada y, por lo tanto, hace que LoRa sea ideal para las comunicaciones de datos desde dispositivos/ sensores que son móviles.

2.3.2 Protocolo de comunicación LoRaWAN.

LoRaWAN es un protocolo de comunicación que usa la tecnología LoRa para comunicar y administrar dispositivos LoRa, este protocolo está específicamente diseñado para aplicaciones de sensores de bajo consumo de energía y largo alcance.

Este protocolo de comunicación proporciona la habilidad de crear redes en estrella de alta capacidad (más de dos dispositivos) y enlaces punto a punto entre dos dispositivos, así también existen varios tipos de redes:

- 1) Una red real donde los nodos finales o dispositivos IoT pueden conectarse enviando una solicitud de activación aérea (OTAA) a una o más puertas de enlace (Gateways) LoRaWAN.
- 2) Una red donde hay una entidad central en la nube que gestiona la red y optimiza duración de la batería del nodo final individualmente haciendo intercambios entre la potencia de transmisión del nodo final y los factores de dispersión.

Una red LoRaWAN consta normalmente de tres componentes: dispositivos finales (nodos), gateways (puerta de enlace) y un servidor de red (en este caso la nube). Después todos los datos involucrados son presentado y visualizados por algún programa de gestión y visualización de datos.

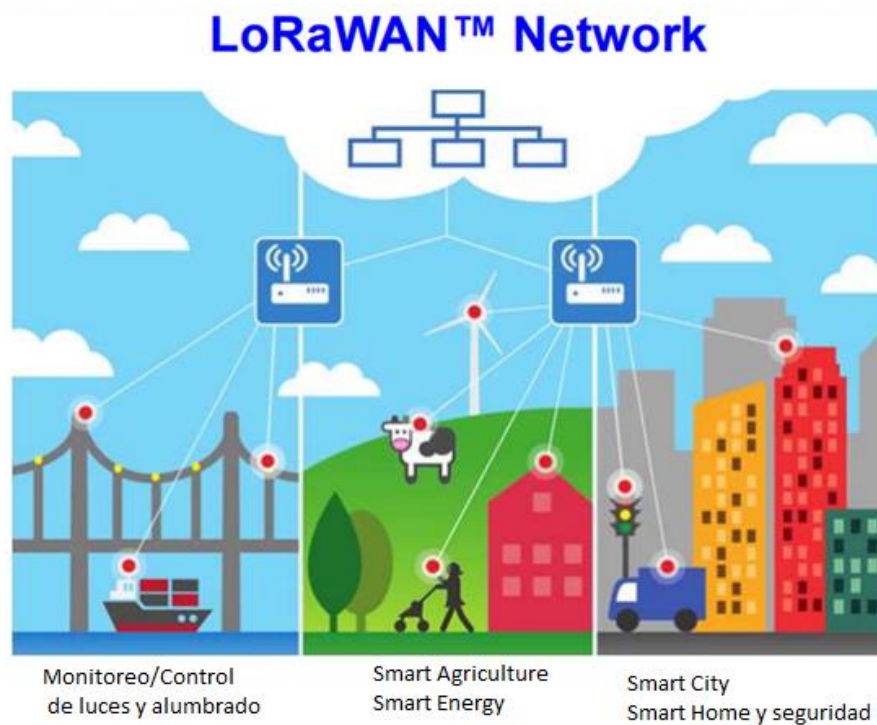


Figura 3. Protocolo de comunicación LoRaWAN

LoRaWAN define el control de acceso al medio (MAC) en la capa de enlace de datos. La comunicación entre dispositivos finales y las puertas de enlace (gateways) se hacen en diferentes canales y a distintas velocidades de datos.

La selección de la velocidad de datos es un compromiso entre la distancia de alcance, y la duración y consumo de energía del mensaje. Debido a la tecnología de espectro ensanchado, las comunicaciones con diferentes tasas de datos no interfieren entre sí y crean un conjunto de canales "virtuales" que aumentan la capacidad del Gateway quien es el encargado de reenviar la información del dispositivo final al servidor.

La tasa de transmisión de datos de LoRaWAN oscila entre 0.3 kbps y 50 kbps. Para maximizar la duración de la batería de los dispositivos finales y la capacidad total de la red, el servidor central de LoRaWAN gestiona la tasa de datos y la salida de RF para cada dispositivo final individualmente mediante un esquema de velocidad de datos adaptativa ADR (Adaptive Data Rate), puesto que el servidor es el encargado de procesar los paquetes de datos recibidos de los dispositivos finales o nodos.

Las principales características de LoRaWAN son:

- Alcance de 10 a 15km en línea de vista
- Encriptación AES 128
- Soporte para 3 clases de nodos
- Administración de dispositivos
- Redes públicas y privadas
- Bajo consumo de energía
- Baja transferencia de datos (hasta 242 bytes)

El método de transmisión utilizado en LoRaWAN es el método ALOHA, el cual permite que los nodos transmitan los mensajes sin comprobar si el canal está libre o no, lo que puede provocar colisiones que impidan la correcta recepción del mensaje en el Gateway. En este caso, el nodo al no recibir una respuesta del Gateway, vuelve a intentar transmitir el mensaje después de esperar un tiempo aleatorio.

En resumen, LoRaWAN define el protocolo de comunicación y la arquitectura del sistema para la red, mientras que la capa física LoRa habilita el enlace de comunicación de largo alcance. El protocolo y la arquitectura de red tienen la mayor influencia para determinar la vida útil de la batería de un nodo, la capacidad de la red, la calidad del servicio, la seguridad y la variedad de aplicaciones atendidas por la red.

2.3.2.1 Arquitectura de una red LoRaWAN.

Una red LoraWAN es implementada normalmente en red estrella como se puede apreciar en la Figura 4. A diferencia de una topología de malla, una topología en estrella es ideal para nodos finales (sensores) con restricciones de energía (es decir, accionados por baterías) porque solo tiene que transmitir sus propios mensajes y no tiene que perder mensajes de retransmisión de vida de la batería de otros nodos finales circundantes. Una red LoraWAN consiste en una o más puertas de enlace LoRaWAN que están conectadas a un coordinador de red central, o también llamado servidor de red (NS).

También, en comparación con las radiobases celulares que tienen un alto nivel de complejidad de hardware y software, además de un alto costo, los gateways LoRaWAN son puentes de protocolos de comunicación básicos con un costo mucho más bajo, diseñados con un hardware y software mucho más sencillo. Los gateways se pueden conectar a través de Wi-Fi, Ethernet con cable o incluso una conexión celular. La puerta de enlace es en esencia, un puente de unión entre el protocolo LoRaWAN y el servidor de red. Un aumento en el número de gateways en un área específica normalmente prolonga la vida útil de la batería del nodo final, ya que la distancia entre los nodos finales y los gateways es más corta. Esto permitirá que los mensajes sean transmitidos con factores de dispersión menores y que requieran menor tiempo al aire.

En la arquitectura de red representada en la Figura 4, el servidor de red opera la comunicación LoRaWAN. Coordina los mensajes entrantes (MAC) entre todas las puertas de enlace y los servidores de aplicaciones. A continuación se muestra una lista de las principales funciones de administración implementadas en el servidor de red:

- a) **Activación de dispositivos IoT (Over the Air Activation OTAA).** El servidor de red se las arregla para acceder a la red (llamadas solicitudes de unión o por su nombre en inglés “join request”) informa a cada nodo en qué canal va a operar, qué factor de dispersión (SF) va a usar y a qué potencia va a transmitir.
- b) **Evita duplicación de datos.** El servidor de red borra los mensajes de radio duplicados recibidos de los propios dispositivos IoT desde distintas puertas de enlace (gateways).
- c) **Enrutamiento dinámico de tramas.** Selecciona la puerta de enlace más adecuada para conectarse con cada nodo para recibir mensajes de enlace de datos.
- d) **Control adaptativo de frecuencia (Adaptive Rate Control ADR).** El objetivo principal de ADR es guardar la energía de la batería de los nodos finales LoRaWAN. Al tener los nodos finales más cercanos a una puerta de enlace de transmisión utilizando el SF más bajo, su tiempo en el aire se reduce al mínimo, lo que prolonga la duración de la batería. Los sensores más distantes transmiten a un SF más alto. Se establece una compensación entre la potencia de la batería y la distancia, ya que un SF más alto permite que una puerta de enlace se conecte a un sensor más alejado.

- e) **Gestión en la red.** El servidor de red puede indicar individualmente a los nodos finales que se conecten a una puerta de enlace diferente al cambiar sus planes de canales.
- f) **Reenvía todos los datos de la aplicación al servidor de aplicaciones correcto.** Cuando un nodo final LoRaWAN solicita unirse a la red utilizando un comando MAC de solicitud de unión (“join request”), el nodo final transmite un código de aplicación específico (AppEUI) que indica a qué servidor de aplicaciones deben reenviarse sus datos. Una vez que el nodo final recibe un comando MAC de aceptación de unión (“join accept”), todas las tramas adicionales que contienen datos de usuario (aplicación) se entregarán a su servidor de aplicaciones designado.
- g) **Administración.** El servidor de red realiza la administración general de la red, reporte de funciones y el aprovisionamiento de la misma.

El servidor de red tiene control sobre las configuraciones operativas y de red clave en cada nodo final LoRaWAN: Nivel de potencia de transmisión, SF, Plan de canales (es decir, que establece canales disponibles), temporización de ventanas de recepción, etc. El control sobre los parámetros mencionados anteriormente proporciona al servidor de red la capacidad de optimizar tanto el rendimiento de la red como la duración de la batería del nodo final.

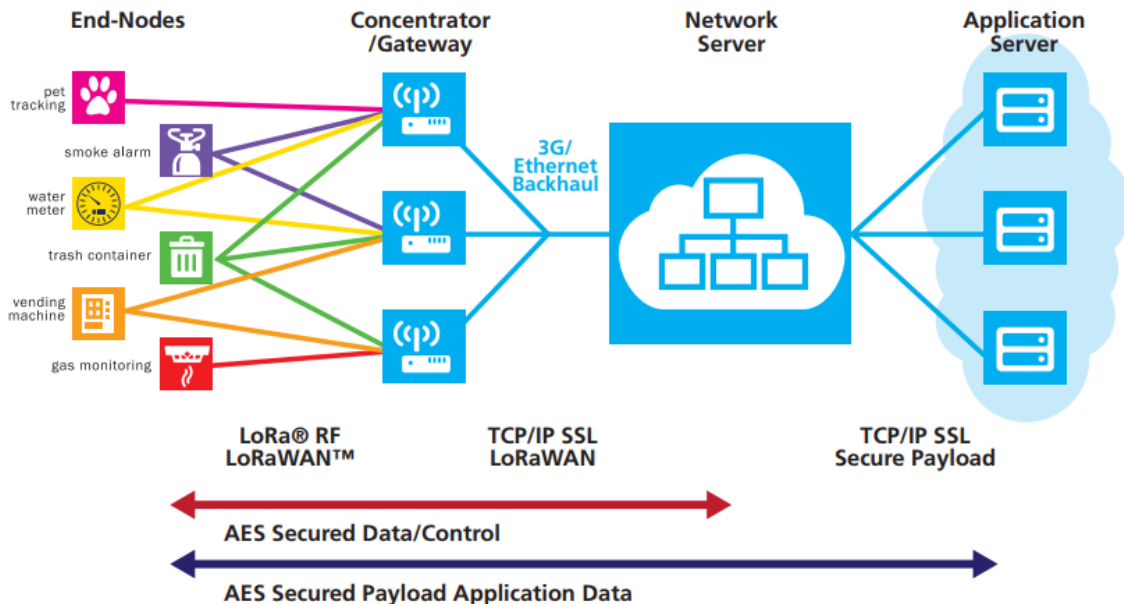


Figura 4 Arquitectura de una red LoRaWAN.

2.3.3 Clasificación de los dispositivos LoRa.

El protocolo de comunicación LoRaWAN ha definido una división. Todos los dispositivos LoRaWAN deben implementar la clase A, mientras que las clases B y C son extensiones de la especificación de los dispositivos de la clase A.

2.3.3.1 Clase A

Esta clase es la más soportada en casi todos los dispositivos, este tipo de clase ofrece el mayor ahorro de energía debido a que sólo entra en modo escucha (llamado ventana TX o de transmisión) después de enviar un dato hacia el Gateway, por eso es ideal para dispositivos que usan una batería.

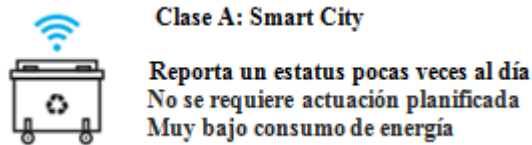


Figura 5 Ejemplo de dispositivo LoRa clase A

2.3.3.2 Clase B

Los dispositivos de clase B añaden la capacidad de recibir datos (downlink) sin necesidad de enviar un paquete (uplink), de esta forma la aplicación puede enviar los datos a los dispositivos de forma programada.

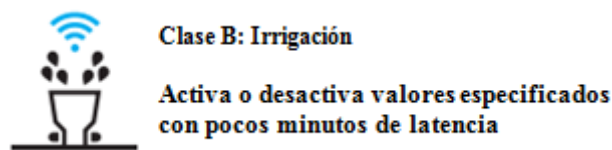


Figura 6 Ejemplo de dispositivo LoRa clase B

2.3.3.3 Clase C

Este tipo de clase ofrece el menor ahorro de energía debido a que los dispositivos de clase C siempre están en modo de recepción y sólo cuando es necesario cambian a modo de transmisión. Esta clase de dispositivos proporciona los mejores tiempos de respuesta y capacidad de envío desde el servidor a los dispositivos. Sin embargo, su consumo de energía es mayor, por lo que en este caso es mejor utilizar esta clase en dispositivos que cuentan con una fuente externa de alimentación.

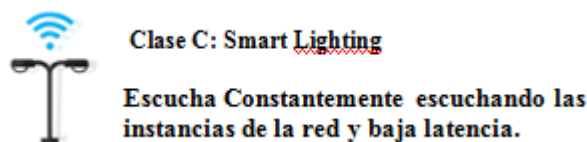


Figura 7 Ejemplo de dispositivo LoRa clase C

2.4 Seguridad en LoRaWAN.

Cualquier tecnología de comunicación que se ocupe de nodos sensores conectados necesita una seguridad robusta de extremo a extremo. LoRa logra esto mediante la implementación de seguridad en dos capas diferentes:

1. En la red
2. En la aplicación

La seguridad en la red garantiza la autenticidad del nodo en la red y la seguridad en la aplicación garantiza que el operador no tenga acceso a los datos de la aplicación del usuario final. LoRa emplea

claves de seguridad AES (Advanced Encryption Standard), que es un esquema de cifrado por bloques de 128 bits implementado por IEEE 802.15 / 42006 Anexo B [IEEE802154]. Mediante el uso de claves separadas para el cifrado de los datos del usuario, la autenticación y la integridad de la red, LoRaWAN ofrece un mayor nivel de seguridad en comparación con las implementaciones de una sola clave.

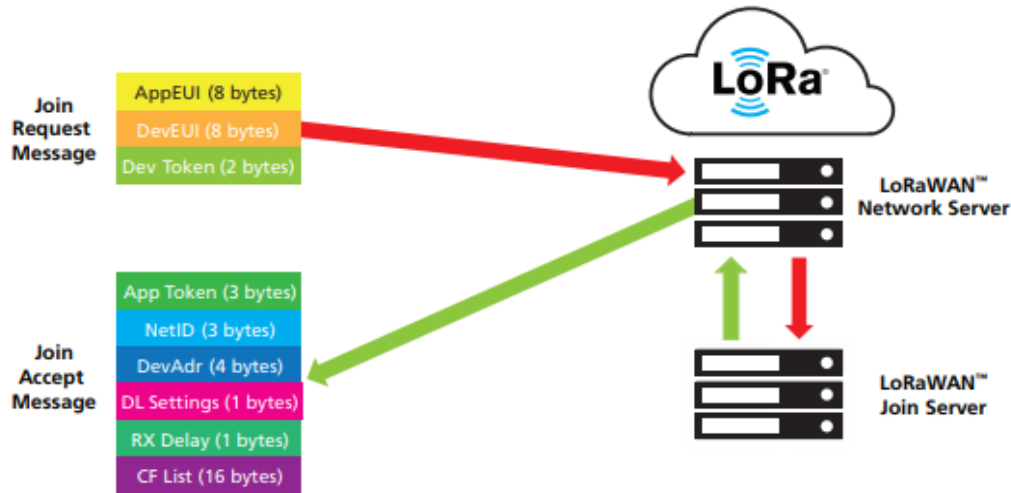


Figura 8 Formato de mensajes « Join Request y Join Accept »

La Figura 8 muestra las dos capas de seguridad de la red: una para datos de control entre los nodos finales y el Servidor (autenticación e integridad) y otra para los datos del usuario que se transportan desde el extremo al servidor de aplicaciones (y viceversa). Como tal, hay dos claves cifradas AES específicas de 128 bits definidas, la clave de sesión de red (NwkSKey) y la clave de sesión de la aplicación (AppSKey). Cada dispositivo IoT tiene su propia NwkSKey y AppSKey.

Hay dos métodos para que un dispositivo IoT o nodo final se conecte a una red LoRaWAN. El primero se llama Activación por personalización (ABP). En este método, las claves NwkSKey y AppSKey que ya están almacenadas en el dispositivo IoT junto con una dirección única del dispositivo de 32 bits y una identificación de red única de 24 bits que permite identificar la red LoRaWAN específica a la que el dispositivo está destinado a conectarse.

El segundo método predeterminado para conectar un dispositivo LoRaWAN IoT es a través de un procedimiento llamado OTAA. En este método, cada dispositivo de IoT envía un mensaje de solicitud de unión (join request) al Servidor de Red (Network Server), que luego reenviará este mensaje a un servidor de combinación como se muestra en la Figura 8.

Existe un identificador de dispositivo "DevEUI" de 64 bits definido por IEEE (equivalente a una dirección "Ethernet MAC" para dispositivos LoRa) que identifica de forma única el nodo final LoRaWAN. También enviará una clave AppEUI única que identifica el servidor de aplicaciones al que el nodo desea conectarse.

Los datos finales constan de un token (autenticador) de dispositivo aleatorio de 2 bytes. El servidor de unión almacenará estos datos de los mensajes de solicitudes (join request) previos de cada nodo. Si el servidor de unión recibe una solicitud de unión (join request) futura de un nodo específico con un token de dispositivo idéntico a uno recibido recientemente, el servidor de unión ignorará esta solicitud de unión. Esto evitará los llamados "ataques de repetición" cuando un pirata informático podría capturar de algún modo un mensaje de radio de solicitud de unión desde un nodo final particular y reproducir (es decir, retransmitir) el mismo mensaje con la intención de desconectar el dispositivo original de la red LoRaWAN.

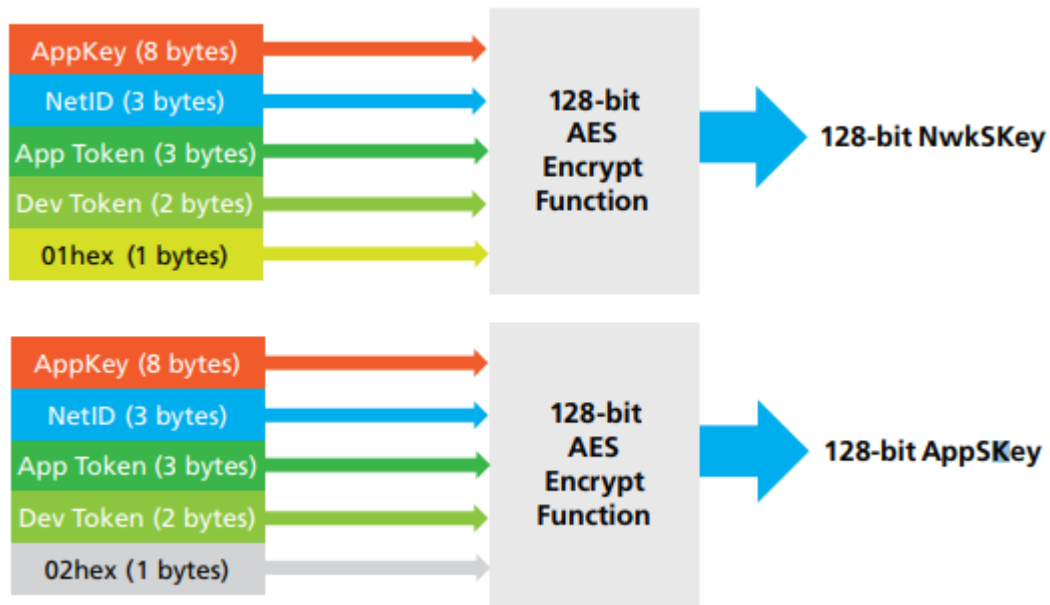


Figura 9 Generación de NwkSKey y AppSKey (Especificación LoRaWAN 1,02)

Solo si el servidor de unión intenta autenticar la combinación de DevEUI y AppEUI, emitirá una dirección de dispositivo única de 32 bits, una identificación de red única de 24 bits y un token de aplicación de 3 bytes. Estos parámetros serán recibidos por el nodo final en un llamado comando de respuesta de unión del NS como se muestra en la Figura 9. El nodo final puede generar sus propias claves de seguridad. La Figura 9 muestra qué campos se utilizan para generar tanto la AppSKey como la NwkSKey. Uno de los campos clave para generar estas claves es AppKey. Este es un valor fijo de 128 bits que es único para cada nodo.

Además de los campos mencionados anteriormente, el mensaje de respuesta de unión (accept request) también informará al nodo final LoRaWAN en qué plan de canal recibir y transmitir, así como algunos otros parámetros de aprovisionamiento de RF.

Cada vez que el nodo final pierde conectividad de red y / o potencia, intentará reactivarse transmitiendo un nuevo mensaje de solicitud de unión (join request).

2.5 Aplicaciones de LoRa

La tecnología LoRa es relativamente nueva, desde hace tres años estas aplicaciones han mostrado tener grandes ventajas dentro de las aplicaciones IoT. Como por ejemplo:

Aplicaciones medio ambientales como son:

- Seguimiento de animales
- Monitorización de las condiciones ambientales en cultivos
- Riego
- Agricultura de precisión
- Detección de incendios forestales
- Detección de inundaciones
- Estudios de contaminación
- Prevención de desastres

- Monitorización de áreas afectadas por desastres, etc.
- Estudios sísmicos
- Seguridad de estructuras

Aplicaciones médicas como son:

- Telemonitorización de datos fisiológicos en pacientes
- Diagnóstico
- Administración de medicamentos
- Seguimiento de médicos y pacientes en hospitales, etc.

Aplicaciones en el hogar/edificios

- Domótica
- Control de electrodomésticos
- Entornos inteligentes
- Control ambiental

Aplicaciones industriales

- Seguimiento de vehículos
- Control de flota
- Control de inventarios

Aplicaciones turísticas

- Interactividad en museos y espacios turísticos, control de acceso

Aplicaciones militares como son:

- Monitorización de fuerzas y equipos enemigos
- Vigilancia en el campo de batalla
- Reconocimiento del terreno
- Detección de ataques biológicos químicos o nucleares, etc.

2.5.1 Desarrollos actuales con LoRa

2.5.1.1 Monitoreo industrial de temperatura.

Esta aplicación es un elemento clave en los procesos industriales tales como fusión de aleaciones en la industria automotriz, termoformado de láminas (plásticos) y la producción de jeringuillas de vidrio. La calidad de fusión de aleaciones en la industria automotriz, la termoformación de láminas en la industria del plástico o la producción de jeringuillas de vidrio, requiere una medición de temperatura en toda la línea de producción. Una anomalía no detectada en tales procesos de producción podría costar millones de dólares.

Mediante la implementación de un sistema industrial de monitoreo de temperatura compuesto por sensores y gateways embebidos con la tecnología LoRa y un protocolo inteligente de antenas de baja potencia LoRaWAN, los fabricantes controlan mejor la calidad de su proceso de producción.

Este sistema funciona de la siguiente manera:

La Tecnología LoRa permite la conectividad, análisis en tiempo real, informes y funciones adicionales como la geolocalización, para lograr esto los sistemas de sensores con tecnología LoRa se colocan dentro de una planta de fabricación para monitorear la temperatura de diferentes procesos. El sensor periódicamente mide la temperatura y envía los datos a un de Gateway basado en LoRa, éste envía información a la red donde los datos son analizados por un servidor.

El servidor genera una alerta que es vista por trabajadores o supervisores a través de una computadora o dispositivo móvil, para que puedan tomar las medidas necesarias.

Por último, los datos son recopilados del sistema y se almacenan en una base de datos de control de calidad para futuras referencias.

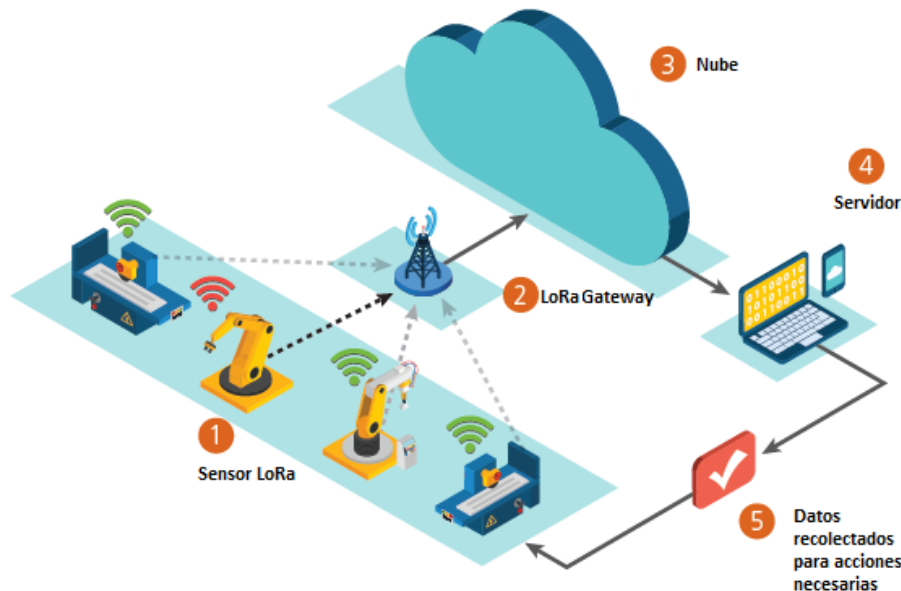


Figura 10 Simulación de proceso de monitoreo de temperatura en un proceso de producción.

Este proceso de monitoreo permite aumentar el rendimiento y la calidad de la producción, permitiendo a los fabricantes utilizar una red pública o privada de manera segura creando un enlace de comunicación confiable entre la infraestructura de detección y el sistema de producción.

2.5.1.2 Monitoreo inalámbrico de niveles de gas.

Esta es una solución inalámbrica que monitorea el nivel de combustible en botellas de gas estacionarias y portátiles que generan alertas de recarga a tiempo para asegurar a los clientes un suministro constante de combustible.

Los dispositivos LoRa y la tecnología de radiofrecuencia inalámbrica (LoRaWAN) hacen que sea fácil y asequible monitorear y administrar los niveles de gas de forma remota. Las botellas de gas “auto-informables” agregan eficiencia y niveles más altos de servicio a la extensa red de clientes atendidos por

un distribuidor típico de gas envasado. Los sensores inalámbricos de largo alcance con capacidad LoRA transforman una botella de almacenamiento convencional en un dispositivo inteligente, que detecta su propio estado de llenado y comunica la información a una aplicación de gestión y programación de inventario en la nube, las botellas de gasolina que se están quedando sin combustible pueden iniciar una solicitud de entrega que se despliega en un teléfono móvil del controlador de ruta apropiado, así mismo, la tecnología LoRa también permite a los distribuidores de gas embotellado anticipar sus patrones de consumo. Esta tecnología se puede adaptar fácilmente a casi cualquier aplicación que involucre tanques de metal cerrados conteniendo líquidos, combustibles o gases líquidos.

Los contenedores inteligentes de las botellas de gas se comunicarán con su aplicación asociada de gestión y programación de inventarios a través de una red de gateways inalámbricos que brindan cobertura en todo el territorio del distribuidor. Los gateways pueden implementarse en una red privada o agregarse a una infraestructura de red de área amplia de baja potencia (LPWAN) existente, como las que utilizan los operadores de telefonía celular y los operadores de cable.

2.5.1.2.1 Caso de uso real de monitoreo inalámbrico de niveles de gas.

Butano 24, una división de Servi Glp SL, una compañía española que ofrece servicios de procesamiento de pedidos para productos de combustible embotellado a empresas de energía y servicios públicos, evaluó varias tecnologías inalámbricas para su aplicación de administración y detección del nivel de botellas de gas basada en IoT. Al final, Butano 24 eligió la tecnología LoRa como base para su plataforma ya que ésta ofrecía una combinación única de ventajas como:

- Bajo costo de implementación. Una aplicación de tecnología LoRa puede operar sobre infraestructuras públicas cuando están disponibles, eliminando la necesidad de grandes gastos de capital. Esto permite a Butano 24 implementar rápidamente su aplicación de medición de nivel de gas en redes LoRaWAN comerciales con requisitos de CAPEX (Capital Expenditures o inversiones en bienes capitales) muy reducidos.
- Bajo costo por unidad. Agregar la tecnología LoRa a un nodo sensor final requiere de bajo costo de inversión, lo que permitió que Butano 24 agregara rápidamente capacidades de “autoinforme” a la flota completa de contenedores y vehículos de entrega de un distribuidor. Los gateways basados en LoRa son igualmente asequibles, con unidades de nivel de operador, capaces de cubrir un área de servicio de más de 15 millas cuadradas, disponible por alrededor de los \$ 1,500 USD.
- Geolocalización sin GPS. El protocolo LoRaWAN admite técnicas de triangulación de radio que pueden identificar la ubicación de un camión de reparto, de una botella de gas u otros activos equipados con un transceptor LoRa de bajo costo y larga duración. Además, LoRa Technology opera en la banda ISM de 900 MHz que puede alcanzar profundamente en caminos cubiertos de árboles, valles empinados y otros lugares donde las señales de GPS de frecuencia más débil no pueden.
- Seguridad. LoRa asegura toda la comunicación utilizando encriptación AES128 “end-to-end”, lo que protege al sistema Butano 24 de ciberataques y robo de datos.
- Alta capacidad. Una sola estación base LoRa puede manejar millones de mensajes, lo que garantiza que las soluciones de detección y administración de botellas de gas Butano 24 sean capaces de soportar múltiples operaciones de distribución en la misma red inalámbrica de gateways.

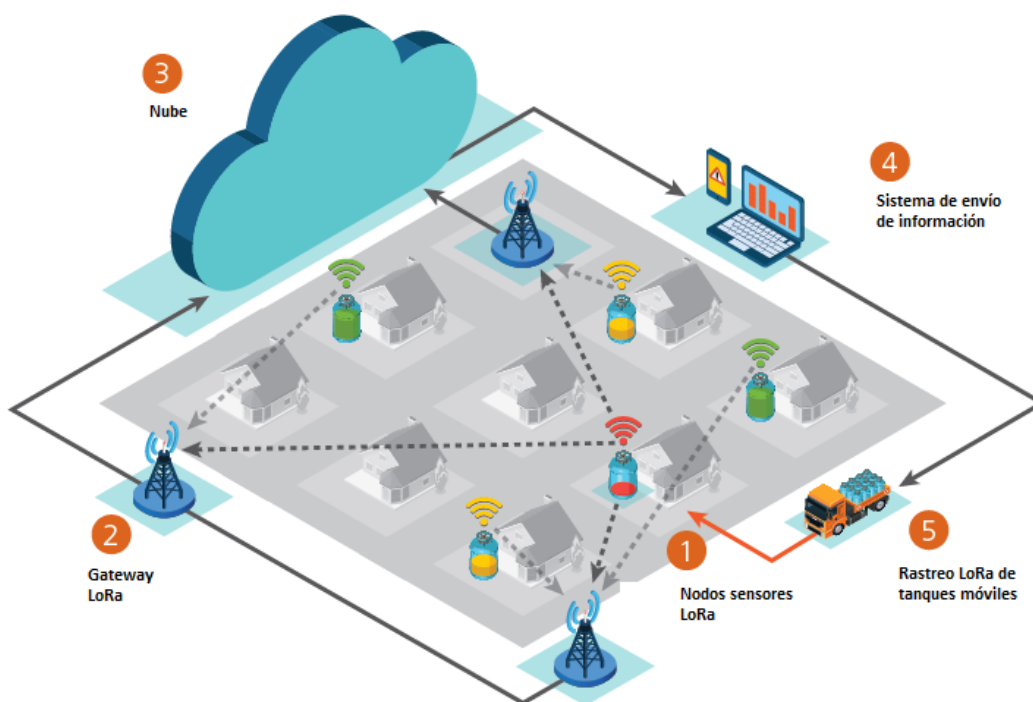


Figura 11. Simulación de una aplicación LoRa de monitoreo inalámbrico de niveles de gas a base

2.5.1.3 *IoT-LoRa Estacionamiento inteligente.*

Otra de las aplicaciones de los dispositivos LoRa y la tecnología de radiofrecuencia inalámbrica (LoRa Technology) es en la transformación de espacios de estacionamiento municipales y comerciales en "lotes inteligentes" gestionados de forma remota que pueden compartir su estado de ocupación en tiempo real con otras aplicaciones habilitadas para IoT. Hasta ahora, las ciudades y los propietarios de estacionamientos no tenían una forma efectiva de informar remotamente a los conductores sobre las vacantes, o de notificarles que estaban llenos los sitios. Como resultado, los conductores urbanos pasan un promedio de 20 minutos por viaje en busca de estacionamientos y se le atribuye un 30% de tráfico innecesario a las calles ya congestionadas de las ciudades.

El advenimiento de IoT puede cambiar esto al permitir que las áreas de estacionamiento informen su estado a los operadores y clientes potenciales en tiempo real. Al proporcionar una conexión robusta entre una red de sensor de ocupación del vehículo y un sistema de gestión basado en la nube, LoRa Technology permite a los operadores optimizar la ocupación, reducir los costos operativos y ayudar a calmar la congestión del tráfico en entornos urbanos, industriales y comerciales.

La implementación de una solución de estacionamiento inteligente basada en LoRa comienza con la instalación de un sensor de ocupación inalámbrico robusto y de alta precisión en cada espacio de estacionamiento. Los sensores utilizan la tecnología LoRa para comunicarse con un gateway que proporciona acceso a IoT.

Una vez desplegado en un espacio de estacionamiento, el sensor de ocupación utiliza el protocolo abierto LoRaWAN para transmitir información confiable sobre la actividad del vehículo a una aplicación de administración en la nube. El sistema de gestión comparte la información de ocupación que recopila con aplicaciones de terceros que notifican a los conductores de espacios abiertos a través de teléfonos

inteligentes u otros dispositivos inalámbricos. Los datos del sensor también se pueden compartir con los sistemas municipales de monitoreo del tráfico para permitir la administración de la congestión predictiva.

2.5.1.3.1 Caso de uso real IoT-LoRa Estacionamiento inteligente.

PNI Sensor Corporation, fabricante líder de localización de precisión y sistemas de seguimiento de movimiento cuenta con una aplicación de conectividad IoT LoRa llamada PlacePod.

Cada espacio de estacionamiento está equipado con un sensor de ocupación alimentado por una batería que puede detectar la ausencia, presencia, llegada y salida de un vehículo. El sensor tiene auto-provisión, con una dirección MAC única asignada a cada unidad que está asociada con su número de serie y código de barras. Estos sensores de baja potencia pueden durar hasta 10 años y no requieren instalación de cableado externo.

Cuando un sensor de ocupación detecta la actividad del vehículo, su transceptor LoRa envía un paquete de mensajes cortos que contiene el cambio de estado a cualquier Gateway inalámbrico dentro de su rango, éste reenvía los paquetes a la aplicación PNI Parking Cloud Service. Esta solución de administración de espacios de estacionamiento utiliza los mensajes del sensor para realizar un seguimiento de los espacios abiertos y ocupados. Puede presentar eventos de los estacionamientos en una interfaz web, y proporcionar datos del mismo para generar información de facturación, y publicar espacios de estacionamiento disponibles para los automovilistas a través de sus teléfonos inteligentes u otros dispositivos inalámbricos. También se puede personalizar para admitir una gama más amplia de servicios especializados. Así mismo la aplicación en la nube de PNI comparte datos en tiempo real con otros servicios de ciudades inteligentes operados por un gobierno municipal o regional.

Esta tecnología de monitoreo de estacionamientos sirve para eliminar atascos a corto plazo, el exceso de tráfico en las zonas de carga y otras perturbaciones que mantienen a los automovilistas dando vueltas en busca de un espacio de estacionamiento.

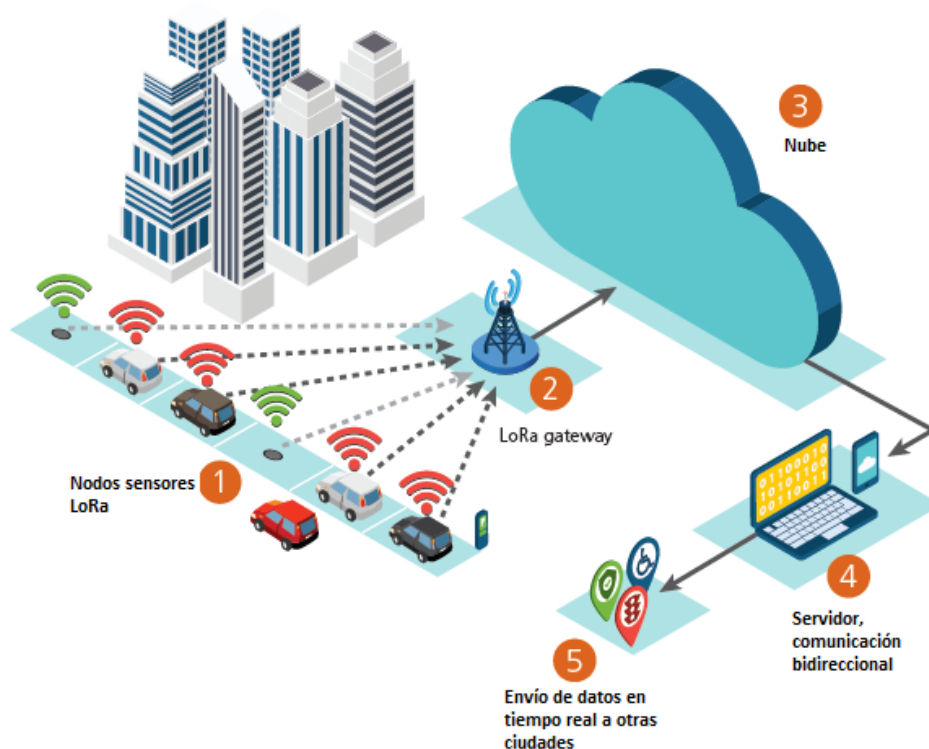


Figura 12 Aplicación estacionamiento inteligente para IoT-LoRa

Las aplicaciones implementadas proporcionan algunos atributos claves que determinan el potencial de desarrollo de los nodos sensores.

Aplicaciones en áreas tan diversas como las citadas anteriormente tienen características y requerimientos específicos. Esta situación junto con la tecnología desarrollada en la actualidad lleva a la aparición de plataformas hardware y software específicos para cada una de estas aplicaciones. Diversas plataformas hardware y tecnologías han sido desarrolladas durante los últimos tres años de una manera acelerada; sin embargo, es necesario seguir con el desarrollo de más dispositivos de este tipo para hacer estas aplicaciones más fiables y robustas en un entorno real.

2.6 Gateway

El Gateway o puerta de enlace es una interfaz entre la plataforma de aplicación y los nodos sensores. Toda la información recibida por el Gateway es acumulada y reenviada a un servidor en este caso The Things Network.

A grandes rasgos un Gateway es un dispositivo que permite conectar redes con protocolos y arquitecturas diferentes. Su propósito es traducir la información del protocolo utilizado en una red al protocolo utilizado en la red de destino, en este caso LoRaWAN, mientras que los nodos son los dispositivos finales que envían información hacia el Gateway.

El gateway es una parte clave de esta solución, dado que el nodo sensor necesita enviar la información a algún lugar dónde pueda analizarse y así enviarse a la nube. Funciona con un programa interno llamado “Packet Forwarder” que le permite retransmitir los mensajes al servidor. A grandes rasgos el ciclo de funcionamiento de transmisión de mensajes va desde el nodo, pasa por el Gateway y los mensajes son reenviados hacia el servidor “The Things Network”. Este proceso se explica más detalladamente en el capítulo 3.

2.7 Servidor The Things Network

The Things Network es una plataforma de uso público que está construyendo una red para el Internet de las cosas mediante la creación de una abundante conectividad de datos.

The Things Network (TTN) es una iniciativa basada en la comunidad para establecer una red global de IoT. La iniciativa fue lanzada por Wienke Giezeman en 2015 y actualmente cubre más de 3.000 Gateways LoRaWAN instalados en más de 90 países. Los voluntarios se encargan de la construcción, el cuidado y el pago de los portales.

En Amsterdam, ha sido posible cubrir gran parte del área urbana en solo unas pocas semanas. Los Países Bajos ya están ampliamente abastecidos con TTN. Lo mismo se aplica a Zurich, Berna y Berlín. En Berlín, se necesitaron solo nueve meses para brindar acceso a LoRaWAN a aproximadamente 3,500,000 personas. Este servidor es el encargado de gestionar la red, almacenar los mensajes y reenviarlos a la aplicación de visualización.

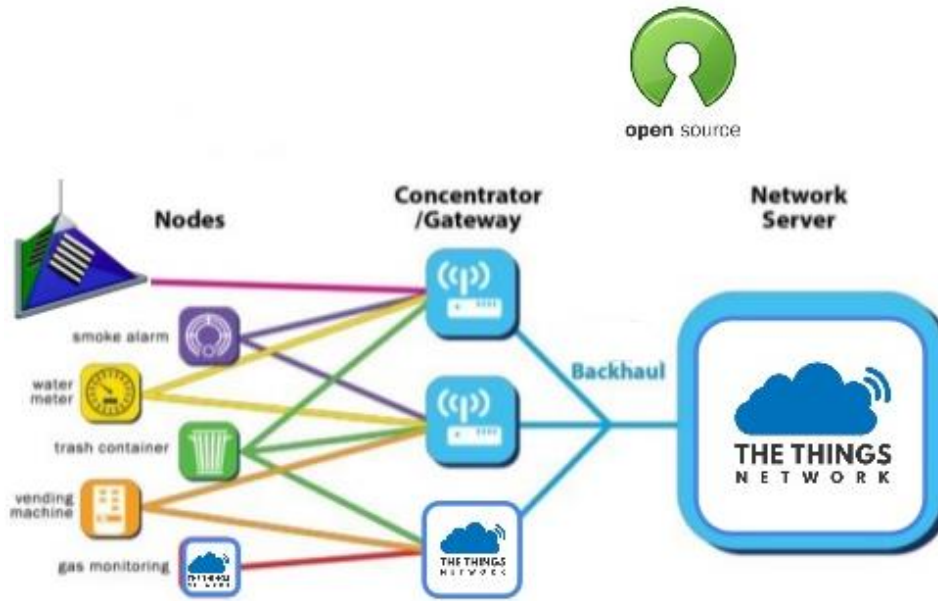


Figura 14 Servidor The Things Network

2.8 Diagrama de Gantt de planeación de actividades del proyecto.

Para llevar a cabo la realización del proyecto y el cumplimiento de objetivos, se han programado las siguientes tareas, asignándoles un tiempo de ejecución detallados en la fecha de inicio y fin del diagrama.

Actividad	Inicio	Final	Febrero	Marzo	Abril	Mayo	Junio	Julio
Investigación del Estado del Arte LoRa	12/02/2018	16/02/2018						
Estudio de LPWANs	19/02/2018	21/02/2018						
Estudio de LoRa	22/02/2018	26/02/2018						
Búsqueda de elementos para el nodo sensor	26/02/2018	28/02/2018						
Diseño electrónico del nodo	01/03/2018	05/03/2018						
Construcción del sistema embebido	06/03/2018	07/03/2018						
Soldado de la tarjeta Mini Ultra Pro al tranceptor RFM95 W	07/03/2018	07/03/2018						
Conexión de los sensores a la tarjeta Ultra Pro	08/03/2018	08/03/2018						
Programación del nodo	08/03/2018	16/03/2018						
Selección de la antena para el nodo	19/03/2018	20/03/2018						
Conexión de la antena al nodo	21/03/2018	21/03/2018						
Configuración de la aplicación LoRa en el Servidor The Things Network	22/03/2018	27/03/2018						
Configuración del Gateway Multichannel	28/03/2018	30/03/2018						
Configuración del Gateway Lopy Monochannel	02/04/2018	04/04/2018						
Configuración de los Gateways en el Servidor The Things Network	05/04/2018	06/04/2018						
Pruebas de funcionamiento del nodo sensor con el Servidor The Things Network	09/04/2018	13/04/2018						
Diseño Mecánico de la carcasa para el nodo	16/04/2018	20/04/2018						
Selección de junta Tórica	20/04/2018	20/04/2018						
Impresión 3D de carcasa para el nodo	23/04/2018	26/04/2018						
Pruebas de funcionamiento de la carcasa	27/04/2018	30/04/2018						
Pruebas de alcance internas con el nodo terminado (antena de hilo y antena helicoidal)	01/05/2018	18/05/2018						
Validación del funcionamiento de la red LoRa dentro de Ikerlan	21/05/2018	08/06/2018						
Pruebas de consumo de energía	11/06/2018	15/06/2018						
Documentación	15/06/2018	11/07/2018						
Presentación a Ikerlan del trabajo de Fin de Máster	12/07/2018	12/07/2018						

Diagrama 1 Planeación de actividades del trabajo de fin de máster.

3. ESTRUCTURA Y DESCRIPCIÓN DEL NODO SENSOR (HARDWARE)

En este capítulo se ha desarrollado la descripción de los elementos que conforman el nodo sensor. A continuación se despliegan las funciones en detalle los diferentes elementos seleccionados para la implementación del hardware y sus características más relevantes.

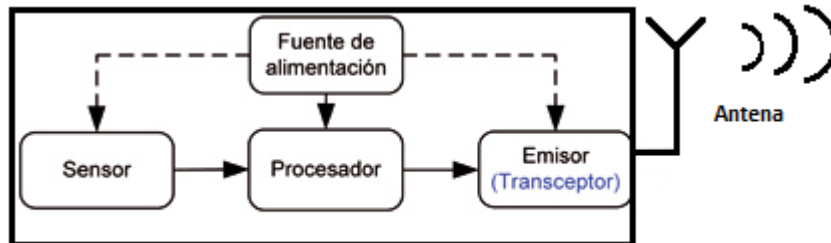


Figura 15 Estructura de un nodo sensor

3.1. Tarjeta Mini Ultra Pro.

Para este proyecto se seleccionó la tarjeta Mini Ultra Pro del fabricante Rocket Scream, esta tarjeta puede ser utilizada como un nodo LoRaWAN clase A, cuyas principales características principales son:

- Microcontrolador – ATSAM21G18A-AU (montado en la tarjeta).
- Reloj – 48 MHz
- Voltaje de operación de 3.3 V.
- Modo de bajo consumo de energía (sleep) de 20.0 μ A (conocido como modo espera para este microcontrolador).
- Dimensiones de 25.4 mm x 55.88 mm.

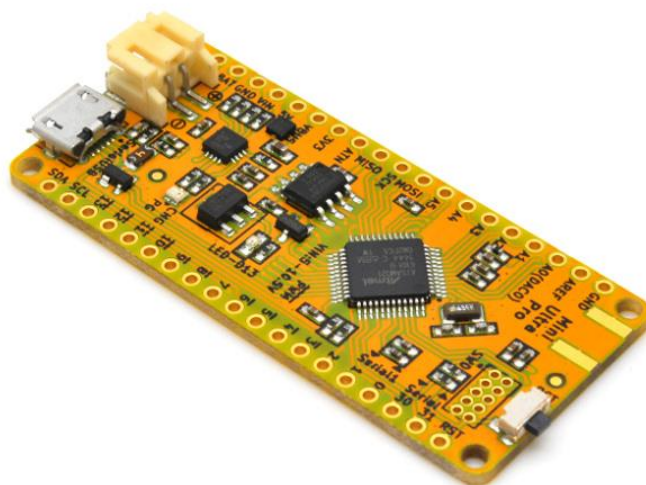


Figura 16 Tarjeta Mini-Ultrapro de Rocket Scream

3.1.1 Microcontrolador ATSAM21G18A-AU

Es un microcontrolador con un procesador ARM Cortex M0+ de 32 bits que cumple con la tarea de recolectar, procesar datos y controlar la funcionalidad de otros componentes en el nodo sensor, así como el almacenamiento de los datos provenientes de los sensores para transmitirlos al radio transmisor, así como la transmisión de los mismos al Gateway.

Dentro de los procesadores que se encuentran en el mercado los Cortex-R y Cortex-M son ideales para dispositivos IoT, en ambos casos se habla de procesadores de 32 bits, además de que las soluciones de ARM cumplen con las expectativas generales del proyecto: son pequeños y, aunque también poco potentes en comparación con otros chips del mercado, son más conocidos debido a su utilidad en dispositivos finales.



Figura 17 Microcontrolador ATSAM21G18A-AU

3.1.2 Transceptor módulo LoRa RFM95W

En este documento se hará énfasis en la tecnología LoRaWAN debido a que es el protocolo estudiado para la implementación del nodo sensor. Para lograr esto se ha seleccionado el transceptor RFM95W, cuya frecuencia de trabajo es de 868 MHz y que transmite a una potencia de 20 dBm.

El transceptor RFM95 cuenta con un modem de largo alcance LoRa, el cual transmite a una frecuencia de 868 MHz, gracias a la técnica de modulación “LoRa” el transceptor RFM95 alcanza una sensibilidad muy alta, además de que cuenta con inmunidad alta a la interferencia, mientras que minimiza el consumo de corriente.



Figura 18 Transceptor LoRa RFM95W

3.2 Sensores

3.2.1 Sensor de temperatura DHT22

El DHT22 es un sensor digital de temperatura y humedad. Se utiliza un sensor de humedad capacitivo y un termistor para medir el aire circundante, y arroja una señal digital en el pin de datos, puede obtener nuevos datos una vez cada 2 segundos.

El sensor está soldado en una placa y con tres pines de conexión, además de una resistencia pull-up de 4.7 k Ω .



Figura 19 Sensor de humedad y temperatura DTH22

3.2.1.1 Secuencia y calificación de Pines (Orden ascendente)

En la siguiente tabla se muestra la asignación de pines:

Número de PIN	Función
1	VDD---Voltaje
2	DATA (Señal SDA)
3	No conectado
4	GND

Tabla 4 Asignación de Pines del sensor de temperatura

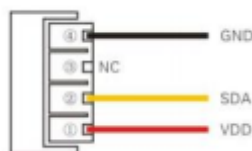


Figura 20 Pines del sensor DHT

Este sensor cuenta con las siguientes características:

- Alimentación: 3.3v – 5.5v, tomando como valor recomendado 5v.
- Resolución decimal, es decir, los valores tanto para humedad como para temperatura serán números con una cifra decimal.
- Tiempo de muestreo: 2 segundos, es decir, sólo ofrece datos cada 2 segundos.

3.2.1.2 Rango de mediciones de temperatura

- Rango de valores desde -40°C hasta 80°C de temperatura.
- Precisión: $\pm 0.5^{\circ}\text{C}$, $\pm 1^{\circ}\text{C}$ como máximo en condiciones adversas.
- Tiempo de respuesta: <10 segundos, es decir, de media, tarda menos de 10 segundos en reflejar un cambio de temperatura real en el entorno.

3.2.2 Acelerómetro ADXL355

Este sensor acelerómetro ADXL355 permite medir la aceleración en los tres ejes espaciales X, Y, Z. Es un sensor compacto y de tamaño muy reducido, funciona a 3.3 V y el rango de medición es de -3G a +3G (G=aceleración de la gravedad, 1G=9.8m/s²).

El sensor se conecta a tres entradas analógicas, una por cada eje de medición, el conector VCC se conecta a 3.3 V. La finalidad de este acelerómetro es conocer la posición del nodo sensor, suponiendo el nodo en reposo es posible encontrar su orientación en el espacio y conocer si ha sufrido cambios de posición en algún momento determinado

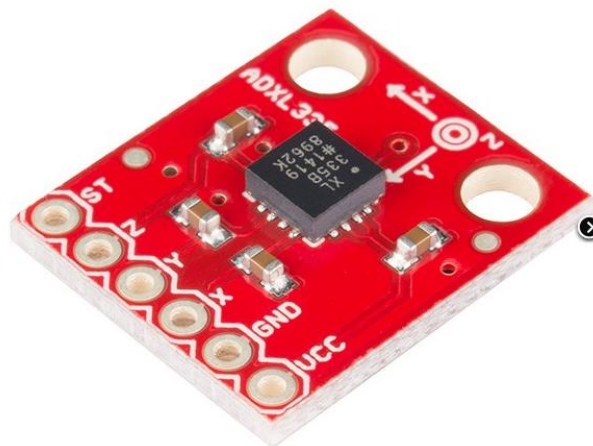


Figura 21 Acelerómetro ADXL355

Este sensor cuenta con las siguientes características:

- Voltaje de alimentación: 3.0 a 5.0 VDC
- Bajo consumo: 350uA
- Dimensiones: 20mm x 15mm
- Peso: 04 g

3.3 Antenas para el nodo sensor

Para este proyecto se hicieron pruebas con dos tipos de antenas, una antena de cable y la antena helicoidal, para ambos casos se realizaron pruebas de alcance dentro de Ikerlan, con los dos tipos de Gateways a utilizar.

De acuerdo a las pruebas realizadas se sabe que la implementación de antena más económica es una antena de cable de un cuarto de longitud de onda. Uno de los problemas de la producción en volumen de

los dispositivos finales IoT radica en minimizar las variaciones en la ubicación de la antena. Se debe tener cuidado al insertar el cable de la antena dentro de la carcasa del nodo para garantizar un rendimiento de la propagación de señal RF consistente en el dispositivo.

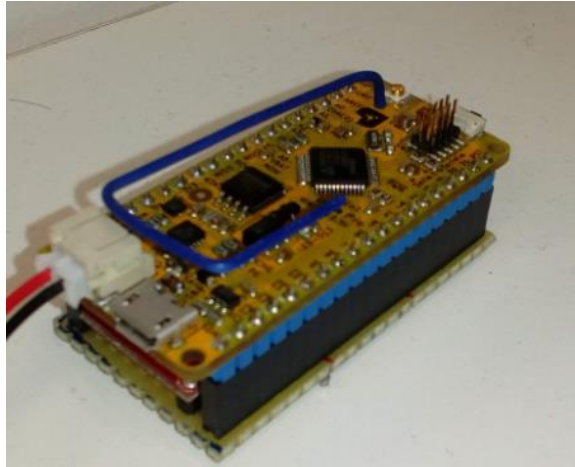


Figura 22 Antena de hilo nodo sensor 2

Como segunda opción se utilizó una antena de bobina (helicoidal), esta es una alternativa de menor costo en comparación con una antena basada en chip. Una antena helicoidal es simplemente un cable que se enrolla en forma de una bobina. La longitud total del cable determina su frecuencia de resonancia, pero enrollarlo puede reducir en gran medida su tamaño físico. Las antenas helicoidales tienen un ancho de banda estrecho, lo cual no es un problema para LoRaWAN ya que solo una parte estrecha del espectro ISM 868 MHz es ocupada. Las antenas helicoidales se pueden estropear fácilmente por la presencia de otros objetos en la PCB, por lo que se debe tener cuidado en el diseño.



Figura 23 Antena helicoidal nodo sensor 3

3.4 Batería

Un aspecto importante en el diseño de un nodo sensor inalámbrico es asegurarse que hay siempre la energía adecuada disponible para alimentar el sistema. El nodo sensor consume potencia del sensor, comunicando y procesando datos. La mayor cantidad de energía es requerida para la comunicación de datos que para cualquier otro proceso.

La energía que cuesta de transmitir 1 Kb de una distancia de 100 metros (330 ft) es aproximadamente igual a aquella utilizada para la ejecución de 3 millones de instrucciones por unos 100 millones de instrucciones por segundo/W de procesador. Ya que los nodos de acuerdo a su aplicación son a menudo inaccesibles, su vida útil depende de la duración de sus recursos energéticos. Teniendo en cuenta que el

tamaño de éstos es limitado, el almacenamiento de energía se ve directamente afectado y, por tanto, en aplicaciones que requieren un tiempo de vida mayor es necesario buscar alternativas.

El dispositivo que se encarga de alimentar los otros componentes del nodo para este proyecto consiste en una batería de Li-Po 3.7V similar a los teléfonos móviles, su capacidad se mide en mAh (mili amperios hora) que es básicamente una indicación de cuánta corriente es consumida por hora.

Este tipo de baterías proporciona una fuente de alimentación muy eficiente, a la vez que son muy ligeras y soportan altos picos de corriente. Esta a su vez tiene incorporado un conector tipo Molex para conectarse directamente a un circuito integrado al cargador USB y cuenta con las siguientes características:

- Dimensiones: 50.8 x 33.5 x 5.9 mm
- Capacidad nominal: 1000 mAh
- Voltaje nominal: 3.7 V
- Temperatura de trabajo: 0-40°C
- Peso: 20 g

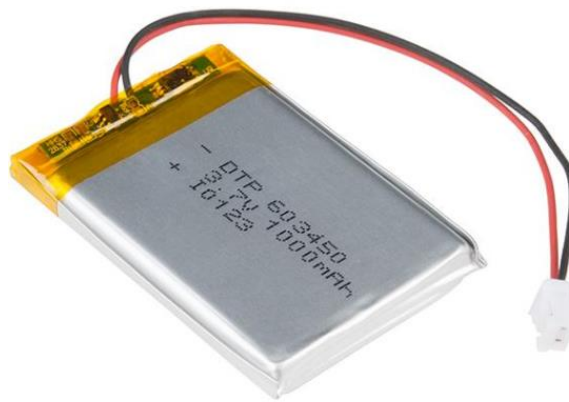


Figura 24 Batería LiPo para nodo sensor

3.5 Descripción de Gateways

Hay dos tipos de Gateways a utilizar en este proyecto, el prototipo diseñado funciona con dos soluciones, cuya finalidad es que en caso de fallo del Gateway principal multi-canal, exista otro Gateway en redundancia capaz de continuar con la transmisión de los datos a un bajo costo. Una de las soluciones trabaja con 8 canales de transmisión, mientras que el otro funciona en un único canal.

3.5.1 Gateway Multitech (multicanal)

Este Gateway que se utiliza especialmente en aplicaciones IoT, funciona en un ambiente de desarrollo Linux, tiene una capacidad de ocho canales de escucha simultánea en una frecuencia de 868 MHz.



Figura 25 Gateway Multitech

Para su configuración se utilizó el software de licencia libre Putty y un cable Ethernet, el Gateway cuenta con su propio programa de gestión que permite establecer parámetros de configuración para la comunicación de éste con la nube. La figura 26 muestra el IDE de Multitech para este Gateway. Cuenta con un modo de trabajo llamado “Packet forwarder” que debe instalarse manualmente para retransmitir los mensajes al servidor.

The screenshot displays the 'Access Configuration' page of the Multitech MultiConnect Conduit web interface. The page is titled 'MultiConnect® Conduit - Application Execution Platform' and shows the user is logged in as 'admin'. The interface includes a sidebar with navigation options like Home, Setup, Firewall, and Administration. The main content area is divided into several sections:

- Web Server:** Contains settings for 'HTTP Redirect to HTTPS' (Enabled, Port 80, Via LAN checked, Via WAN checked) and 'HTTPS' (Port 443, Via WAN checked, Timeout Minutes 5, Username admin).
- SSH:** Contains settings for 'Enabled' (checked), 'Port' (22), 'Via LAN' (checked), and 'Via WAN' (checked).
- ICMP:** Contains settings for 'Enabled' (checked), 'Respond to LAN' (checked), and 'Respond to WAN' (unchecked).
- IP Defense:** Contains settings for 'DoS Prevention' (Enabled unchecked, Per Minute 60, Burst 100), 'Ping Limit' (Enabled unchecked, Per Second 10, Burst 30), and 'Brute Force Prevention' (Enabled unchecked, Attempts 3, Lockout Minutes 5).

A 'Submit' button is located at the bottom right of the configuration area.

Figura 26 IDE de gestión del Gateway Multitech

En el capítulo 4 se hace una descripción detallada de la configuración de los dispositivos.

3.5.2 Nano Gateway Lopy (Mono canal)

En este proyecto se utilizó un segundo Gateway para la comunicación con el Servidor The Things Network, este Gateway es de la marca Pycom, a diferencia del Gateway Multitech, este Gateway es mucho más pequeño y funciona en un solo canal, además de ser mucho más barato en precio. El funcionamiento general de este Gateway se basa en tres archivos que se utilizan para configurar y especificar cómo debe configurarse para poder conectarse a la red preferida y al igual que el Gateway Multitech, cómo debe de actuar el “Packet forwarder” para funcionar en conjunto con el servidor. Estos archivos son: main.py, config.py v nanogateway.py.



Figura 27 LoPy nano Gateway

3.5.2.1 Antena de Gateways

Ambos Gateways funcionan con una antena externa IP67, impermeable que trabaja en la frecuencia 868 MHz, con las siguientes especificaciones:

Características	Descripción
Ancho de banda	CF+-3MHz
Polarización	Lineal
Ganancia	2 dBi
Impedancia	50 Ω
Temperatura de operación	-40°C ~+85°C
Humedad Relativa	Mayor al 95 %
Dimensiones	75 mm x 8.5 mm

Tabla 5 Especificaciones antena para Gateways



Figura 28 Antena para Gateways.

4. CONFIGURACIÓN DEL NODO SENSOR (SOFTWARE)

Para participar en una red LoRaWAN, cada dispositivo final o nodo necesita ser personalizado y activado, la activación OTAA (over the air activation) es el método de activación utilizado para este proyecto

4.1. Configuración del nodo en la nube The Things Network.

Para comunicarse a través de The Things Network, es necesario registrar el nodo sensor en la nube. Para ello, es necesario contar con la clave “Dev EUI”, que es una dirección única, codificada en la tarjeta MiniUltra Pro, únicamente es necesario cargar el siguiente código desde el IDE de Arduino modificando la frecuencia a trabajar con TTN_FP_EU868:

```
#include <TheThingsNetwork.h>
#define loraSerial Serial1
#define debugSerial Serial

// Replace REPLACE_ME with TTN_FP_EU868 or TTN_FP_US915
#define freqPlan REPLACE_ME

TheThingsNetwork ttn(loraSerial, debugSerial, freqPlan);

void setup()
{
  loraSerial.begin(57600);
  debugSerial.begin(9600);
}

void loop()
{
  debugSerial.println("Device Information");
  debugSerial.println();
  ttn.showStatus();
  debugSerial.println();
  debugSerial.println("Use the EUI to register the device for OTAA");
  debugSerial.println("-----");
  debugSerial.println();

  delay(10000);
}
```

Una vez cargado el código al abrir la ventana device monitor la información con el DEV_EUI es desplegada.

Con este dato es necesario crear una cuenta en TTN y añadir una aplicación en donde los dispositivos deben registrarse con el DEV_EUI para poder obtener comunicación entre ellos y la nube.

Dentro del entorno de configuración de la nube como se muestra a continuación:

En console.thethingsnetwork.org, hacer clic en **agregar aplicación**.

- Para el **ID de la aplicación**, se debe seleccionar un ID único compuesto por minúsculas, caracteres alfanuméricos y no consecutivos `-y_`.
- En la sección de **Descripción**, se puede agregar una breve descripción del dispositivo o se puede dejar en blanco.

- Por último, la opción “Register application on the handler ttn-handler-eu” debe dejarse seleccionado, a menos que se vaya a utilizar esta aplicación en una región diferente a su región predeterminada.

The screenshot shows the 'ADD APPLICATION' form in the TTN console. The form has the following fields and options:

- Application ID:** A text input field with the description 'The unique identifier of your application on the network'.
- Description:** A text input field with the description 'A human readable description of your new app'.
- Register application on the handler ttn-handler-eu:** A checked checkbox with the subtext 'If selected, registers your app to the handler ttn-handler-eu, to make it usable right away'.

At the bottom right of the form, there are two buttons: 'Cancel' and 'Add application'. The TTN logo and navigation menu are visible at the top.

Figura 29 Registro de aplicación en la nube TTN

Una vez hecho este paso se registra el nodo en la aplicación que se acaba de crear, para esto se selecciona el menú superior derecho en la misma ventana de aplicación y seleccionar “Register Device”.

Los datos que deben rellenarse en esta ventana son ID de dispositivo, el cuál debe estar escrito en el mismo formato que la aplicación.

El dato DevEUI debe ser el mismo que se obtuvo con el código descrito anteriormente. Por último se generarán dos datos automáticamente por lo que no es necesario introducir ningún dato en estos apartados “App Key” y “App EUI”.

The screenshot shows the 'REGISTER DEVICE' form in the TTN console. The form has the following fields and values:

- Device ID:** An empty text input field with the description 'This is the unique identifier for the device in this app. The device ID will be immutable.'
- Device EUI:** A text input field containing 'Device EUI will be generated' with a pencil icon on the left.
- App Key:** A text input field containing 'App Key will be generated' with a pencil icon on the left.
- App EUI:** A dropdown menu containing the value '70 B3 D5 7E F0 00 00 3E'.

At the bottom right of the form, there are two buttons: 'Cancel' and 'Register'. The TTN logo and navigation menu are visible at the top.

Figura 30 Registro de aplicación en la nube TTN

Al dar click en “Register” una ventana con la “App Key” es generada y desplegada, esto con la finalidad de activar o dar de alta el nodo en la nube

4.2 Activación del nodo en la nube.

El servidor The Things Network tiene la opción de personalizar diferentes dispositivos IoT. El método utilizado para activar el nodo sensor es OTTA (Over the air activation), este método es el más confiable ya que la activación de los dispositivos es más segura pues las claves de sesión o “Session Key” se negocian cada vez que el dispositivo es activado en la red.

Para agregar el nodo al servidor de red se necesitan los siguientes parámetros:

- **DevEUI:** clave de 64 bits que identifica al dispositivo globalmente.
- **AppEUI:** clave de 64 bits, identificador de la aplicación.
- **AppKey:** clave de 128 bits de aplicación de la cual se derivan las claves NwkSKey y el AppSKey.

Una vez que activado el dispositivo en el servidor se guardan los siguientes datos:

- **DevAddr:** cclave de 32 bits que identifica al dispositivo dentro de la red.
- **AppEUI:** 64 bits, identificador de la aplicación.
- **NwkSKey:** clave de sesión de red utilizado para identificar el nodo.
- **AppSKey:** clave de sesión de aplicación, utilizado para encriptar y desencriptar el payload del nodo en la capa MAC.

Todas las claves anteriores tienen un propósito fundamental de administración y seguridad en una red LoRaWAN que se explica a detalle en el apartado Seguridad en LoRaWAN.

Para la activación de la tarjeta Ultra Mini Pro, es necesario cargar el siguiente código de manera especial por primera vez ya que los pines de conexión de entrada y salida son diferentes a los pines que están programados por defecto con el código general para dar de alta los dispositivos IoT.

```
// Pin mapping
const lmic_pinmap lmic_pins = {
    .nss = 5,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = 3,
    .dio = {2, 6, LMIC_UNUSED_PIN},
};
```

Es de suma importancia cargar esta configuración o de lo contrario no se podrá tener comunicación con el servidor.

4.3 Configuración general del Gateway Multitech Conduit.

Este tipo de Gateway es utilizado para aplicaciones IoT, consta de 8 canales de transmisión y recepción con una frecuencia de trabajo de 868 MHz.

Para la correcta configuración del Gateway son necesarios los siguientes pasos:

1.-Abrir la página de configuración en cualquier buscador de internet con la siguiente dirección:
<http://192.168.2.1/>, la cual abrirá una ventana de diálogo de MULTITECH



Figura 31 Ventana de acceso al Gateway Multitech

Las credenciales de acceso por defecto son:

Username: admin

Password: admin

Después de entrar al portal principal de configuración del Gateway es necesario cambiar el usuario y la contraseña.

Después de cambia la configuración de la zona horaria y la fecha en la cual estará trabajando el Gateway:

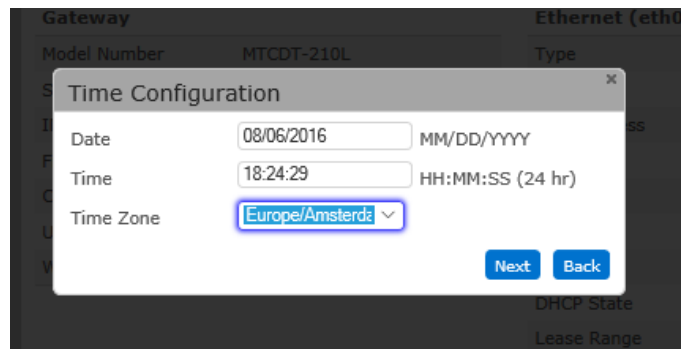


Figura 32 Configuración de hora y fecha del Gateway

Para la configuración de internet se debe seleccionar el modo DHCP client:

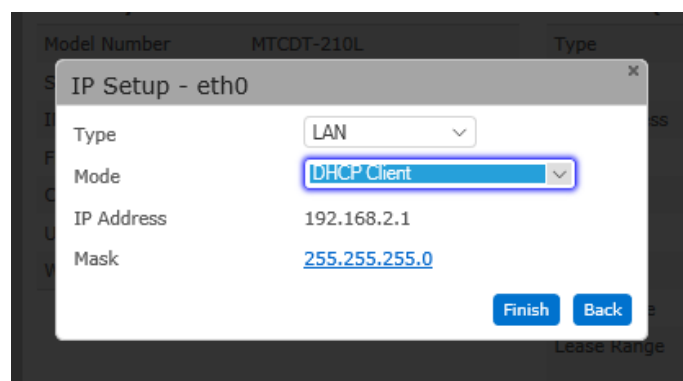


Figura 33 Ventana de configuración de internet para el Gateway

Al acceder a la configuración principal del Gateway se deben seleccionar las siguientes opciones de funcionamiento:

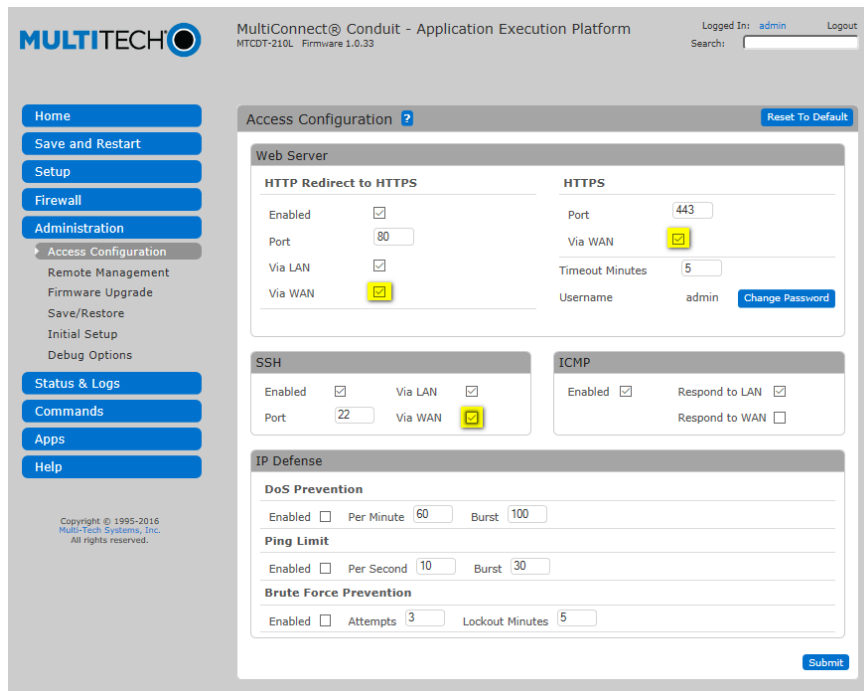


Figura 34 Configuración del equipo Gateway para su administración

Cuando se activa la opción Via WAN una ventana de advertencia es desplegada, para esto simplemente se omite esta opción seleccionando “continuar”

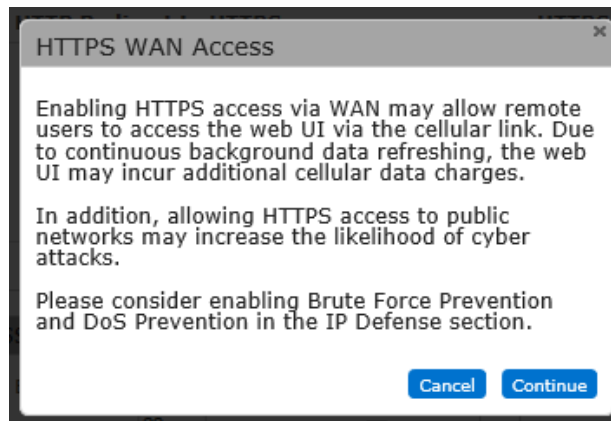


Figura 35 Ventana de acceso WAN

Una vez seleccionadas las tres opciones descritas dar click en Submit, para guardar la configuración, a continuación seleccionar Setup>Network Interfaces del menú como se observa en la imagen:

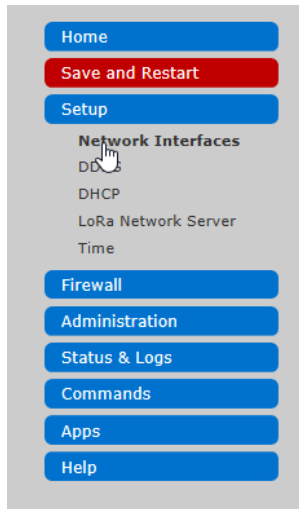


Figura 36 Ventana de configuraciones Gateway Multitech

Y en la pantalla siguiente seleccionar la línea eth0:

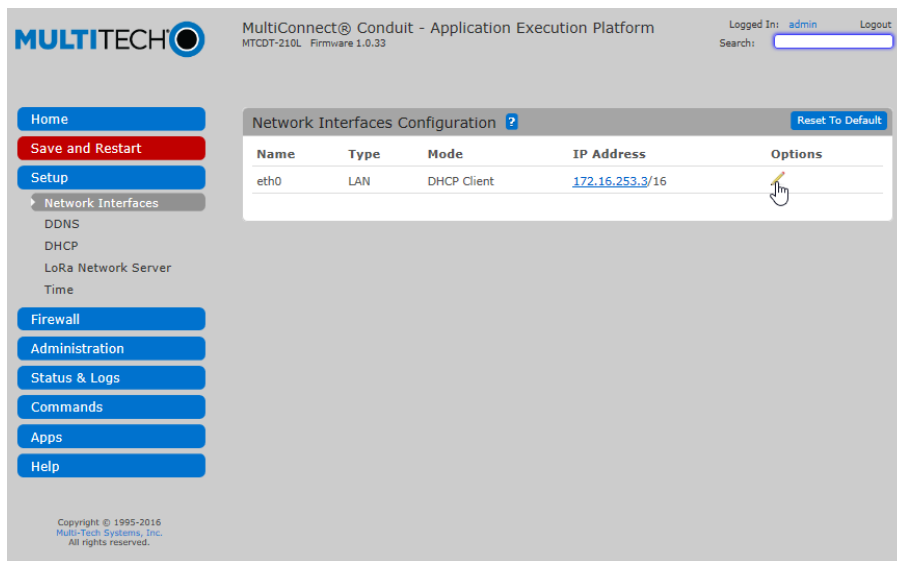


Figura 37 Ventana de configuración de interfaces de internet.

Para esta opción se creó una IP estática para poder administrar el Gateway, como DNS fue configurada la dirección pública de Google 8.8.8.8. , seguido de la opción Finish.

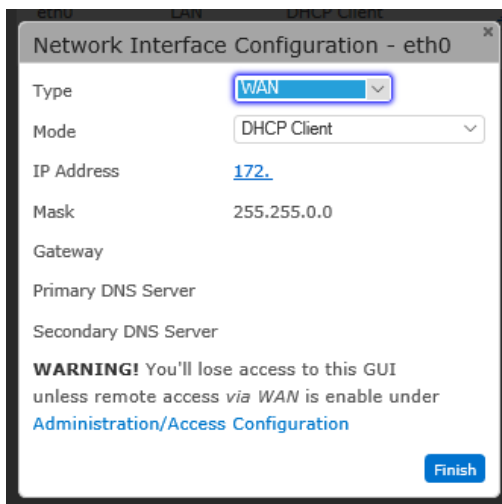


Figura 38 Configuración de IP estática.

Y seleccionar las opciones guardar y reiniciar.

El Gateway se reiniciará y se podrá acceder a él por medio de un cable Ethernet y PuTTY con la IP que se estableció por efecto en la configuración anterior con el comando admin@192.168.8.151 y seleccionar el puerto al que se encentra conectado.

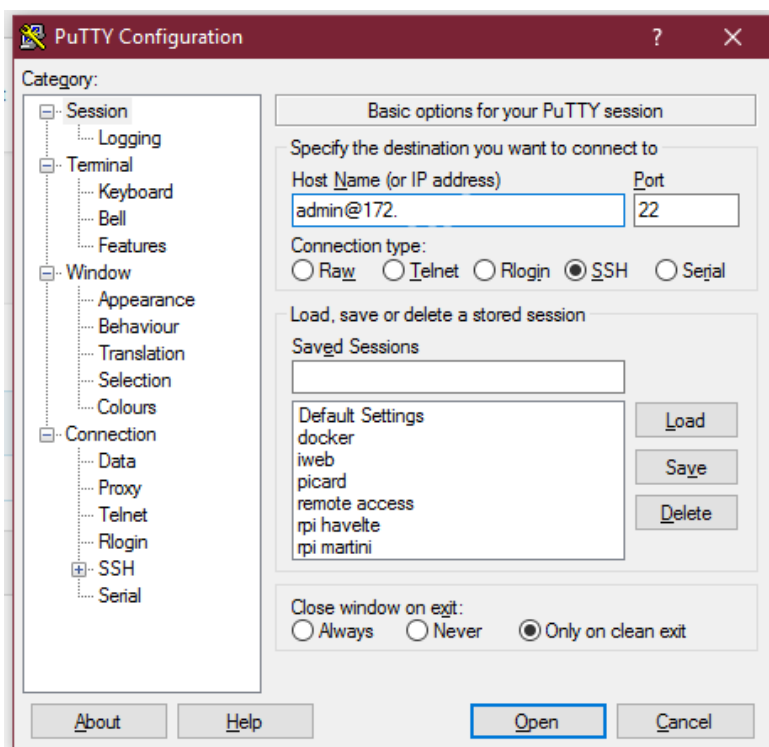


Figura 39 Ventana de configuración de PuTTY.

Para el funcionamiento del Gateway se necesita cargar la aplicación Packet forwarder para The Things Network.

4.4. Configuración del packet forwarder para el gateway multitech

El Packet forwarder es un programa que se ejecuta en el Gateway para que éste pueda interactuar con el chip LoRa para recibir y transmitir paquetes LoRa, y para permitirle interactuar con la red para transmitir estos paquetes a las aplicaciones.

La siguiente imagen muestra a base de bloques el funcionamiento del Packet forwarder en una aplicación LoRa para la nube de The Things Network.

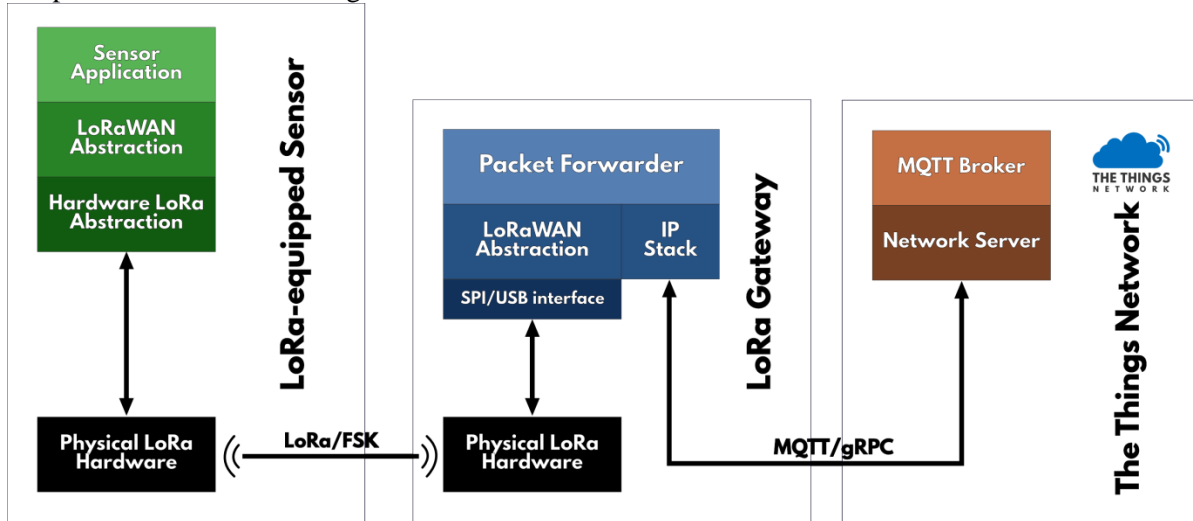


Figura 40 Funcionamiento del Packet Forwarder LoRa

Como se puede observar en la figura 40 los Gateways están conectados al servidor de red mediante una conexión IP estándar utilizando el protocolo de comunicación MQTT, mientras que los nodos o dispositivos finales usan el protocolo LoRaWAN para conectarse a uno o varios Gateways.

El programa Packet Forwarder se puede descargar directamente a Gateway pulsando el siguiente comando en el entorno de configuración:

```
wget https://github.com/kersing/multitech-installer/raw/master/installer.sh --no-check-certificate
```

Después de descargarlo se ejecuta el comando `sh installer.sh` y la siguiente ventana de datos será desplegada:

```

admin@mtcdt:~# sh installer.sh
AEP Model detected, have time zone and network been setup?
1) Yes
2) No
#? 1
 6 Aug 19:01:33 ntpdate[26583]: adjust time server 178.79.160.57 offset 0.000223
  sec
SETUP FREQUENCY PLAN
Detected 868MHz card, use TTN 868 configuration?
1) Yes
2) No
#? 1
SETUP LORA GATEWAY CONFIGURATION
E-mail address of gateway operator: myemail@example.com
Gateway description: Demo Gateway
Include location information?
NOTE: No location information means the gateway status information will not be a
available on-line
1) Yes
2) No
#? 1
Gateway location information
latitude: 53.189768
longitude: 6.557384
altitude: 5

Your gateway information is:
e-mail contact: myemail@example.com
description    : Demo Gateway
Check Location: https://maps.google.com/?q=53.189768,6.557384

Is the information correct?
1) Yes
2) No
#? 1
Disable MultiTech packet forwarder
Stopping lora-network-server: OK
Installing TTN Poly Packet Forwarder
Installing poly-packet-forwarder (2.1-r5) to root...
Configuring poly-packet-forwarder.
cp: can't stat '/var/config/lora/samples/global_conf_multitech-eu868-ttn.json':
No such file or directory
*****
A sample configuration has been supplied at /var/config/lora/global_conf.json
VERIFY this sample is valid for your location and UPDATE if required
*****

Get up-to-date TTN configuration for packet forwarder
Found lora card MTAC-LORA-868
Get up-to-date TTN configuration for packet forwarder
Starting ttn-packet-forwarder: OK
admin@mtcdt:~# █

```

Figura 41 Ventana de configuración del Paket Forwarder de TTN para el Gateway Multitech

La instalación es completada una vez que el mensaje “Starting ttn-packet-forwarder: OK” aparece en la pantalla.

4.5 Configuración del Nano Gateway Lopy

Para comenzar con la configuración del Nano Gateway es necesario actualizar la versión de Firmware disponible en la página de pycom bajo el link:

<https://docs.pycom.io/chapter/gettingstarted/installation/firmwaretool.html>

Una vez actualizado el Firmware, es necesario registro del Nanogateway en la nube siguiendo los pasos que se mencionan a continuación:

- 1.- Ir a la página de activación donde se le pedirá el inicio de sesión con la cuenta de The Things Network.
- 2.- El siguiente paso es ingresar el ID de la puerta de enlace y seleccionar el plan de frecuencia que usará la puerta de enlace. Es importante tomar en cuenta que una vez que se ingrese la identificación de la puerta de enlace, no podrá cambiarse, a menos que se reinicie el equipo al punto de configuración de fábrica.
- 3.- Para configurar el punto de acceso WiFi de la puerta de enlace, se debe establecer la conexión al punto WiFi llamado 'Things Gateway - xxxx' visualizado desde el buscador de redes del ordenador. La contraseña para este punto de acceso es 'the thigs'. Una vez que esté conectado el Wifi, se debe pulsar el botón "continuar" que se pondrá de color azul.
- 4.- En este paso es donde se selecciona cómo se desea que el Gateway se conecte a internet. La conexión será inalámbrica por lo tanto al seleccionar WiFi se debe ingresar el SSID de la red, seguido del modo de seguridad y la contraseña, y haga clic en 'Configurar Gateway'.
- 5.- Una vez dentro del Gateway se visualizará en los próximos momentos la página con el estado del mismo. Se necesitan entre 30 segundos y 2 minutos para establecer los ajustes de configuración, el navegador actualizará la página para volver a la conexión WiFi y, si todo se hace correctamente, el primer mensaje enviado a través de su puerta de enlace será desplegado.
- 6.- En el buscador Dentro del menú de The Things Network en el apartado "Console" existen dos opciones "applications" y "gateways", lo que permite dar de alta los dispositivos necesarios una vez que se ha creado un usuario en la nube.

Para continuar con la configuración en el apartado "Register Gateway" es necesario configurar lo siguiente:

Protocol = Packet Forwarder

Gateway EUI = *User Defined* (definido en el archivo config.py)

Description = *Descripción breve de la aplicación en minúsculas*

Frequency Plan = Seleccionar frecuencia de trabajo (ejemplo EU - 868 MHz)

Location = *Definido por el lugar*

Antenna Placement = Si la antena es interna o externa

El principio de funcionamiento del Nano Gateway, es un dispositivo de la marca Pycom que se basa en la transmisión de datos en un solo canal en una frecuencia de trabajo de 868 Mhz, funciona en tres archivos de programa que contienen un código dividido en estos mismos, main.py, config.py y nanogateway.py que le permiten conectarse a la red inalámbricamente y trabajar como packet forwarder y de esta forma conectarse con el servidor The Things Network.



Figura 42 Lopy Nano Gateway y el servidor TTN

Dentro del archivo Main (main.py) se ejecutan las funciones de arranque de Gateway llamando a la librería config.py una vez establecida la configuración el nano Gateway comienza a trabajar, el código de este archivo está descrito en el anexo II

En el archivo config.py está contenida la información de configuración del servidor, en este caso la nube y la red, dependiendo de la región a la que el nano Gateway esté conectado variarán los parámetros de configuración del mismo, en este caso varía de acuerdo a la región Europea en una frecuencia de trabajo antes mencionada de 868 Mhz (ver anexo II: archivo config.py)

Es importante cambiar el nombre y la clave de la red inalámbrica a la que se desea conectar el Gateway. El último archivo nanogateway.py controla la generación de los paquetes y su envío por la red LoRa, El resto del código se muestra en el anexo II.

El circuito electrónico cuenta con un Reed Switch que al momento de activarse por medio de un imán el modo de trabajo del nodo cambia de Multichannel (que es por defecto y utilizando el Gateway multitech) a Mono-Channel (utilizando el Gateway Lopy).

4.6 Configuración del portal Dweet.io

Este es un portal web que permite publicar los datos contenidos en la nube. Para publicar los datos de la nube y poder visualizarlos se deben seguir los siguientes pasos:

1. Se debe de crear un link en la plataforma Dweet.io introduciendo un dato, para ello en el navegador se escribe el nombre del nodo por ejemplo (cambiar la palabra *nanogateway* por el nombre con que se tiene registrado el gateway en The things Network):
<https://dweet.io/dweet/for/nanogateway?temperatura=20>
2. Abrir otra pestaña del navegador o utiliza un móvil para seguir el dato:
<https://dweet.io/follow/nanogateway>
3. Probar el funcionamiento añadiendo las variables a obtener, en este caso la temperatura y después la humedad, como se muestra en el siguiente link:
<https://dweet.io/dweet/for/nanogateway?temperatura=20&humedad=8>

Los datos recibidos del nodo son publicados en crudo, es decir en un formato “Raw Data”, estos son tomados de la nube The Things Network, después Dweet.io publica esta información en su plataforma.

4.7 Configuración de la aplicación Freeboard.

Una vez publicados los datos en Dweet.io la aplicación Freeboard se encarga de dar sentido a estos datos, de esta forma es posible visualizar de manera gráfica los datos enviados por el nodo disponibles en tiempo real.

De esta forma se puede ver de una manera sencilla y clara el mensaje que se ha recibido, el nivel de batería, la de temperatura, humedad y posición del nodo. Esta herramienta de visualización también permite ver el nombre de los dispositivos que han enviado el mensaje, el Gateway por el que se ha reenviado el mensaje, el canal, frecuencia y modulación empleada.

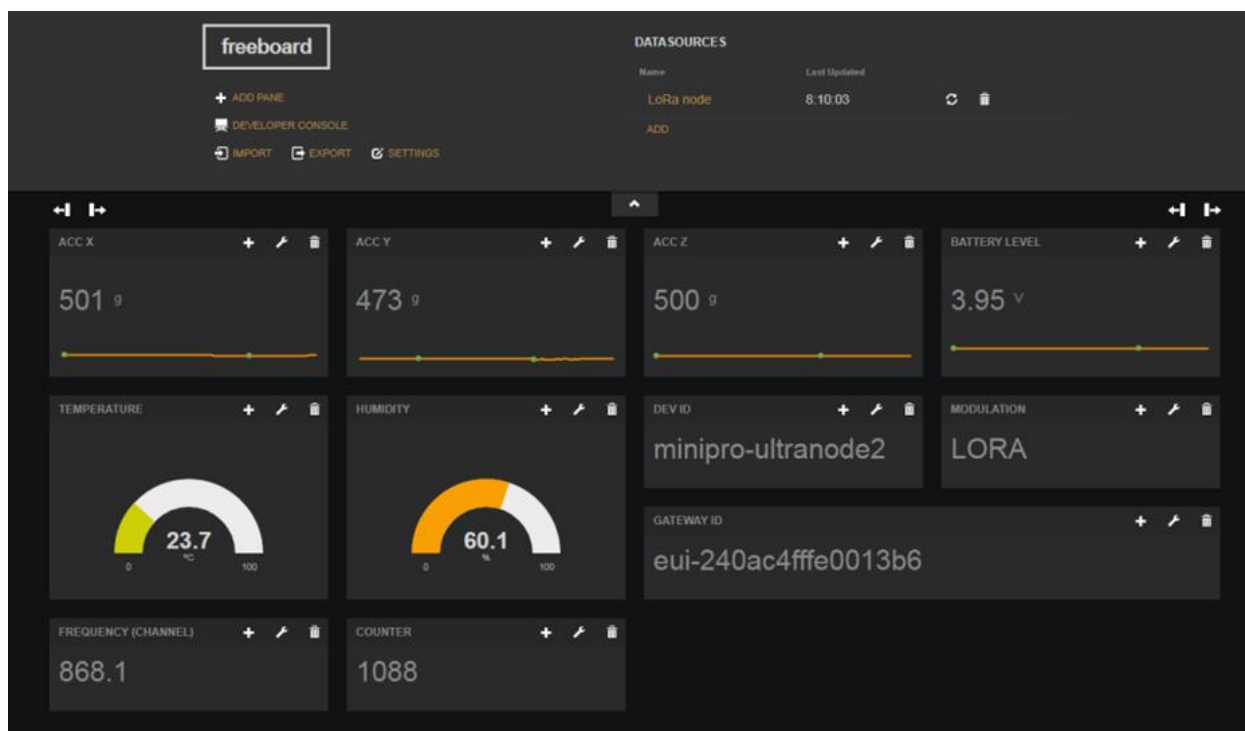


Figura 43 Panel de gestión del nodo sensor utilizando Freeboard

4.8 Pruebas de Alcance dentro de Ikerlan.

Para elaborar las pruebas de cobertura de LoRa se utilizó la herramienta TTN Mapper, esta herramienta permite cargar la información de cobertura como es la ubicación, RSSI y SNR directamente desde el backend de TTN. El objetivo de esta herramienta es proporcionar un mapa de la cobertura real de los gateways que se encuentran registrados en la nube.

La mínima información de localización que debe ser enviada por el nodo es la latitud y la longitud. En este caso no hay un módulo GPS integrado en el nodo sensor, por esta razón fue necesario integrar esta aplicación a un teléfono celular, la idea básica es que el nodo y el móvil se encuentren cerca el uno del otro, esto permite a la aplicación TTN Mapper medir la intensidad de señal del nodo y combinarla con la localización del móvil en el momento en que los paquetes son recibidos en la nube, la información se presenta como se observa en la figura 45.

Para estas pruebas se hizo un recorrido por todas las instalaciones de Ikerlan y así medir el alcance del sensor en base a la posición del Gateway Multitech ya instalado en el tejado y cambiando el tipo de antena para el nodo.

En la figura se observa la ubicación del Gateway Multitech, situado en el tejado en un extremo derecho del edificio Olandixo, que está a 152,61 m de altura, dado que hay 150,86 m en la horizontal y 15,13 m en vertical.



Figura 44 Instalaciones de Ikerlan en Arrasate.

En la siguiente figura se muestra la pantalla del móvil con la aplicación The Things Mapper que muestra el nombre del Gateway con el que se están haciendo las pruebas, su ubicación y la intensidad de señal del nodo:

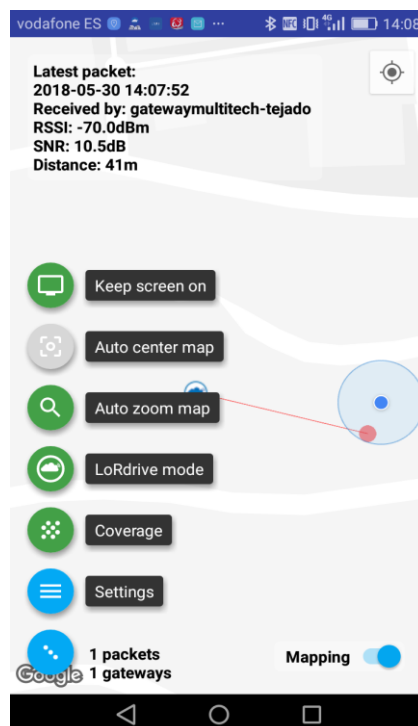


Figura 45 Pantalla de aplicación The Things Mapper

4.8.1 Pruebas de alcance con antena de hilo.

Como se puede observar la cobertura obtenida con la antena de hilo no es la esperada ya que del recorrido hecho dentro de las instalaciones de Ikerlan.

Con referencia al suelo puede observarse en la imagen que la cobertura del LoRa utilizando la antena de hilo no es tan favorable como se esperaba, ya que hay envío de mensajes del nodo, pero con un delay considerable de hasta 5 min, así como pérdida de mensajes. De igual manera fue mucho más difícil obtener los datos de mapeo en el primer piso y en un sitio alejado de las ventanas.

De acuerdo a estos datos se obtuvo un nivel máximo de potencia recibida de -73 dBm y un mínimo de potencia de -120 dBm en una distancia máxima alcanzada de 150 m. Cabe mencionar que la intensidad esperada para evitar pérdida de paquetes va de un rango de -59dBm a -70 dBm.

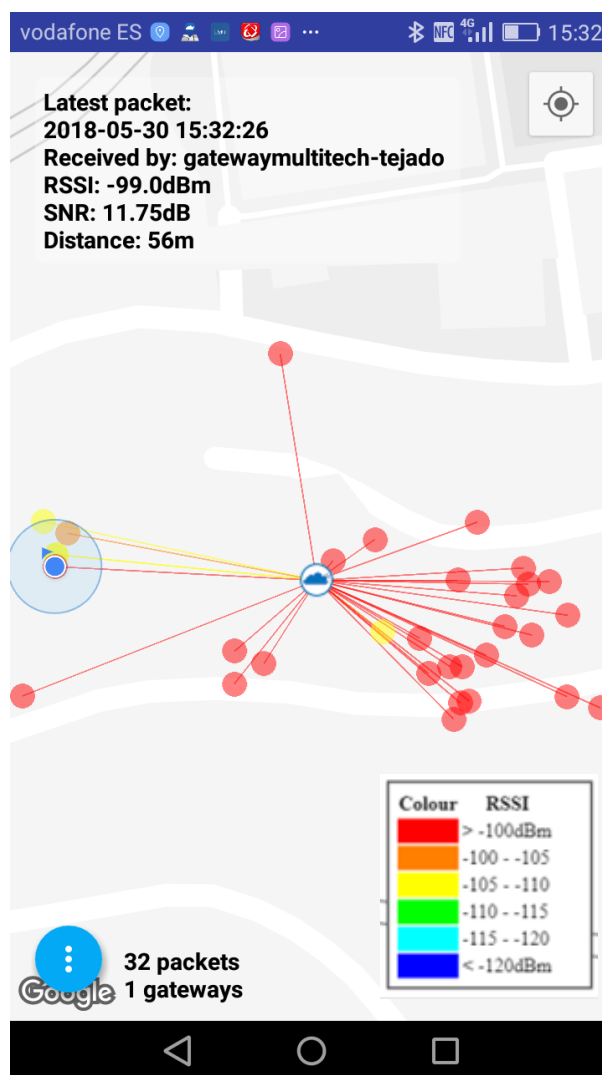


Figura 46 Resultado final pruebas de alcance con antena de Hilo

En la siguiente imagen se muestra la pérdida un paquete debido a la carencia de cobertura del nodo.

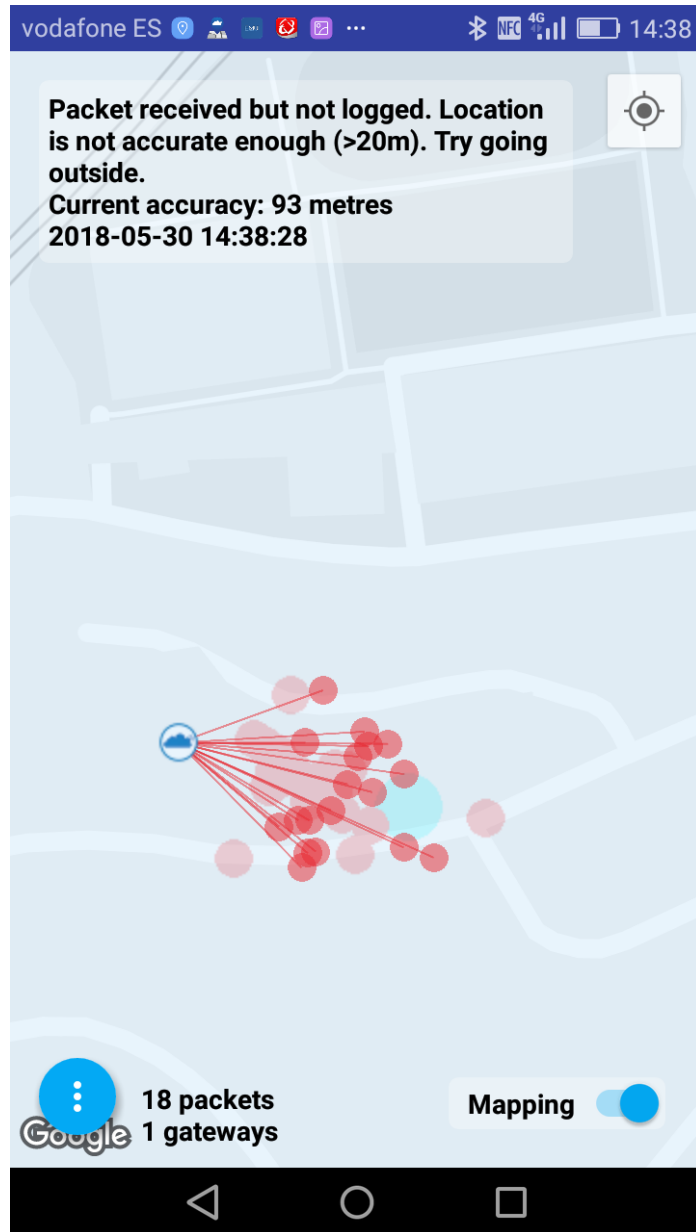


Figura 47 paquete 18 enviado pero no recibido por el Gateway

4.8.2 Pruebas de alcance con antena helicoidal.

La antena helicoidal alcanzó niveles de cobertura mejores de RSSI en comparación con los niveles de cobertura de la antena de hilo. Estos niveles variaron de entre -59dBm a -69 dBm, sin notar la pérdida de paquetes durante la prueba.

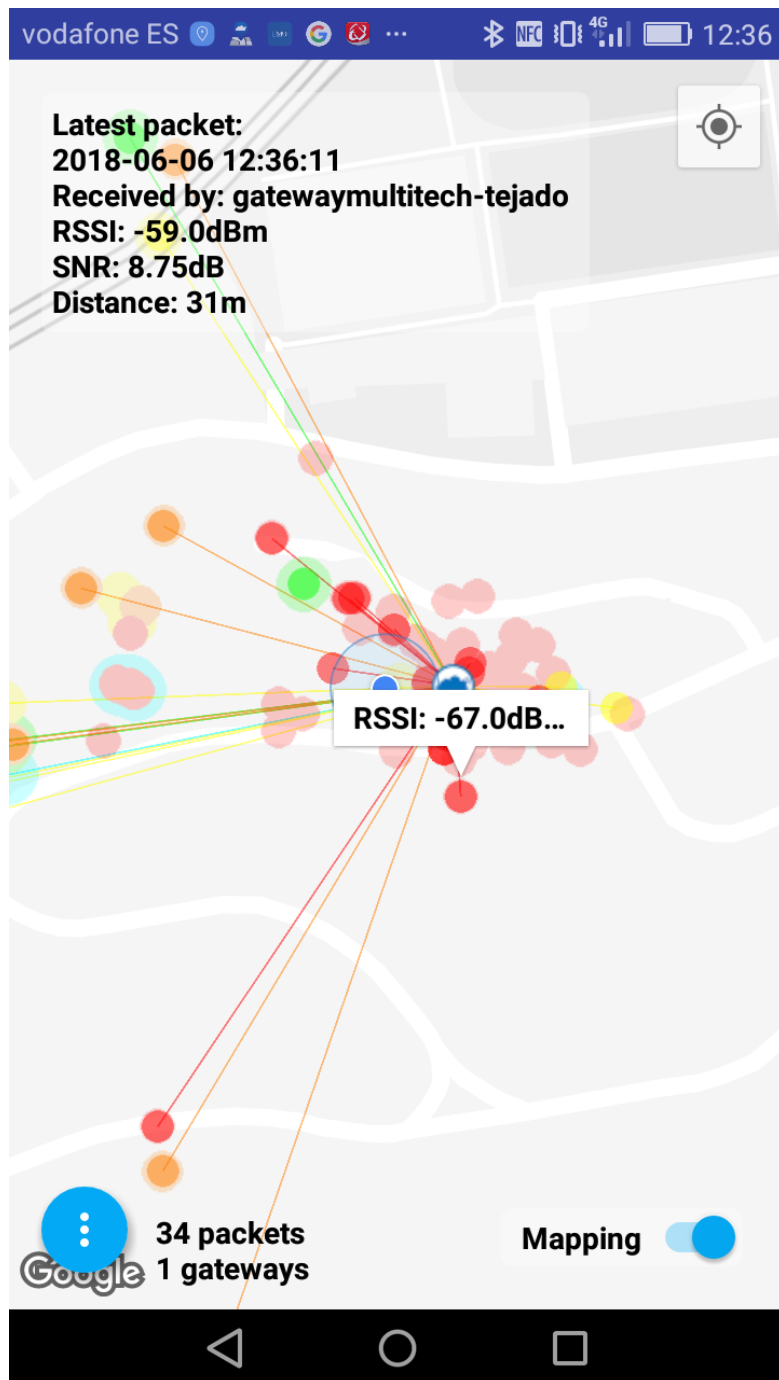


Figura 48 Resultado final pruebas de alcance con antena de helicoidal

5. DISEÑO MECÁNICO DE LA CARCASA

Otra de las fases a desarrollar en este proyecto fue el diseño 3D y fabricación de la carcasa para el nodo con un grado de protección IP67, este grado de protección está orientado a la industria y su índice de especificación es el más común para los sensores.

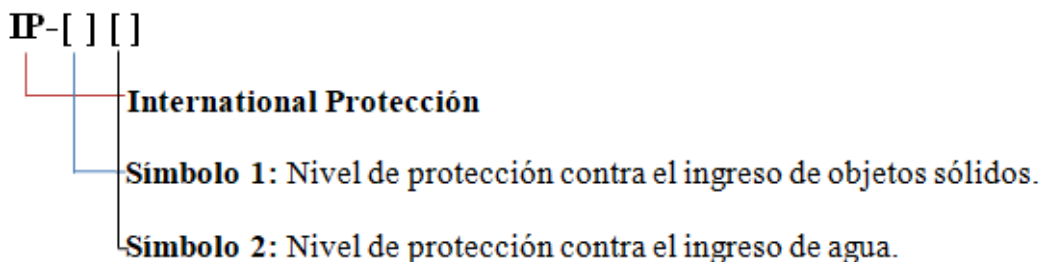
A medida que los componentes se van haciendo cada vez más pequeños en los ambientes industriales, dispositivos como los contenedores de nodos sensores tienen que cumplir con un grado de protección IP67, por lo que necesitan contar con propiedades mecánicas adicionales. Los diversos tipos de nodos sensores se encuentran en soluciones al aire libre en el mayor de los casos, es por eso que se ha seleccionado este grado de alta protección frente al polvo y el agua.

Uno de los retos de este proyecto fue que mediante la impresión 3D de esta carcasa, el molde sea capaz de cumplir con todos los requisitos que demanda el grado de protección IP67.

5.1. Grado de protección IP67

EL grado de protección IP hace referencia a la norma internacional CEI 60529 (Ingress Protection) utilizado con mucha frecuencia en los datos técnicos de equipamiento eléctrico o electrónico, en general de uso industrial como sensores, medidores, controladores, etc. Especifica un efectivo sistema para clasificar los diferentes grados de protección aportados a los mismos por los contenedores que resguardan los componentes que constituyen el equipo.

Nomenclatura CEI 60529.



La norma CEI 60529 establece para el primer dígito del equipo a ser certificado debe cumplir con alguna de las siguientes condiciones.

NIVEL	DESCRIPCIÓN
0	Sin protección
1	Protegido contra la entrada de elementos sólidos de hasta 50mm.
2	Protegido contra la entrada de elementos sólidos de hasta 12,5mm.
3	Protegido contra la entrada de elementos sólidos de hasta 2,5mm.
4	Protegido contra la entrada de elementos sólidos de hasta 1mm.
5	Protegido contra la entrada de polvo (la cantidad que entra no interfiere con el funcionamiento del dispositivo).
6	Totalmente protegido contra la entrada de polvo.

Tabla 6 Definición del primer dígito del grado de Protección IP 67

La norma CEI 60529 establece para el segundo dígito del equipo a ser certificado debe cumplir con alguna de las siguientes condiciones.

NIVEL	DESCRIPCIÓN
0	Sin protección
1	No debe entrar el agua cuando se la deja caer, desde 200mm de altura respecto del equipo, durante 10 minutos (a razón de 3-5mm ³ por minuto).
2	No debe entrar el agua cuando se la deja caer, durante 10 minutos (a razón de 3-5mm ³ por minuto). Dicha prueba se realizará 4 veces a razón de una por cada giro de 15° tanto en sentido vertical como horizontal, partiendo cada vez de la posición normal de trabajo.
3	No debe entrar el agua nebulizada en un ángulo de hasta 60° a derecha e izquierda de la vertical a un promedio de 11 litros por minuto y a una presión de 800-100 kN/m ² durante un tiempo que no sea menor a 5 minutos.
4	No debe entrar el agua arrojada desde cualquier ángulo a un promedio de 10 litros por minuto y a una presión de 800-100 KN/m ² durante un tiempo que no sea menor a 5 minutos.
5	No debe entrar el agua arrojada a chorro (desde cualquier ángulo) por medio de una boquilla de 6,3 mm de diámetro, a un promedio de 12,5 litros por minuto y a una presión 30 KN/m ² durante un tiempo que no sea menor a 3 minutos y a una distancia que no sea menor de 3 metros.
6	No debe entrar el agua arrojada a chorros (desde cualquier ángulo) por medio de una boquilla de 12,5 mm de diámetro, a un promedio de 100 litros por minuto y a una presión 100 KN/m ² durante un tiempo que no sea menor a 3 minutos y a una distancia que no sea menor de 3 metros.
7	El equipo debe soportar sin filtración alguna la inmersión completa a 1 metro durante 30 minutos.
8	El equipo debe soportar sin filtración alguna la inmersión completa y continua a la profundidad y durante el tiempo que especifique el fabricante del producto con el acuerdo del cliente, pero siempre que resulten condiciones más severas que las especificadas para el valor 7.

Tabla 7 Definición del segundo dígito del grado de Protección IP 67

Este estándar ha sido desarrollado para calificar de una manera alfa-numérica a equipamientos en función del nivel de protección que sus materiales contenedores le proporcionan contra la entrada de materiales extraños. Mediante la asignación de diferentes códigos numéricos, el grado de protección del equipamiento puede ser identificado de manera rápida y con facilidad tal y como se muestra en las tablas anteriores.

De esta manera, por ejemplo, se deduce que un grado de protección IP67 indica lo siguiente:

-Las letras "IP" identifican al estándar (del inglés: Ingress Protection).

-El valor "6" en el primer dígito numérico describe el nivel de protección ante el polvo, en este caso: el polvo no debe de entrar bajo ninguna circunstancia.

-El valor "7" en el segundo dígito numérico describe el nivel de protección frente a los líquidos (normalmente agua), en este caso: El objeto debe resistir (sin filtración alguna) la inmersión completa a 1 metro durante 30 minutos.

Como regla general se puede establecer que cuando mayor es el grado de protección IP, más protegido está el equipamiento.

Actualmente la mayoría de los sensores inductivos, capacitivos, y fotoeléctricos que se comercializan en el mercado tienen un nivel de protección mínimo de IP67, los cuales los hacen aptos para soportar la mayoría de los ambientes agresivos que se dan en la industria.

5.2. Selección de la junta tórica (Toroide)

Las juntas tóricas son elementos de estanqueidad circulares provistas de una sección transversal circular. La sencillez de su forma y las características elásticas de sus materiales confieren a la junta tórica una aplicabilidad universal y la convierten en el elemento de estanqueidad más utilizado. Las juntas tóricas pueden fabricarse de gran variedad de elastómeros y están normalizadas en la norma DIN ISO 3601.

Para definir las dimensiones de la junta tórica se determina el diámetro interior (d_1) y el espesor de la sección (d_2) como se puede observar en la figura 44 el material y la dureza. La junta tórica se utiliza como cierre dinámico o estático y trabaja, en general, en un alojamiento de sección rectangular.

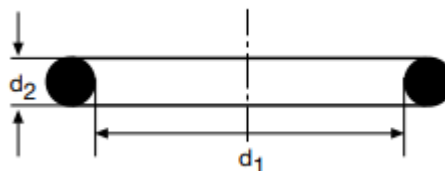


Figura 49 Diámetros interior y exterior de la junta Tórica

El cierre de un elemento está determinado por la fuerza radial de reacción F que resulta de sus dos componentes:

1. Fuerza radial (f) debida a la interferencia elástica (aplastamiento) de la sección de la junta tórica en el alojamiento (Figura 45)

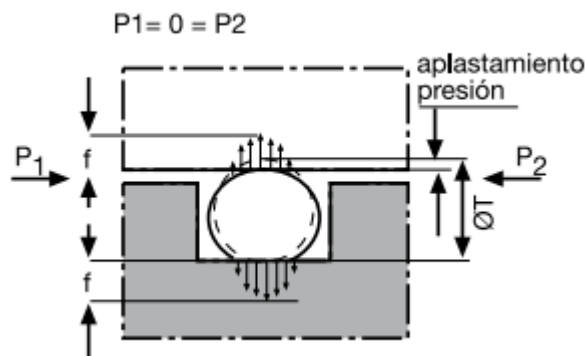


Figura 50 Interferencia elástica de la junta tórica.

2. Fuerza radial derivada de la activación de la presión de la pieza que hay que cerrar (f_1); en efecto, gracias al comportamiento del elastómero, el empuje de la presión del fluido determina una reacción elástica de la sección de la junta hacia las partes del contenedor contra las cuales se debe efectuar el cierre.

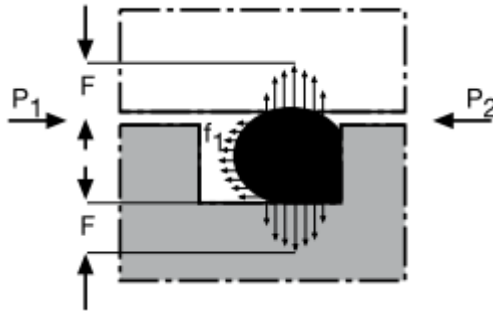


Figura 51 Fuerza radial derivada de la activación de la presión del medio.

La fuerza F depende de la sección de la junta tórica y de su nivel de compresión, de la dureza del elastómero y de la entidad de la presión del fluido que se debe retener.

$$P_1 > P_2$$

Es importante notar que, para el cierre del contenedor, la fricción desarrollada en las superficies y la entrada de posibles fluidos o polvo depende de la fuerza F , esta fuerza decrece con el tiempo en función de la deformación permanente, característica típica de cualquier elastómero. Para un elastómero determinado, la deformación permanente está estrechamente unida al nivel de calidad del elastómero en sí. De esto deriva la exigencia de valorar apropiadamente el nivel de calidad de la junta tórica para conseguir el mayor éxito en su aplicación.

Por lo tanto la junta tórica seleccionada es de un diámetro externo de 64 mm y un grosor de 1mm, cuyo material es de caucho de nitrilo resistente al calor NBR (NBR Nomenclatura del material según DIN ISO 1629)

5.3. Diseño del molde

5.3.1. Estanqueización estática, compresión radial

En este tipo de montaje la sección transversal de la junta tórica se comprime en sentido radial. La mecanización del alojamiento de la junta tórica se debe hacer o bien en la parte interior o bien en la parte exterior. En el caso del molde para este proyecto la ranura se encuentra en la parte interior.

Cuando las partes del contenedor a estanqueizar no ejecutan movimientos relativos entre ellas se refiere a la estanqueización estática, las juntas tóricas se prestan idealmente para este tipo de estanqueización y por ende se ha seleccionado una junta tórica para el cierre hermético de la carcasa nodo sensor.

5.4. Impresión 3D de la carcasa

Como paso final se llevó a cabo la impresión la carcasa del nodo sensor a base de una resina Epóxica utilizando uno de los métodos de impresión 3D más comunes en la actualidad.

5.4.1. Proceso de estereolitografía.

La estereolitografía (SLA) es una tecnología de fabricación aditiva, comúnmente denominada impresión 3D o SL. Al igual que con cualquier técnica de impresión 3D, se requiere un archivo digital en 3D. Esto se puede obtener a través del software CAD (SolidWorks, Sculpt o Maya, por ejemplo).

Entre los diversos componentes de una máquina de estereolitografía se encuentran una bandeja de resina, una plataforma móvil (eje Z), un sistema de raspado (eje X), un láser UV, óptica de enfoque y un espejo galvanométrico (ejes X e Y).

El rayo láser barre la superficie de la resina líquida de acuerdo con el modelo 3D digital suministrado a la impresora. Una vez que la primera capa de material solidificado, la plataforma desciende un nivel, que corresponde al grosor de una capa de impresión, y una nueva sección se solidifica. Hay tantos ciclos de impresión como capas para obtener el volumen completo de la pieza.

En Ikerlan la máquina de estereolitografía realiza la producción del modelo de abajo hacia arriba, es decir la plataforma se sumerge en la bandeja de resina después de cada capa solidificada mientras que el láser actúa de abajo hacia arriba. Convierte materiales líquidos en partes sólidas, haciéndolo capa por capa, cuidándolos selectivamente utilizando una fuente de luz en un proceso llamado fotopolimerización.

La estereolitografía se utiliza ampliamente para crear modelos, prototipos, patrones y piezas de producción para una amplia gama de industrias, desde ingeniería y diseño de productos hasta fabricación, odontología, joyería, fabricación de modelos y educación.

Este proceso es un procedimiento de fabricación aditiva que utiliza un receptáculo de fotopolímeros líquidos de resinas epóxicas foto sensibles (una sustancia que solidifica a la exposición a un láser ultravioleta) para estereolitografía. El modelo 3D se construye capa por capa, en una plataforma móvil. El láser toca el envase solidificando las piezas necesarias para lograr la creación del prototipo SLA.

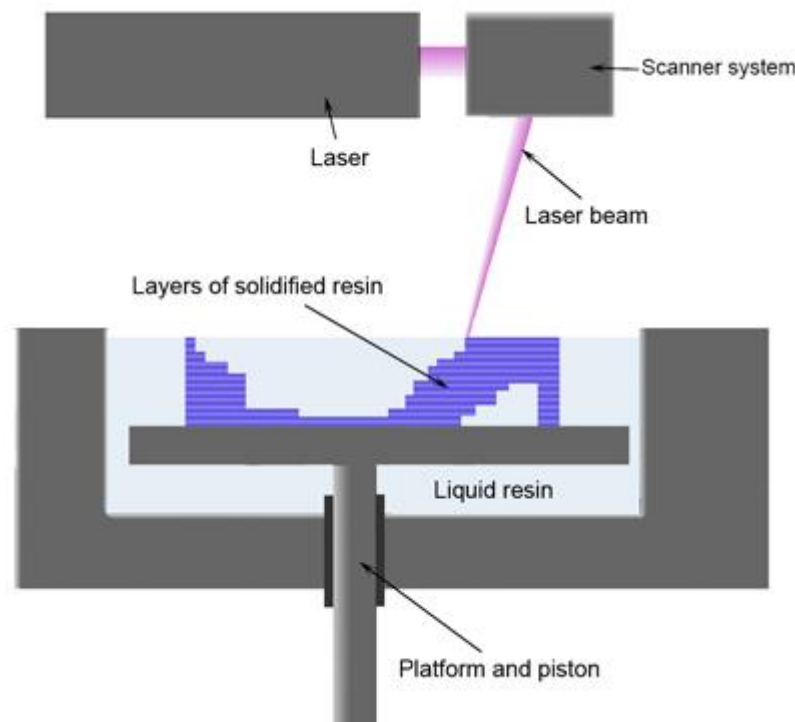


Figura 52 Proceso de estereolitografía por capas

Cuando se completa una capa, la plataforma móvil sobre la que se colocan las capas sólidas se baja una distancia que es igual al espesor de cada capa de material que está siendo utilizado (por lo general entre 0,05 mm y 0,15 mm).

Esto expone una superficie líquida una vez más, lo que permite que el láser inicie el procedimiento nuevamente, creando así una nueva capa que se coloca directamente sobre la de abajo. Los períodos de fabricación para los procedimientos que implican la estereolitografía varían de acuerdo con el tamaño y la

complejidad del objeto que está siendo producido, con diferencias que van desde unas pocas horas a poco más de 24 horas sin procesamiento. Para este proyecto fueron necesarias 24 horas.

5.4.2. Posprocesamiento de la carcasa

Todas las tecnologías de impresión 3D llevan un pos procesamiento, que puede ir desde el desprendimiento de la pieza y una ligera limpieza, hasta horas de trabajo.

Una vez que el proceso de impresión ha sido completado, el objeto 3D impreso se enjuaga con una solución química para limpiarlo del exceso de resina. Después de esto, se lo "cocina" en un horno ultravioleta para endurecerlo.

Para la tecnología SLA se comienza quitando el modelo y remover los restos de resina, para esto se debe colocar el modelo en un baño de alcohol isopropílico de una manera delicada utilizando guantes en todo momento, esto es necesario para remover el exceso de resina no solidificada.

A diferencia de otras técnicas tales como la sinterización selectiva por láser (SLS), la Deposición de Material Fundido (FDM) y la impresión PolyJet 3D, se requiere de este post-procesamiento para finalizar el proceso de fotopolimerización y conseguir la máxima fuerza del material.

La estereolitografía utiliza soportes al imprimir figuras complejas en forma de andamios, que permiten soportar las partes que se precipitan en el vacío, después de la limpieza del modelo se deben remover los soportes de forma manual o con ayuda de alguna herramienta sencilla, pues es sencillo remover estos soportes.

A medida que se forma cada capa, los monómeros de resina reaccionan y forman enlaces covalentes que proporcionan altos grados de resistencia lateral, pero la reacción de polimerización no se lleva a término; más bien, el proceso de impresión se modula de una manera que mantiene la capa en un estado semi-reaccionado llamado el "estado verde". Este estado difiere del estado completamente curado de una manera muy importante: todavía hay grupos polimerizables en la superficie a los que las capas posteriores pueden unirse covalentemente.

La isotropía tiene beneficios mecánicos y ópticos. Las piezas isotrópicas son ideales para aplicaciones de creación de prototipos de ingeniería, ya que reflejan el rango de propiedades asociadas con los termoplásticos moldeados por inyección comunes y no sufren delaminación o porosidad superficial.

El proceso de unión química y la falta de líneas de capas visibles dentro de las piezas permiten la impresión de piezas ópticamente transparentes. A continuación se muestra la impresión de la carcasa final para el nodo, desarrollada en Solid Works e impresa a base de una resina epóxica.



Figura 53 Carcasa para el nodo sensor

Como ya se mencionó la ranura para la junta tórica se encuentra en la parte intermedia de la carcasa y cuenta con 6 barrenos que permiten el cierre del nodo desde afuera.

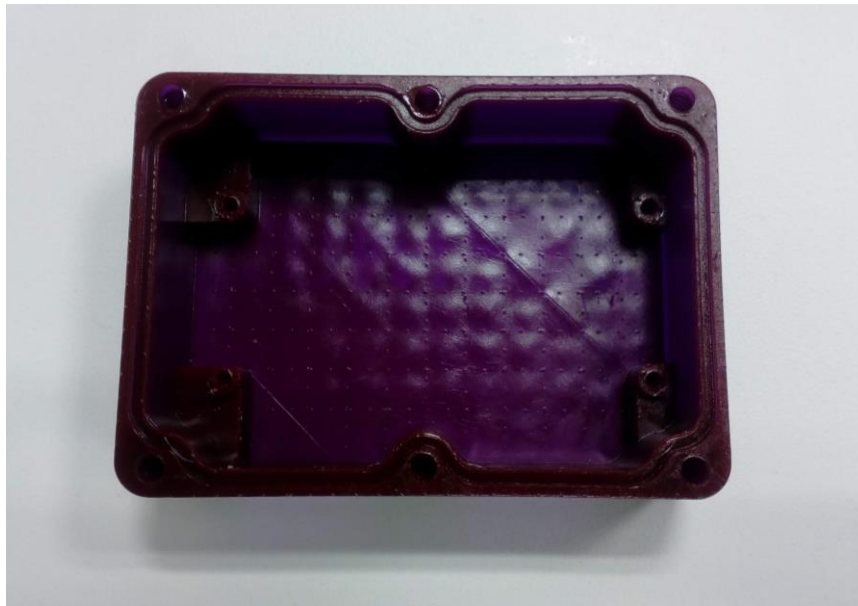


Figura 54 ranura para la junta tórica de 1mm

Dando como resultado una carcasa de medidas: 70 mm x 45mm x 40 mm



Figura 55 carcasa completa para el nodo sensor

Para comprobar el buen funcionamiento de la carcasa se realizaron pruebas sumergiendo la carcasa en un recipiente con agua, aunque el fin del desarrollo del nodo sensor no es estar sumergido uno de los requisitos del grado de protección IP-67 es que el agua no se filtre dentro de la carcasa provocando daños a la electrónica del nodo. Por lo tanto, por un lapso de 30 min exactamente, se sumergió el nodo como se muestra a continuación:



Figura 56 Vista interna del nodo sensor 1

Por dentro el nodo sensor tiene el sistema embebido con la antena de hilo dentro de la carcasa como se muestra a continuación:



Figura 57 Nodo sumergido en agua por 30 min

Gracias a la buena selección de la junta tórica fue posible lograr el funcionamiento del nodo bajo el agua, es importante mencionar que el fin de esta aplicación no es trabajar con el nodo sumergido todo el tiempo, únicamente se hicieron estas pruebas para validar el diseño de la carcasa ya que el nodo estará trabajando en exteriores donde posiblemente habrá lluvia o nieve en algún momento.

Líneas futuras y mejoras.

Con el nuevo diseño de los sensores y la arquitectura de la red LoRa, se pretende hacer una nueva red en las instalaciones de Ikerlan en San Sebastián, que utilice únicamente un Gateway monocal, ya que es más barato y la red de nodos no será mayor a 5 elementos.

Los datos del nodo sensor son enviados al servidor The Things Network y se quedan almacenados únicamente durante 7 días, sin embargo, no se tiene una base de datos para la gestión adecuada de esta información y puede perderse. Por esto se sugiere el uso de otra herramienta para la gestión de los datos obtenidos por el nodo que permita tener un historial de los mismos.

Se pretende también hacer un diseño de carga inalámbrica para el nodo a través de otros tipos de fuente de alimentación que no sea una batería Li-Po, sino una mejora en el diseño de una batería de carga completamente inalámbrica reduciendo la intervención humana al mínimo.

Conclusiones

El diseño de un nodo sensor inalámbrico para transmisión de datos usando la tecnología LoRa, cumplió con todas sus características y objetivos establecidos, uno de las cuales fue la de diseñar un nodo de tamaño pequeño, más pequeño de los existentes en el mercado, el alcance de este nodo fue hasta de 8 km de distancia. Se sugiere el rediseño de la colocación de la antena, en una posición donde no se encuentre ubicada sobre el plano de masa del nodo, las pérdidas generadas pueden mejorarse si la antena se coloca en otro sitio.

En cuanto al diseño mecánico con capacidad IP67, el proceso de impresión 3D utilizado es uno de los más antiguos en el mercado y también uno de los más caros, por tanto en este sentido se sugiere buscar otro método de impresión 3D que permita obtener las mismas propiedades de la carcasa a un precio más económico.

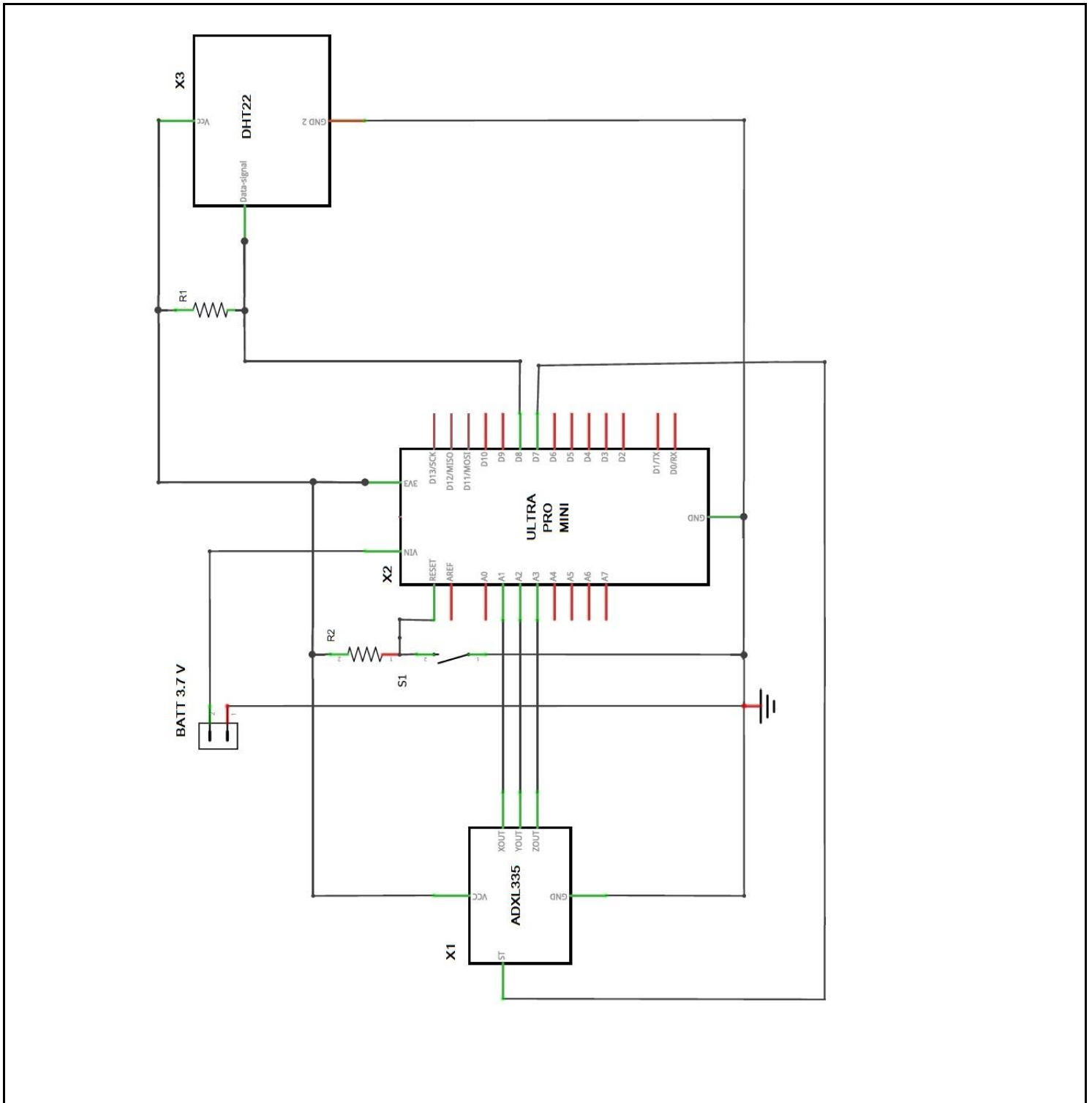
Es de suma importancia apegarse a las normas de diseño mecánico, ya que para la primera carcasa diseñada la distancia entre barrenos fue más larga de la necesaria, esto provocó la reingeniería de la carcasa, esta vez donde la distancia entre los barrenos no excediera $\frac{2}{3}$ partes de la longitud total de la carcasa por lo que fue necesaria la colocación de un par de barrenos centrales y así asegurar el grado de protección contra el agua.

BIBLIOGRAFÍA

- [1] LinkLabs. *A comprehensive look at Low Power Wide Area Networks*. Network Configuration, 3: 16, 3-5, 2016
- [2] LoRaAlliance. *LoRaWAN What is it?*, 7: 20, 7-11, 2015
- [3] Semtech. *Wireless Gas monitoring*, 2-3,2018
- [4] Semtech. *IoT_Enabled Smart parking mnagement.*, 1-3, 2018.
- [5] F Adelantado. *Understanding the Limits of LoRaWAN*. Overview of LPWAN and Cellular Technologies for IoT, 2: 7, 1-5, 2017.
- [6] Gemalto, Actility and Semtech. *LoRaWAN Security*. Full end-to-end encryption for IoT application providers, 4: 3, 2-4, 2007.
- [7] The Things Network, *Learn how to grow the network*. Disponible via. <http://www.thethingsnetwork.org>. Acceso el 15/02/20108
- [8] Juntas Bisma. *Juntas tóricas*, España, 2003.

ANEXO I

Diagrama Electrónico



**MÁSTER EN INGENIERÍA MECATRÓNICA
- TRABAJO FIN DE MÁSTER -**

	FECHA	NOMBRE	FIRMA	Diseño de un nodo sensor para aplicaciones IoT
Proyectado				
Dibujado	19/06/2019	Paula Estrada		
Comprobado	19/06/2019	Miguel A. José		
ESCALA SIN	Diagrama electrónico			PLANO Nº 1
				Sustituye a
				Sustituido por

ANEXO II

Código de configuración del Nodo Sensor


```

//Librerías
#include <lmic.h>
#include <hal/hal.h>
#include <SPI.h>
#include <Wire.h>
#include <RTCZero.h>
#include <SerialFlash.h>
#include <DHT.h>

#define Serial SerialUSB

// Definición de los parámetros de registro de la nube TTN
#define EUI64_CHIP_ADDRESS 0x50
#define EUI64_MAC_ADDRESS 0xF8
#define EUI64_MAC_LENGTH 0x08
#define MAX_DATA_SIZE 12

// Parámetros para el sensor de temperatura (pines de trabajo)
#define DHTTYPE DHT22
#define DHTPIN 2 // Sensor Pin
#define DHTPIN 9 // Sensor Pin
DHT dht(DHTPIN, DHTTYPE);

#define GW_CONFIGURATION_PIN 4

//Parámetros para acelerómetro
#define X_AXIS A1
#define Y_AXIS A2
#define Z_AXIS A3
#define ADXL335_PWR_PIN 7
#define ADXL335_SELF_TEST_PIN 8

// Parámetro para la obtención de voltaje
#define BATTERY_VOLTAGE A5

bool gw_multichannel_mode;

// Credenciales para inicio de sesión en la red
RTCZero rtc;
// Este EUI debe estar en formato little-endian, por lo que primero debe de
// ir el byte menos significativo, esto significa invertir los bytes. Para los
// EUI emitidos por TTN, los últimos bytes deben ser 0xD5, 0xB3,
// 0x70.
//static const u1_t PROGMEM APPEUI[8] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0xD5,
//0xB3, 0x70 };
//static const u1_t PROGMEM APPEUI[8] = { 0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x00,
//0x80, 0x17 };
static const u1_t PROGMEM APPEUI[8] = { 0xA4, 0xA7, 0x00, 0xD0, 0x7E, 0xD5,
0xB3, 0x70 };

void os_getArtEui (u1_t* buf) {
  memcpy_P(buf, APPEUI, 8);
}

// Dev EUI en information Little endian.
u1_t DEVEUI[EUI64_MAC_LENGTH];
void os_getDevEui (u1_t* buf) {
  memcpy(buf, DEVEUI, EUI64_MAC_LENGTH);
}

```

```

// Esta clave debe estar en formato big endian (o, dado que no es realmente
// un número sino un bloque de memoria). //En la práctica, una clave tomada de
// ttnctl se puede copiar así como está.
//La clave que se muestra aquí es la clave predeterminada de semtech.
//static const ul_t PROGMEM APPKEY[16] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
//0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
static const ul_t PROGMEM APPKEY[16] = { 0x44, 0xBB, 0x19, 0x31, 0x6E, 0xEA,
0x38, 0x53, 0x97, 0x4C, 0xDE, 0x7E, 0xF6, 0x6C, 0x09, 0xFC };
void os_getDevKey (ul_t* buf) {
  memcpy_P(buf, APPKEY, 16);
}

static uint8_t data[MAX_DATA_SIZE];
static osjob_t sendjob;

// Para programar el envío de datos (TX) cada tantos segundos (podría
// alargarse debido a las limitaciones del ciclo de trabajo).
const unsigned TX_INTERVAL = 900;

// Declaración de los PINES para la tarjeta mini Ultra PRO
//IMPORTANTE, sin esta configuración no se podrán enviar datos a la nube TTN.
const lmic_pinmap lmic_pins = {
  .nss = 5,
  .rxtx = LMIC_UNUSED_PIN,
  .rst = 3,
  .dio = {2, 6, LMIC_UNUSED_PIN},
};

// Definición de nodo mono canal
int channel = 0;
int dr = DR_SF7;

// Disables all channels, except for the one defined above, and sets the
// data rate (SF). This only affects uplinks; for downlinks the default
// channels or the configuration from the OTAA Join Accept are used.
//
// Not LoRaWAN compliant; FOR TESTING ONLY!
//
void forceTxSingleChannelDr() {
  for(int i=0; i<9; i++) { // For EU; for US use i<71
    if(i != channel) {
      LMIC_disableChannel(i);
    }
  }
  // Ajuste la velocidad de datos (SF) y la potencia de transmisión para el
  // enlace ascendente o envío (uplink)
  LMIC_setDrTxpow(dr, 14);
}
void onEvent (ev_t ev)
{
  Serial.print(os_getTime());
  Serial.print(": ");
  switch (ev) {
    case EV_SCAN_TIMEOUT:
      Serial.println(F("EV_SCAN_TIMEOUT"));
      break;
    case EV_BEACON_FOUND:
      Serial.println(F("EV_BEACON_FOUND"));
      break;
    case EV_BEACON_MISSED:
      Serial.println(F("EV_BEACON_MISSED"));
      break;
    case EV_BEACON_TRACKED:

```

```

    Serial.println(F("EV_BEACON_TRACKED"));
    break;
case EV_JOINING:
    Serial.println(F("EV_JOINING"));
    break;
case EV_JOINED:
    Serial.println(F("EV_JOINED"));

if (gw_multichannel_mode)
{
    //LMIC_setupChannel(0, 868100000, DR_RANGE_MAP(DR_SF12, DR_SF7),
BAND_CENTI); // g-band
    LMIC_setupChannel(1, 868300000, DR_RANGE_MAP(DR_SF12, DR_SF7B),
BAND_CENTI); // g-band
    LMIC_setupChannel(2, 868500000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI);
// g-band
    LMIC_setupChannel(3, 867100000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI);
// g-band
    LMIC_setupChannel(4, 867300000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI);
// g-band
    LMIC_setupChannel(5, 867500000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI);
// g-band
    LMIC_setupChannel(6, 867700000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI);
// g-band
    LMIC_setupChannel(7, 867900000, DR_RANGE_MAP(DR_SF12, DR_SF7), BAND_CENTI);
// g-band
    LMIC_setupChannel(8, 868800000, DR_RANGE_MAP(DR_FSK, DR_FSK), BAND_MILLI);
// g2-band

}
else
{
// Ignorar los canales del join accepted y utilizar sólo un canal
forceTxSingleChannelDr();

}

// Deshabilitar la validación de verificación de enlace (habilitado
//automáticamente)
LMIC_setLinkCheckMode(0);
break;
case EV_RFU1:
    Serial.println(F("EV_RFU1"));
    break;
case EV_JOIN_FAILED:
    Serial.println(F("EV_JOIN_FAILED"));
    break;
case EV_REJOIN_FAILED:
    Serial.println(F("EV_REJOIN_FAILED"));
    break;
case EV_TXCOMPLETE:
    Serial.println(F("EV_TXCOMPLETE (includes waiting for RX windows)"));
    if (LMIC.txrxFlags & TXRX_ACK)
        Serial.println(F("Received ack"));
    if (LMIC.dataLen) {
        Serial.println(F("Received "));
        Serial.println(LMIC.dataLen);
        Serial.println(F(" bytes of payload"));
    }
}

```

```

// Sección para asegurar que todos los mensajes de depuración se envíen
antes de pasar a modo Sleep
Serial.flush();

// Modo Sleep en el periodo definido TX_INTERVAL utilizando un single shot
alarm
rtc.setAlarmEpoch(rtc.getEpoch() + TX_INTERVAL);
rtc.enableAlarm(rtc.MATCH_YMMDDHMMSS);
rtc.attachInterrupt(alarmMatch);
// Deshabilitar puerto USB durante el modo sleep para ahorro de energía
USBDevice.detach();
// Para entrar a modo SLEPP
rtc.standbyMode();
// Reinitialize USB for debugging
USBDevice.init();
USBDevice.attach();
Serial.println(F("USB attach"));

// Programa la próxima transmisión para ser enviada inmediatamente después
de esto
os_setTimedCallback(&sendjob, os_getTime() + sec2osticks(1), do_send);
break;
case EV_LOST_TSYNC:
Serial.println(F("EV_LOST_TSYNC"));
break;
case EV_RESET:
Serial.println(F("EV_RESET"));
break;
case EV_RXCOMPLETE:
// data received in ping slot
Serial.println(F("EV_RXCOMPLETE"));
break;
case EV_LINK_DEAD:
Serial.println(F("EV_LINK_DEAD"));
break;
case EV_LINK_ALIVE:
Serial.println(F("EV_LINK_ALIVE"));
break;
default:
Serial.println(F("Unknown event"));
break;
}
}
// Información a enviar
void do_send(osjob_t* j)
{
unsigned char counter;
float batteryVoltage; // Variable para nivel de Batería
int adcReading; // Variable para lectura de la Batería
int voltage, accX, accY, accZ; // Variables de posición para el sensor

float t = dht.readTemperature(); // Variable para la temperatura
Serial.print(F("Temperature: ")); Serial.print(t); Serial.println(F(" °C"));
float h = dht.readHumidity(); // Variable para la humedad
Serial.print(F("Humidity: ")); Serial.print(t); Serial.println(F(" %"));
int32_t temp = t * 10;
int32_t humidity = h * 10;

digitalWrite(LED_BUILTIN, HIGH);
digitalWrite(ADXL335_PWR_PIN, HIGH); // Encendido del acelerómetro ADXL335

// Para verificar que no haya una transmisión o recepción (TX/RX)
ejecutándose
if (LMIC.opmode & OP_TXRPEND)

```



```

{
  Serial.println(F("OP_TXRXPEND, not sending"));
}
else
{
  // ***** Battery monitor connection
  //
  // VBAT-----1M-----3M3-----GND
  //      |
  //      ---0.1uF---GND
  //      |
  //      A5

  adcReading = analogRead(BATTERY_VOLTAGE); // Discard inaccurate 1st reading
  adcReading = 0;
  // Perform averaging
  for (counter = 10; counter > 0; counter--) adcReading +=
analogRead(BATTERY_VOLTAGE);
  adcReading = adcReading / 10;
  batteryVoltage = adcReading * (4.3 / 1023.0); // Convert to volts
  Serial.print(F("Battery: ")); Serial.print(batteryVoltage);
  Serial.println(F(" V"));

  accX = analogRead(X_AXIS); // Discard inaccurate 1st reading
  adcReading = 0;
  for (counter = 10; counter > 0; counter--) adcReading +=
analogRead(X_AXIS); // Perform averaging
  Serial.print(F("X_AXIS: ")); Serial.print(accX); Serial.println(F(" g"));

  accY = analogRead(Y_AXIS); // Discard inaccurate 1st reading
  adcReading = 0;
  for (counter = 10; counter > 0; counter--) adcReading +=
analogRead(Y_AXIS); // Perform averaging
  Serial.print(F("Y_AXIS: ")); Serial.print(accY); Serial.println(F(" g"));

  accZ = analogRead(Z_AXIS); // Discard inaccurate 1st reading
  adcReading = 0;
  for (counter = 10; counter > 0; counter--) adcReading +=
analogRead(Z_AXIS); // Perform averaging
  Serial.print(F("Z_AXIS: ")); Serial.print(accZ); Serial.println(F(" g"));

  // Arreglo para envío de parámetros Voltaje, Temperatura y Humedad.
  voltage = batteryVoltage * 100;
  data[0] = voltage >> 8; data[1] = voltage;
  data[2] = temp>> 8; data[3] = temp;
  data[4] = humidity>> 8; data[5] = humidity;
  data[6] = accX>> 8; data[7] = accX;
  data[8] = accY>> 8; data[9] = accY;
  data[10] = accZ>> 8; data[11] = accZ;

  // Prepare upstream data transmission at the next possible time.
  LMIC_setTxData2(1, data, sizeof(data), 0);
  Serial.println(F("Packet queued"));
}
// Siguiete transmisión TX programada después del evento TX_COMPLETE.

digitalWrite(ADXL335_PWR_PIN, LOW); // ADXL335 OFF
digitalWrite(LED_BUILTIN, LOW);
}

void setDevEui(unsigned char* buf)
{
  Wire.begin();
  Wire.beginTransmission(EUI64_CHIP_ADDRESS);

```

```

Wire.write(EUI64_MAC_ADDRESS);
Wire.endTransmission();
Wire.requestFrom(EUI64_CHIP_ADDRESS, EUI64_MAC_LENGTH);

// Format needs to be little endian (LSB...MSB)
while (Wire.available())
{
  *buf-- = Wire.read();
}
}

void setup()
{
  delay(5000);
  int count;
  unsigned char pinNumber;

  // ***** Poner los pines inutilizados en estado conocido*****
  pinMode(0, INPUT_PULLUP);
  pinMode(1, INPUT_PULLUP);
  // D7-D13, A0 (D14)-A5 (D19), SDA (D20), SCL (D21), MISO (D22)
  for (pinNumber = 7; pinNumber <= 22; pinNumber++)
  {
    pinMode(pinNumber, INPUT_PULLUP);
  }
  // RX_LED (D25) & TX_LED (D26) (both LED not mounted on Mini Ultra Pro)
  pinMode(25, INPUT_PULLUP);
  pinMode(26, INPUT_PULLUP);
  // D30 (RX) & D31 (TX) of Serial
  pinMode(30, INPUT_PULLUP);
  pinMode(31, INPUT_PULLUP);
  // D34-D38 (EBDG Interface)
  for (pinNumber = 34; pinNumber <= 38; pinNumber++)
  {
    pinMode(pinNumber, INPUT_PULLUP);
  }
  // ***** Fin de la inicialización del estado de los pines no utilizados.
  *****

  // Proporción de tierra y corriente utilizando las entradas analógicas como
  pines digitales normales.
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(ADXL335_PWR_PIN, OUTPUT); digitalWrite(ADXL335_PWR_PIN, LOW);
  pinMode(ADXL335_SELF_TEST_PIN, OUTPUT); digitalWrite(ADXL335_SELF_TEST_PIN,
LOW); // ADXL335 SELFTEST OFF

  pinMode(GW_CONFIGURATION_PIN, INPUT_PULLUP);
  if (digitalRead(GW_CONFIGURATION_PIN) == HIGH) gw_multichannel_mode = true;
  else gw_multichannel_mode = false;
  Serial.println(gw_multichannel_mode);

  setDevEui(&DEVEUI[EUI64_MAC_LENGTH - 1]);
  while (!Serial && millis() < 10000);
  Serial.begin(115200);
  Serial.println(F("Starting"));
  Serial.print(F("DEVEUI: "));

  for (count = EUI64_MAC_LENGTH; count > 0; count--)
  {
    Serial.print("0x");
    if (DEVEUI[count - 1] <= 0x0F) Serial.print("0");
    Serial.print(DEVEUI[count - 1], HEX);
  }
}

```

```

    Serial.print(" ");
}
Serial.println();

// Inicializar serial flash
SerialFlash.begin(4);
// Configurar serial flash en modo sleep
SerialFlash.sleep();

// Inicializar RTC
rtc.begin();
// Usar RTC como segundo timer
rtc.setEpoch(0);

//Configurar los canales utilizados por The Things Network, que corresponde a
//los valores predeterminados de la mayoría de las puertas de enlace. Sin
//esto, solo se usan tres canales base de la especificación LoRaWAN, lo que
//ciertamente funciona, por lo que es bueno para la depuración, pero puede
//sobrecargar esas frecuencias, hay que asegurarse de configurar el rango
//completo de frecuencias de la red aquí (a menos que la red los configure
//automáticamente).
//La configuración de los canales debe ocurrir después de LMIC_setSession, ya
que configura un conjunto mínimo de canales.

// LMIC init
os_init();
// Restablecer el estado de MAC. Sesiones y transferencias de datos
pendientes serán descartadas.
LMIC_reset();
LMIC_setClockError(MAX_CLOCK_ERROR * 1 / 100);

// Hace que LMIC inicialice los canales predeterminados, elija un canal y
 programe la activación por aire OTAA
LMIC_startJoining();

// LMIC ya habrá decidido enviar en uno de los 3 canales predeterminados;
Asegúrate de que usa el que queremos, esto para modo monocanal.

LMIC.txChnl = channel;

// Inicia el envío (al enviar automáticamente empieza también la activación
OTAA)
do_send(&sendjob);
}

//Ciclo de repetición de envío
void loop() {
    os_runloop_once();
}

void alarmMatch()
{
}

```

Código para Gateway Lopy:

main.c

```

""" LoPy LoRaWAN Nano Gateway example usage Ikerlan """

import config
from nanogateway import NanoGateway

if __name__ == '__main__':
    nanogw = NanoGateway(
        id=config.GATEWAY_ID,
        frequency=config.LORA_FREQUENCY,
        datarate=config.LORA_GW_DR,
        ssid=config.WIFI_SSID,
        password=config.WIFI_PASS,
        server=config.SERVER,
        port=config.PORT,
        ntp_server=config.NTP,
        ntp_period=config.NTP_PERIOD_S
    )

    nanogw.start()
    nanogw._log('You may now press ENTER to enter the REPL')
    input()

```

config.py

```

""" LoPy LoRaWAN Nano Gateway configuration Ikerlan """

import machine
import ubinascii

WIFI_MAC = ubinascii.hexlify(machine.unique_id()).upper()
# Set the Gateway ID to be the first 3 bytes of MAC
address + 'FFFF' + last 3 bytes of MAC address
GATEWAY_ID = WIFI_MAC[:6] + "FFFF" + WIFI_MAC[6:12]

SERVER = 'router.eu.thethings.network'
PORT = 1700

NTP = "pool.ntp.org"
NTP_PERIOD_S = 3600

WIFI_SSID = 'CIoT_001'
WIFI_PASS = 'CIoT_001'

# for EU868
LORA_FREQUENCY = 868100000
LORA_GW_DR = "SF7BW125" # DR_5
LORA_NODE_DR = 5

```

nanogateway.py

```

""" LoPy Nano Gateway class """

from network import WLAN
from network import LoRa
from machine import Timer
import os
import ubinascii
import machine
import json
import time
import errno
import thread
import socket

PROTOCOL_VERSION = const(2)

PUSH_DATA = const(0)
PUSH_ACK = const(1)
PULL_DATA = const(2)
PULL_ACK = const(4)
PULL_RESP = const(3)

TX_ERR_NONE = "NONE"
TX_ERR_TOO_LATE = "TOO_LATE"
TX_ERR_TOO_EARLY = "TOO_EARLY"
TX_ERR_COLLISION_PACKET = "COLLISION_PACKET"
TX_ERR_COLLISION_BEACON = "COLLISION_BEACON"
TX_ERR_TX_FREQ = "TX_FREQ"
TX_ERR_TX_POWER = "TX_POWER"
TX_ERR_GPS_UNLOCKED = "GPS_UNLOCKED"

STAT_PK = {"stat": {"time": "", "lati": 0,
                    "long": 0, "alti": 0,
                    "rxnb": 0, "rxok": 0,
                    "rxfw": 0, "ackr": 100.0,
                    "dwnb": 0, "txnb": 0}}

RX_PK = {"rxpk": [{"time": "", "tmst": 0,
                  "chan": 0, "rfch": 0,
                  "freq": 868.1, "stat": 1,
                  "modu": "LORA", "datr": "SF7BW125",
                  "codr": "4/5", "rssi": 0,
                  "lsnr": 0, "size": 0,
                  "data": ""}]}

TX_ACK_PK = {"txpk_ack":{"error":""}}

class NanoGateway:

    def __init__(self, id, frequency, datarate, ssid, password, server, port,
ntp='pool.ntp.org', ntp_period=3600):
        self.id = id
        self.frequency = frequency
        self.sf = self._dr_to_sf(datarate)
        self.ssid = ssid

```

```

self.password = password
self.server = server
self.port = port
self.ntp = ntp
self.ntp_period = ntp_period

self.rxnb = 0
self.rxok = 0
    self.rxfw = 0
    self.dwnb = 0
    self.txnb = 0

self.stat_alarm = None
    self.pull_alarm = None
    self.uplink_alarm = None

self.udp_lock = _thread.allocate_lock()

self.lora = None
self.lora_sock = None

def start(self):
    # Change WiFi to STA mode and connect
    self.wlan = WLAN(mode=WLAN.STA)
    self._connect_to_wifi()

    # Get a time Sync
    self.rtc = machine.RTC()
    self.rtc.ntp_sync(self.ntp, update_period=self.ntp_period)

    # Get the server IP and create an UDP socket
    self.server_ip = socket.getaddrinfo(self.server, self.port)[0][-1]
    self.sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM,
socket.IPPROTO_UDP)
    self.sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    self.sock.setblocking(False)

    # Push the first time immediately
    self._push_data(self._make_stat_packet())

    # Create the alarms
    self.stat_alarm = Timer.Alarm(handler=lambda t:
self._push_data(self._make_stat_packet()), s=60, periodic=True)
    self.pull_alarm = Timer.Alarm(handler=lambda u: self._pull_data(), s=25,
periodic=True)

    # Start the UDP receive thread
    _thread.start_new_thread(self._udp_thread, ())

    # Initialize LoRa in LORA mode
    self.lora = LoRa(mode=LoRa.LORA, frequency=self.frequency,
bandwidth=LoRa.BW_125KHZ, sf=self.sf,
        preamble=8, coding_rate=LoRa.CODING_4_5, tx_iq=True)
    # Create a raw LoRa socket
    self.lora_sock = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
    self.lora_sock.setblocking(False)
    self.lora_tx_done = False

    self.lora.callback(trigger=(LoRa.RX_PACKET_EVENT | LoRa.TX_PACKET_EVENT),
handler=self._lora_cb)

def stop(self):
    # TODO: Check how to stop the NTP sync
    # TODO: Create a cancel method for the alarm

```

```

# TODO: kill the UDP thread
self.sock.close()

def _connect_to_wifi(self):
    self.wlan.connect(self.ssid, auth=(None, self.password))
    while not self.wlan.isconnected():
        time.sleep(0.5)
    print("WiFi connected!")

def _dr_to_sf(self, dr):
    sf = dr[2:4]
    if sf[1] not in '0123456789':
        sf = sf[:1]
    return int(sf)

def _sf_to_dr(self, sf):
    return "SF7BW125"

def _make_stat_packet(self):
    now = self.rtc.now()
    STAT_PK["stat"]["time"] = "%d-%02d-%02d %02d:%02d:%02d GMT" % (now[0],
now[1], now[2], now[3], now[4], now[5])
    STAT_PK["stat"]["rxnb"] = self.rxnb
    STAT_PK["stat"]["rxok"] = self.rxok
    STAT_PK["stat"]["rxfw"] = self.rxfw
    STAT_PK["stat"]["dwnb"] = self.dwnb
    STAT_PK["stat"]["txnb"] = self.txnb
    return json.dumps(STAT_PK)

def _make_node_packet(self, rx_data, rx_time, tmst, sf, rssi, snr):
    RX_PK["rxpk"][0]["time"] = "%d-%02d-%02dT%02d:%02d:%02d.%dZ" %
(rx_time[0], rx_time[1], rx_time[2], rx_time[3], rx_time[4], rx_time[5],
rx_time[6])
    RX_PK["rxpk"][0]["tmst"] = tmst
    RX_PK["rxpk"][0]["datr"] = self._sf_to_dr(sf)
    RX_PK["rxpk"][0]["rssi"] = rssi
    RX_PK["rxpk"][0]["lsnr"] = float(snr)
    RX_PK["rxpk"][0]["data"] = ubinascii.b2a_base64(rx_data)[-1]
    RX_PK["rxpk"][0]["size"] = len(rx_data)
    return json.dumps(RX_PK)

def _push_data(self, data):
    token = os.urandom(2)
    packet = bytes([PROTOCOL_VERSION]) + token + bytes([PUSH_DATA]) +
ubinascii.unhexlify(self.id) + data
    with self.udp_lock:
        try:
            self.sock.sendto(packet, self.server_ip)
        except Exception:
            print("PUSH exception")

def _pull_data(self):
    token = os.urandom(2)
    packet = bytes([PROTOCOL_VERSION]) + token + bytes([PULL_DATA]) +
ubinascii.unhexlify(self.id)
    with self.udp_lock:
        try:
            self.sock.sendto(packet, self.server_ip)
        except Exception:
            print("PULL exception")

def _ack_pull_rsp(self, token, error):
    TX_ACK_PK["txpk_ack"]["error"] = error
    resp = json.dumps(TX_ACK_PK)

```

```

    packet = bytes([PROTOCOL_VERSION]) + token + bytes([PULL_ACK]) +
    ubinascii.unhexlify(self.id) + resp
    with self.udp_lock:
        try:
            self.sock.sendto(packet, self.server_ip)
        except Exception:
            print("PULL RSP ACK exception")

def _lora_cb(self, lora):
    events = lora.events()
    if events & LoRa.RX_PACKET_EVENT:
        self.rxnb += 1
        self.rxok += 1
        rx_data = self.lora_sock.recv(256)
        stats = lora.stats()
        self._push_data(self._make_node_packet(rx_data, self.rtc.now(),
stats.timestamp, stats.sf, stats.rssi, stats.snr))
        self.rxfw += 1
    if events & LoRa.TX_PACKET_EVENT:
        self.txnb += 1
        lora.init(mode=LoRa.LORA, frequency=self.frequency,
bandwidth=LoRa.BW_125KHZ,
        sf=self.sf, preamble=8, coding_rate=LoRa.CODING_4_5, tx_iq=True)

def _send_down_link(self, data, tmst, datarate, frequency):
    self.lora.init(mode=LoRa.LORA, frequency=frequency,
bandwidth=LoRa.BW_125KHZ,
        sf=self._dr_to_sf(datarate), preamble=8,
coding_rate=LoRa.CODING_4_5,
        tx_iq=True)
    while time.time() < tmst:
        pass
    self.lora_sock.send(data)

def udp_thread(self):
    while True:
        try:
            data, src = self.sock.recvfrom(1024)
            _token = data[1:3]
            _type = data[3]
            if _type == PUSH_ACK:
                print("Push ack")
            elif _type == PULL_ACK:
                print("Pull ack")
            elif _type == PULL_RESP:
                self.dwnb += 1
                ack_error = TX_ERR_NONE
                tx_pk = json.loads(data[4:])
                tmst = tx_pk["txpk"]["tmst"]
                t_us = tmst - time.time() - 5000
                if t_us < 0:
                    t_us += 0xFFFFFFFF
                if t_us < 20000000:
                    self.uplink_alarm = Timer.Alarm(handler=lambda x:
self._send_down_link(ubinascii.a2b_base64(tx_pk["txpk"]["data"]),
tx_pk["txpk"]["tmst"] - 10,
tx_pk["txpk"]["datr"],
int(tx_pk["txpk"]["freq"] *
1000000)), us=t_us)
                else:
                    ack_error = TX_ERR_TOO_LATE
                    print("Downlink timestamp error!, t_us:", t_us)
                self._ack_pull_rsp(_token, ack_error)
            print("Pull rsp")

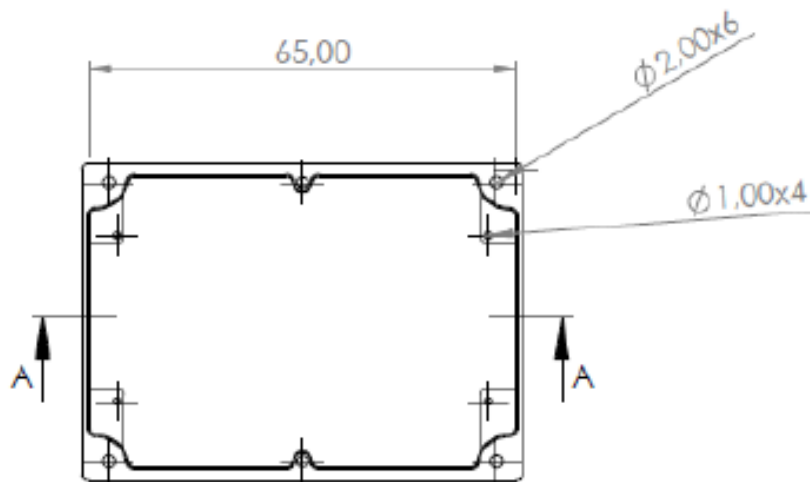
```



```
except socket.timeout:
    pass
except OSError as e:
    if e.errno == errno.EAGAIN:
        pass
    else:
        print("UDP recv OSError Exception")
except Exception:
    print("UDP recv Exception")
# Wait before trying to receive again
time.sleep(0.025)
```


ANEXO III

Planos de carcasa (Diseño mecánico)



SECCIÓN A-A

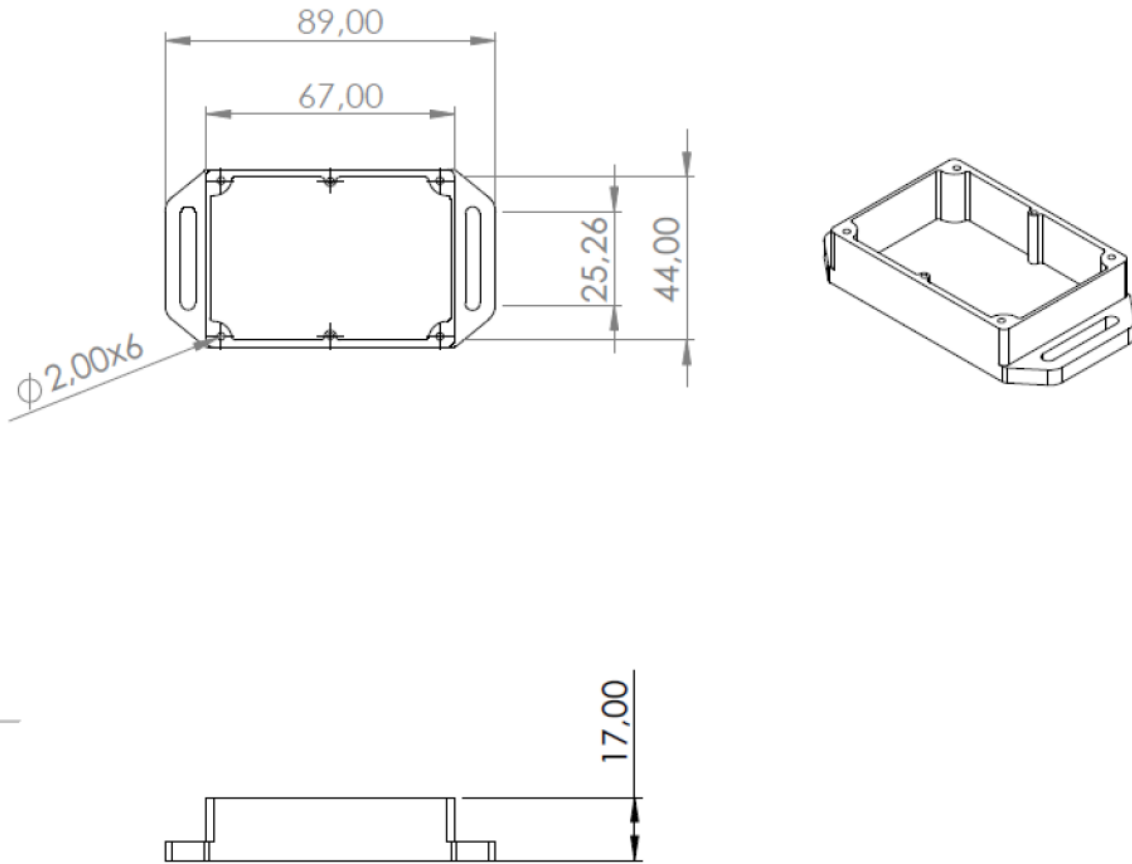


DETALLE B
ESCALA 2 : 1

NOTA: MATERIAL RESINA EPÓXICA

**MÁSTER EN INGENIERÍA MECATRÓNICA
- TRABAJO FIN DE MÁSTER -**

	FECHA	NOMBRE	FIRMA	Diseño de un nodo sensor para aplicaciones IoT
Proyectado				
Dibujado	19/06/2019	Paula Estrada		
Comprobado	19/06/2019	Miguel A. José		
ESCALA 2:1	Tapa de la carcasa			PLANO N° 2 Sustituye a Sustituido por



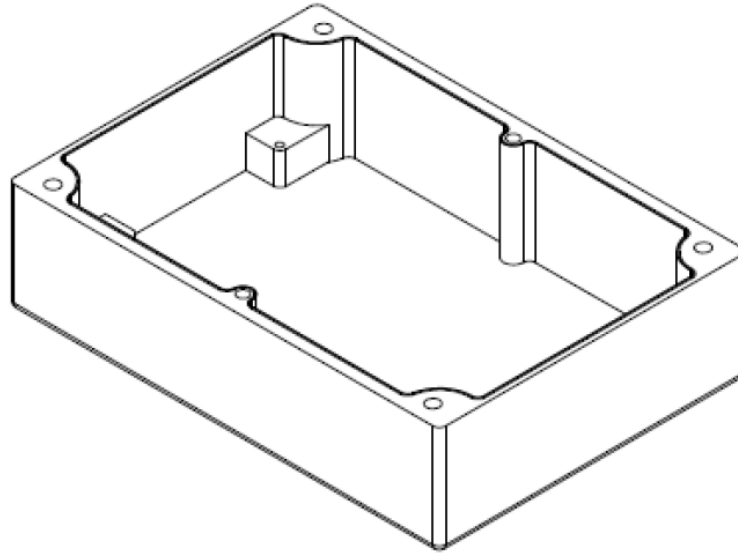
NOTA: MATERIAL RESINA EPÓXICA

**MÁSTER EN INGENIERÍA MECATRÓNICA
- TRABAJO FIN DE MÁSTER -**

	FECHA	NOMBRE	FIRMA
Proyectado			
Dibujado	19/06/2019	Paula Estrada	
Comprobado	19/06/2019	Miguel A. José	

**Diseño de un nodo sensor para
aplicaciones IoT**

ESCALA 2:1	Base de la carcasa	PLANO N° 3
		Sustituye a
		Sustituido por



NOTA: MATERIAL RESINA EPÓXICA

**MÁSTER EN INGENIERÍA MECATRÓNICA
- TRABAJO FIN DE MÁSTER -**

	FECHA	NOMBRE	FIRMA	Diseño de un nodo sensor para aplicaciones IoT
Proyectado				
Dibujado	19/06/2019	Paula Estrada		
Comprobado	19/06/2019	Miguel A. José		
ESCALA 2:1	Isométrico de la tapa			PLANO N° 4
				Sustituye a
				Sustituido por

ANEXO IV

Costos

RECURSO	PRECIO POR UNIDAD (€)	CANTIDAD	COSTE TOTAL (€)
Multitech Multichannel Conduit Gateway	779,50	1	779,50
MiniUltraPro Card (Rocket Planet)	34,90	2	69,8
Tarjeta Pycom LoPy Gateway	46,15	1	46,15
Cable de antena Nodo	10,31	1 m	10,31
Antena helicoidal	0,15	4	0,60
Sensor DHT22	9,95	2	19,90
Reed Switch	2,28	2	4,56
Acelerómetro ADXL335	13,38	2	26,76
Resistencia 4.7 K	1,2	6	7,2
Batería LiPo 3.7 V	12,90	2	25,8
Soldador	8,2	1	8,2
Estaño	4,5	100 g	4,5
Antena IP67 para Gateway	10,79	2	21,58
Resina Epóxica	32,90	710 ml	32,90
Impresión 3D Carcasa	150,0	4 pzas	600,0
Junta tórica 64 mm x 1mm	3,0	20 pzas	3,0
Sueldo becario (hora trabajada)	5.25	960	5040,0
Suma Total			6713,07