Tesis de grado

Red de sensado inalámbrico para riego en frutihorticultura

Ingeniería en Telecomunicaciones

Diego Barquet

Joaquín Frechero

Tutor: Dr. Joel Gak

Noviembre 2021 - Montevideo, Uruguay

Índice

Índice 2

Capítulo 1 - Introducción 4

1.1. Resumen 4

1.2. Problemática 5

1.3. Plan de acción 5

1.4. Estructura del documento 6

Capítulo 2 - Estado del Arte 7

2.1. Tecnologías IoT 7

2.1.1. Cellular IoT 7

2.1.2. Sigfox 9

2.1.3. ¿Por qué LoRaWAN? 12

2.2. LoRa y LoRaWAN 12

2.2.1. LoRa 13

2.2.1.1. Parámetros Regionales de Frecuencia 14

2.2.1.2. AU915 16

2.2.2. LoRaWAN 17

2.2.2.1. Ventajas de LoRaWAN 17

2.2.2.2. LoRa Alliance 18

2.2.2.3. Arquitectura 19

2.2.3. OTAA vs ABP 20

2.2.3.1. OTAA 22

2.2.3.2. ABP 24

2.2.3.3. ¿Qué método es conveniente utilizar? 25

Capítulo 3 - Elección de los componentes de la solución 27

3.1. Elección de la placa de desarrollo 27

3.2. Elección del gateway 28

3.3. Investigación y elección de sensores 29

3.3.1 Humedad del suelo 30

3.3.2. Sensores resistivos, capacitivos, FDR (Frequency Domain Reflectometry) y TDR (Time Domain Reflectometry) 31

3.3.3. Sensor de humedad de suelo Vegetronix VH400 34

3.3.4. Sensor de temperatura del suelo DS18B20 36

3.3.5. Sensor de temperatura y humedad relativa del aire DHT21 (AM2301) 37
3.3.6. Integración de los sensores digitales 37
Capítulo 4 - Nodo integrado 38
4.1. Consumo energético 38
4.2. Autonomía del nodo 42
4.3. Armado del nodo 42
4.4 Costos 48
4.5. Comparación con otras soluciones en el mercado 49
Capítulo 5 - Desarrollo y plataformas 52
5.1. Software 52
5.2. The Things Network 55
5.3. TagoIO 57
Capítulo 6 - Implementación, resultados obtenidos y conclusiones 61
6.1. Pruebas de campo 61
6.2. Resultados obtenidos de la prueba de campo 68
6.3. Conclusiones 73
Bibliografía 75
Comparación con otras soluciones en el mercado: 79
Anexos 81

Capítulo 1 - Introducción

1.1. Resumen

Este proyecto consistió en diseñar y desarrollar una red de sensado inalámbrico para obtención de datos periódicos de las condiciones de humedad y temperatura de suelo y aire. Ésta fue aplicada en el agro con el objetivo de ayudar a tomar decisiones de riego, particularmente en el sector frutihortícola uruguayo. Como protocolo de comunicación inalámbrico se utilizó la tecnología LoRaWAN (Low Power Wide Area Network / Red de área amplia de baja potencia) por sus ventajas de alcance e independencia de la cobertura móvil en el campo, donde comúnmente se desarrolla la actividad agrícola. La solución consta de un gateway, el nodo, o concentrador de sensores, y la plataforma de visualización de los datos.

Para comenzar, el plan fue seleccionar las partes necesarias para diseñar y desarrollar dicha red, teniendo un especial énfasis en la viabilidad económica: placas de desarrollo, sensores, módulos de comunicación inalámbrica y componentes de electrónica. El siguiente paso fue la integración de éstas, armando de un mínimo producto viable, para dejarlo funcionando y midiendo un período de tiempo en exterior. Se pretende lograr una herramienta tecnológica accesible y de fácil uso para el productor uruguayo, que redunde en mejoras productivas con un uso más eficiente de los recursos como el agua.

Los datos medidos son humedad y temperatura tanto en el aire como en el suelo ya que son los datos críticos para la toma de decisiones de riego. También se medirá el voltaje en la batería para monitorear la autonomía energética así como el nivel de señal recibida por el gateway. Una vez armada la red se pueden anexar diversos tipos de sensores, ya que al ser los dispositivos finales no afectan el diseño de la red, generando una solución muy escalable.

1.2. Problemática

La adaptación de nuevas tecnologías en actividades agrícolas es tendencia a nivel mundial [1]. Existen diversas soluciones de sensado remoto con tecnología IoT (Internet of Things / Internet de las Cosas), las cuales se volvieron cada vez más viables con el desarrollo de protocolos de comunicación inalámbrica de muy bajo consumo energético y precios de la electrónica cada vez más bajos.

El problema que da origen a este proyecto es la optimización del riego: hacer un uso más eficiente del agua, mejorar los rendimientos productivos y permitir que productores locales puedan realizar una agricultura de precisión. La agricultura demanda el 70% del agua extraída en el mundo [2] y de esta agua, aproximadamente un 44% se desperdicia [3]. En el mercado exterior existen algunas soluciones para la obtención de datos de humedad del suelo. Las más destacadas serán mencionadas en este documento. También hay empresas que se dedican a proporcionar recomendaciones de riego a través de Big Data (recopilación masiva de datos procesada por medio de algoritmos), las cuales mediante imágenes satelitales y pronósticos del tiempo indican al productor cuándo y cuánto regar según su cultivo [4].

Uruguay, donde en su mayoría se siguen utilizando técnicas de cultivo rudimentarias, de a poco se va adentrando en el mundo de la agricultura de precisión, pero los costos del mercado con soluciones integrales ("*llave en mano*") extranjeras lo impiden para productores pequeños o de mediana escala.

1.3. Plan de acción

Tras el desconocimiento del equipo en agricultura, se inició contacto con ingenieros agrónomos y con el Instituto Nacional de Investigación Agropecuaria (INIA) [5] de Las Brujas, cuya misión es generar y adaptar conocimientos y tecnologías para contribuir al desarrollo sostenible del sector agropecuario y del país, teniendo en cuenta las políticas de Estado, la inclusión social y las demandas de los mercados y de los consumidores. En particular dentro del INIA, nos pusimos en contacto con la Unidad de Agro-clima y Sistemas de Información

(GRAS), encargados de la investigación y el desarrollo de sistemas modernos de información y soporte para la toma de decisiones [6]. Se realizaron una serie de intercambios y reuniones donde, desde un inicio, expresaron su interés en la propuesta, brindaron conocimientos en cuanto a cómo se manejan los productores y facilitaron sus instalaciones en Las Brujas para implementar el piloto. Además, recomendaron acotar el mercado objetivo al sector frutihortícola, y sugirieron que en principio el foco podría ser en los invernaderos donde se tiene un control más riguroso de las condiciones del cultivo. Ellos validan la viabilidad del proyecto, reconociéndose como una solución tecnológica que redundaría en grandes beneficios para el sector.

En cuanto a la visualización y almacenamiento de los datos obtenidos por los sensores, se evaluaron diferentes alternativas posibles. La primera es realizar un desarrollo propio de la aplicación, generando un histórico de los datos y visualización en gráficas. Se entendió que esto excede al alcance de este proyecto y se limita a alcanzar un producto mínimo viable al diseño y desarrollo del nodo y de la red y a la obtención de los datos mediante plataformas gratuitas que facilitan el procesamiento y el análisis [7].

1.4. Estructura del documento

En este documento se detallarán los componentes que integran la solución lograda, el proceso de investigación y armado, así como las tecnologías utilizadas para la misma. Se organiza de la siguiente manera: en el capítulo 2 se desarrolla el estado del arte de las tecnologías IoT de largo alcance y se ahonda aún más en LoRaWAN, la tecnología utilizada, explicando sus generalidades; el capítulo 3 presenta los componentes elegidos para la solución y sus características (placa de desarrollo, componentes electrónicos, batería, gateway y sensores); el capítulo 4 se dedica a explicar el proceso de integración de las partes que forman el nodo, se mencionan las medidas tomadas para lograr minimizar el consumo energético, se presenta una tabla con los costos de la solución y se realiza una comparación con otras soluciones en el mercado extranjero; el capítulo 5 se enfoca en la parte de software: el desarrollo del firmware, se explica qué es y cómo se utiliza The Things Network y se presenta la plataforma IoT utilizada; el capítulo 6 está dedicado a la implementación, las pruebas de campo realizadas en la estación experimental de INIA Las Brujas donde se instala

por dos meses la solución comentando y analizando los resultados obtenidos. Por último, se expresan las conclusiones del trabajo, así como la bibliografía utilizada y una sección de anexos con material de interés (parte del código desarrollado, especificaciones de los equipos utilizados, tablas, etc.).

Capítulo 2 - Estado del Arte

2.1. Tecnologías IoT

En la actualidad existen diversas tecnologías para implementar el Internet de las Cosas de largo alcance, de las cuales las más conocidas y utilizadas son: cellular IoT (NB-IoT [Narrow Band IoT] y LTE-M [Machine Type Communication]), y dentro de las LPWAN (Low-Power Wide Area Network) SigFox y LoRaWAN [8]. Las mismas se explicarán brevemente a continuación así como la elegida, LoRaWAN, a través de un desarrollo más en detalle.

2.1.1. Cellular IoT

Las redes celulares son las más utilizadas para brindar conectividad a gran número de dispositivos. Para 2024, se prevé que habrá 4.150 millones de dispositivos IoT conectados de esta manera en todo el mundo [9].

En los últimos años las redes celulares de cuarta generación 4G/LTE se han expandido en grandes áreas, con cobertura NB-IoT en todos los centros poblados del Uruguay y casi un 80% del territorio nacional [10]. NB-IoT y LTE-M ambas funcionan coexistiendo con la red LTE para datos móviles, por lo que para su implementación no es necesario la instalación de nueva infraestructura, como antenas o radiobases.

<u>NB-IoT</u>

Narrow Band IoT, como lo dice su nombre (ancho de banda angosto o bajo), es para aplicaciones donde los dispositivos conectados envían poca información y lo hacen con poca frecuencia. Bien podría ser un sensor que reporte su lectura cada una hora, o dos veces por

día. La tecnología hace especial énfasis en la optimización de la batería. Para llevarlo a cabo se sacrifican algunas de las principales características de las redes celulares como es el handover entre radio bases sin interrumpir el servicio [10]. En el caso de haber un cambio de radio base, el dispositivo es el encargado de volver a conectarse.

Tiene la capacidad de cubrir casos de cobertura extrema, siendo muy bueno en cuanto a penetración en edificios, con buena respuesta en situaciones como, por ejemplo, en subsuelos o grandes edificios.

La velocidad máxima de transferencia de datos, dada por Antel, es de 64kbps y hoy en día está implementado sobre IPv6 (Internet Protocol - versión 6).

LTE-M o LTE Cat-M1

A diferencia de NB-IoT, LTE-M admite mayores velocidades de transferencias de datos, alcanzando velocidades de 1Mbps (tanto de uplink como downlink) [10] y permitiendo la movilidad entre radiobases. Es por esto que puede admitir transmisión de voz sobre la red (VoLTE o Voice over LTE). Antel se encuentra realizando pruebas pilotos con está tecnología pero aún no se está comercializando.

Dada la supuesta gran cobertura en nuestro país de NB-IoT, y particularmente en zonas rurales [11] como se muestra en la Figura 1, se evaluó su uso para este proyecto. Tras una entrevista con el Dr. Ing. Daniel Perciante, profesor de la UCU y co-CEO de Nettra (empresa uruguaya dedicada al desarrollo de productos y soluciones IoT), le comentamos que estábamos evaluando usar NB-IoT para nuestro proyecto. El Dr. Perciante nos argumenta que si queremos escalar la solución a otros países hay que independizarse de la red celular contemplando que no todos los países tienen la cobertura LTE en zonas rurales como tiene Uruguay. Por lo que LoRaWAN es la tecnología más conveniente.

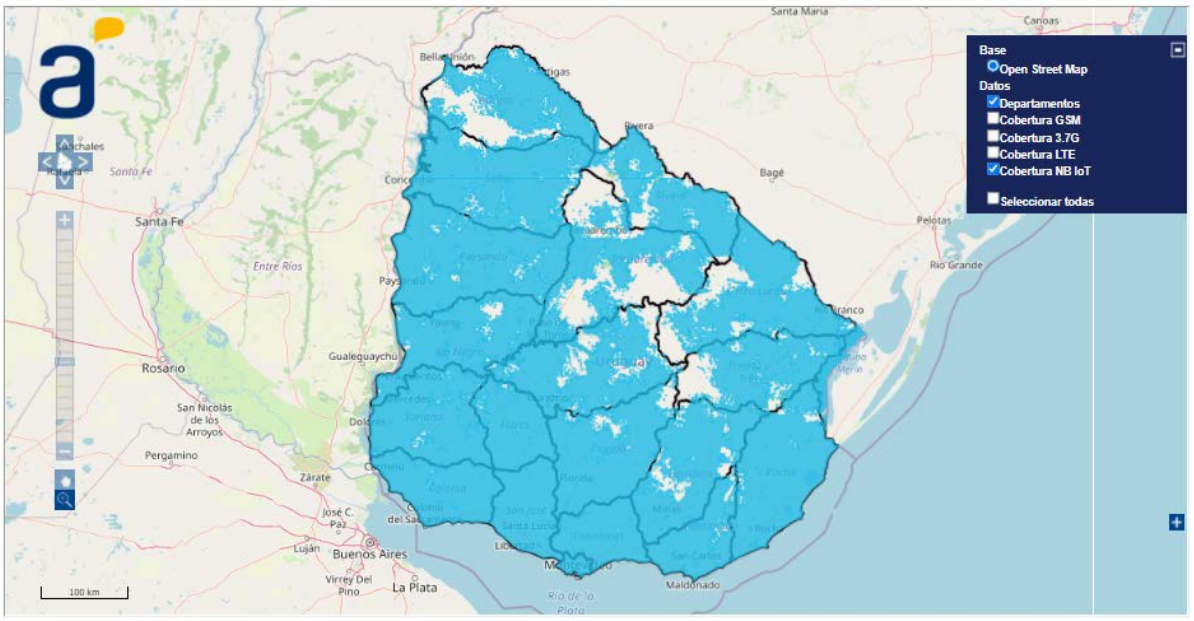

Figura 1 - Cobertura NB-IoT de Antel en Uruguay [11]

2.1.2. Sigfox

Sigfox es un operador global que maneja la red IoT más grande del mundo [12]. Fundado en 2010 en Francia por Ludovic Le Moan y Christophe Fourtet, hoy en día está presente en 75 países, y cuenta con 6 millones de km^2 cubiertos y más de 1200 millones de personas bajo su área de cobertura (Ver Figura 2). Este operador desarrolló su propio protocolo y su propia red, basada en bajo consumo, largo alcance (de hasta 20km) y pequeño tráfico (entre 10 y 1000 bits por segundo), que ofrece un servicio de comunicación end-to-end, es decir, que une los dispositivos finales entre ellos y con el usuario final [13].

La mayor desventaja de Sigfox para los efectos de este proyecto es que implica una suscripción paga, ya que el operador se encarga del despliegue y del mantenimiento de la red con todos los costos que esto implica. Además, es de los contratos con clientes o usuarios que genera su ganancia la empresa.

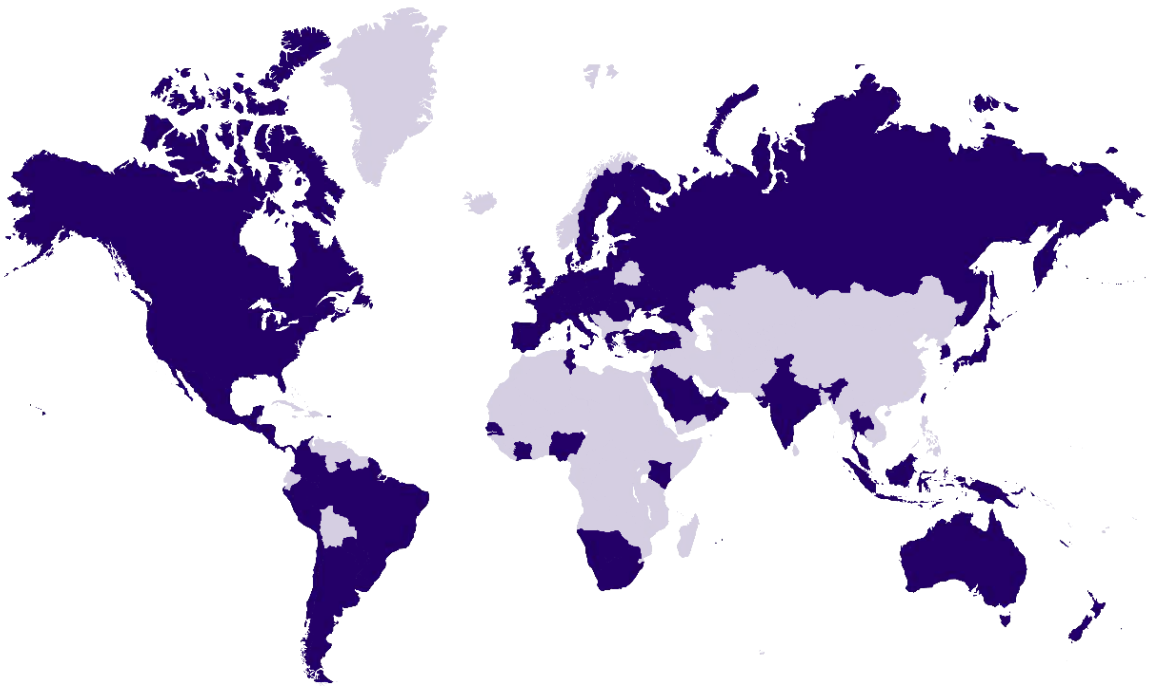

Figura 2 - Cobertura Sigfox por países (azul oscuro) [12]

La estructura de Sigfox (ver Figura 3) se constituye de la siguiente forma:

- En el borde de la red se encuentran los dispositivos finales, los cuales en general son sensores pero pueden ser también (o además) actuadores. Los dispositivos finales son adquiridos por los usuarios finales y se anexan a la red comunicándose con las antenas propias de Sigfox.
- Las antenas son desplegadas propiamente por el operador. Éste se encarga de su instalación y mantenimiento, para asegurar un correcto funcionamiento de la red intercomunicando los dispositivos finales con los servidores internos de Sigfox.
- Los servidores de Sigfox son los últimos componentes propios del operador en la red. También son desplegados y mantenidos por la empresa, asegurando el flujo de información entre los nodos y los servidores del cliente, mediante el uso de APIs.
- Por último, se encuentra el (o los) servidor(es) del cliente, el cual procesa la información recabada por la red, y aloja los recursos para la aplicación de la que harán uso el cliente o los usuarios finales.

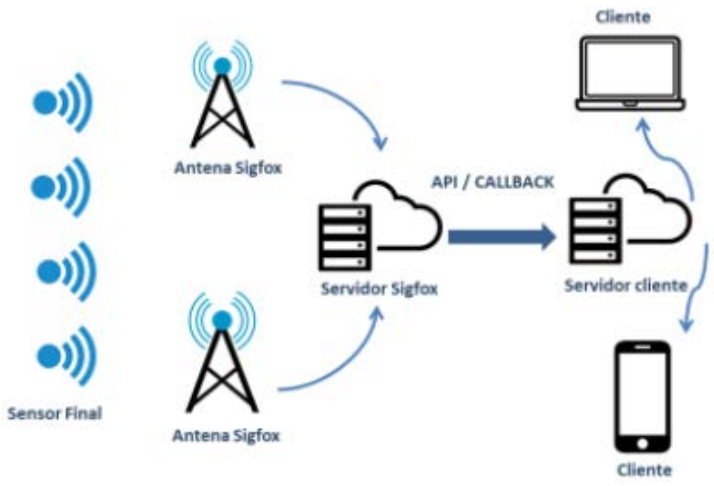

Figura 3 - Infraestructura de Sigfox [13]

En Uruguay, Sigfox tiene una presencia considerable (Ver Figura 4), y muchos usuarios (entre ellos el INIA en algunos proyectos que hoy en día están en fase de prueba) se han inclinado por esta tecnología.

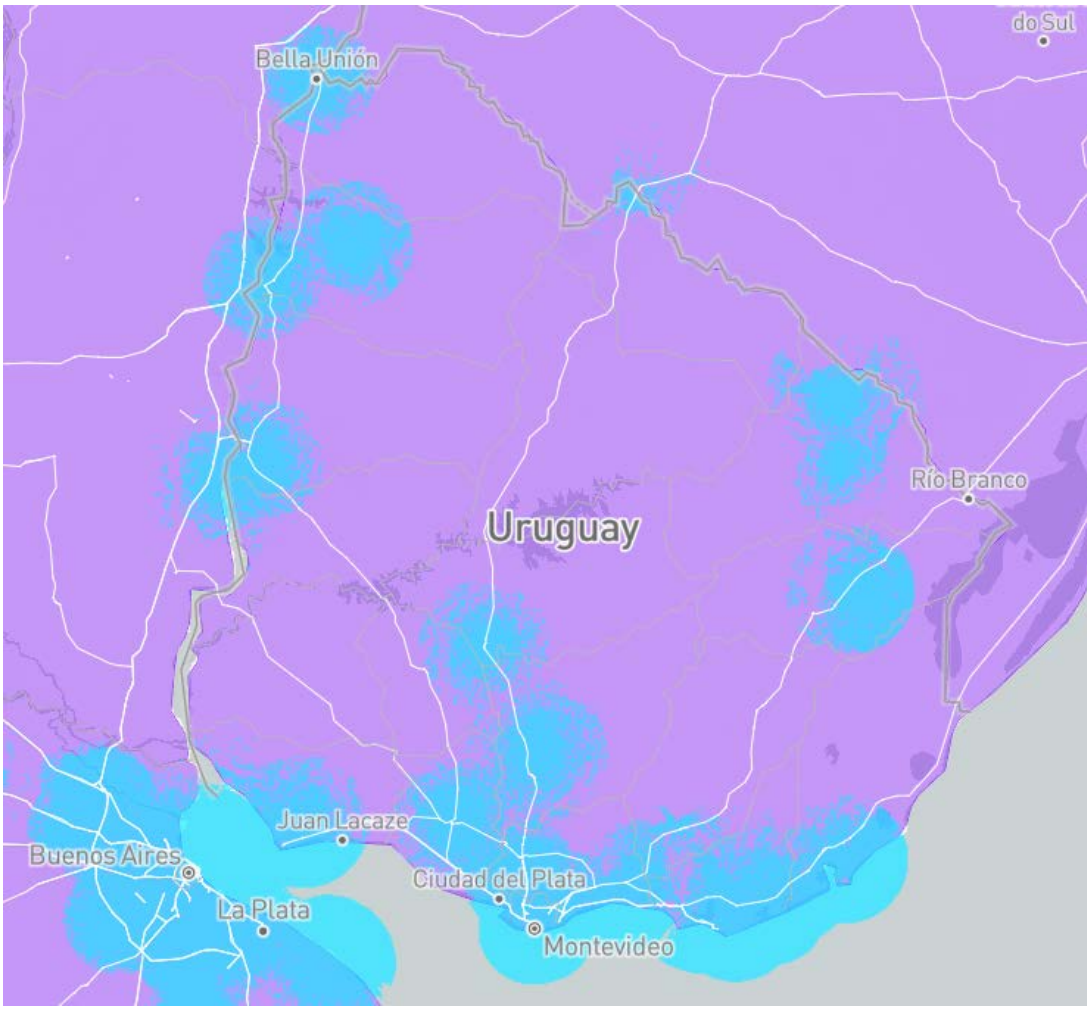

Figura 4 - Cobertura de Sigfox en Uruguay (celeste) [12]

2.1.3. ¿Por qué LoRaWAN?

LoRaWAN es un protocolo de redes IoT en bandas de uso libre que utiliza la tecnología de modulación LoRa. Se caracteriza por su largo alcance y bajo consumo, lo que lo hace ideal para la aplicaciones de tecnología en el agro. Se pasará a detallar este protocolo así como la modulación LoRa, no sin antes explicar por qué fue el designado y en qué entendemos que es superior, para la aplicación de este proyecto, a los otros protocolos mencionados.

Las características de LoRaWAN que fueron decisivas para que éste sea el protocolo de comunicación elegido son las siguientes:

- Independencia de la red celular: Por más que en Uruguay la cobertura celular es muy buena (como fuera explicado anteriormente), es una gran ventaja no depender de ella. De esta forma el nodo puede ser instalado en lugares donde la cobertura no llega, aumentando su usabilidad y permitiendo una escalabilidad a otros sitios donde la red celular no está tan desarrollada. Esta característica inclina la balanza hacia LoRaWAN por sobre Cellular IoT.
- Tecnología de uso libre: LoRaWAN es un protocolo que no necesita licencias ni suscripciones para poder ser utilizado. Es de uso libre y gratuito, pero implica el armado de la red por parte del usuario, lo que para los intereses de este proyecto no era una desventaja sino un atractivo, ya que infiere en un aprendizaje muy valioso. En este apartado fue que se inclinó la balanza hacia LoRaWAN por sobre Sigfox, donde el operador es el que arma la red, la cuál además comprende un costo por utilización.

2.2. LoRa y LoRaWAN

Para comprender qué son LoRa (de Long Range) y LoRaWAN (de Long Range Wide Area Network) y cómo se diferencian, es necesario entender que trabajan en capas diferentes del modelo OSI (Open Systems Interconnection), como se puede apreciar en la Figura 5. LoRa es una técnica de modulación por lo tanto trabaja en la capa física, mientras que LoRaWAN es un protocolo de capa MAC (Media Access Control) construido sobre LoRa.

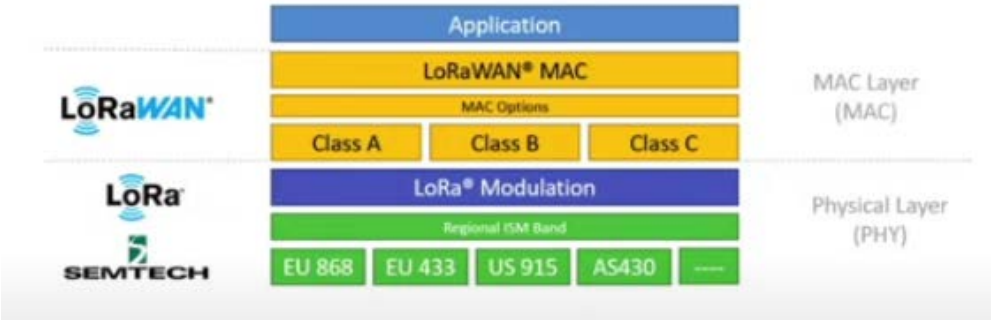

Figura 5 - Modelo de capas para LoRa y LoRaWAN [17]

2.2.1. LoRa

Es una técnica de modulación patentada por Semtech [14], una empresa estadounidense fabricante de chips de radio. Está basada en la tecnología Chirp Spread Spectrum (CSS), una tecnología de modulación que se utiliza en comunicaciones militares y espaciales desde hace décadas [15].

LoRa está administrada por LoRa Alliance [16], una asociación que se encarga de la estandarización de la tecnología, y que es quien certifica a los fabricantes de hardware que deseen trabajar con LoRa. Esta tecnología es ideal para aplicaciones de baja transferencia de datos y grandes distancias (llegando hasta los 20 km de alcance), es por eso que está ampliamente extendida en el mundo IoT. Otras características de LoRa son el bajo consumo energético que implica para los dispositivos finales, la gran sensibilidad de recepción (-168dB), y la alta tolerancia a las interferencias [17]. En la Figura 6 se puede apreciar la gran ventaja de LoRa frente a otras tecnologías de comunicación inalámbrica, en su relación de distancia de cobertura vs. ancho de banda. LoRa es la tecnología ideal para aplicaciones en las que los nodos necesitan independencia de la red eléctrica (por sus bajos consumos las baterías pueden durar muchos años) y de la red celular (por sus largas distancias de cobertura simplemente teniendo el gateway conectado a internet se pueden armar redes de 20 km de diámetro a su alrededor).

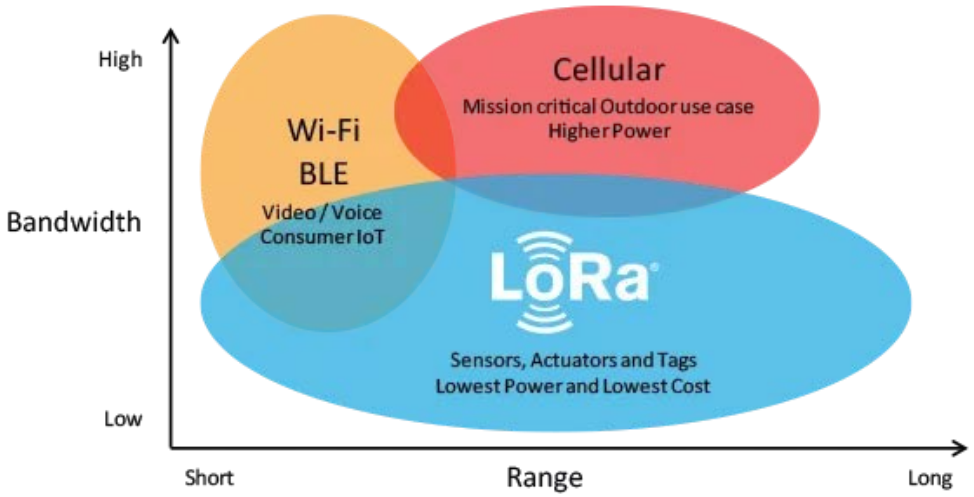

Figura 6 - Distancia de cobertura vs Ancho de banda [17]

2.2.1.1. Parámetros Regionales de Frecuencia

LoRaWAN utiliza frecuencias de uso libre, por lo tanto no es necesario tener licencia sobre espectro para poder aplicar esta tecnología. Es similar a lo que sucede con WiFi, que utiliza 2.4GHz y/o 5GHz libremente en todo el mundo. Sin embargo, los parámetros de frecuencia para LoRaWAN no son fijos para todo el mundo, sino que dependen de la región. En la página de LoRa Alliance está la documentación correspondiente sobre las bandas de frecuencia para cada región [18].

Plan de Canales	Nombre común	Identificador
EU863-870	EU868	1
US902-928	US915	2
CN779-787	CN779	3
EU433	EU433	4
AU915-928	AU915	5
CN470-510	CN470	6
AS923-18	AS923	7
AS923-2	AS923-2	8
AS923-3	AS923-3	9
KR920-923	KR920	10
IN865-867	IN865	11
RU864-870	RU864	12
AS923-4	AS923-4	13

Tabla 1 - Planes de Canales [18]

En la Tabla 1 se muestran los distintos planes de canales, junto a su nombre por el cual se conocen. A grandes rasgos las primeras dos letras de los nombres indican las regiones implicadas. EU para Europa, US para Estados Unidos, CN para China, AU para Australia, AS para Asia, KR para Corea, IN para India y RU para Rusia. Los planes de canales se pueden separar en dos grandes tipos:

- Los planes de canales dinámicos: En los que la mayoría de los canales son definidos luego del proceso de activación. En el Anexo A se encuentran detalladas sus características.
- Los planes de canales fijos: Donde todos los canales son estáticos, definidos previamente al proceso de activación (a partir de la versión LoRaWAN 1.1.1 ya no son todos sino la mayoría, pero en este proyecto sí son todos ya que se utiliza la versión LoRaWAN 1.0.2). En el Anexo B se encuentran detalladas sus características

En el caso de Uruguay se utiliza el plan de canales AU915, del tipo de los planes de canales fijos. La regulación implica que la banda es 902 - 928 MHz, pero a efectos reales solamente está disponible a partir de 915 MHz, por lo tanto la banda utilizada es 915 - 928 MHz.

2.2.1.2. AU915

Este plan de canales define los parámetros para Australia y otras regiones del mundo donde la regulación implica la utilización de la banda 902 - 928 MHz para LoRaWAN. En este apartado entran no sólo Uruguay, sino también la mayoría de los países sudamericanos. Es un plan de canales estáticos que se definen de la siguiente manera:

- Upstream - 64 canales de subida de 125 kHz de ancho de banda, numerados del 0 al 63 (verdes en la Figura 7), que utilizan tecnología de modulación LoRa. El canal 0 comienza en 912,2 MHz y luego se concatenan uno por uno hasta el 63, separados cada 200 kHz y llegando hasta 927,8 MHz.
- Upstream - 8 canales de subida de 500 kHz de ancho de banda, numerados del 64 al 71 (celestes en la Figura 7), que utilizan tecnología de modulación LoRa. El canal 64 comienza en 915,9 MHz y luego se concatenan uno por uno hasta el 71, separados cada 1,6 MHz y llegando hasta 927,1 MHz.
- Downstream - 8 canales de bajada de 500 kHz de ancho de banda, numerados del 0 al 7 (naranjas en la Figura 7), que utilizan tecnología de modulación LoRa. El canal 0 comienza en 923,3 MHz y luego se concatenan uno por uno hasta el 7, separados cada 600 kHz y llegando hasta 927,5 MHz.

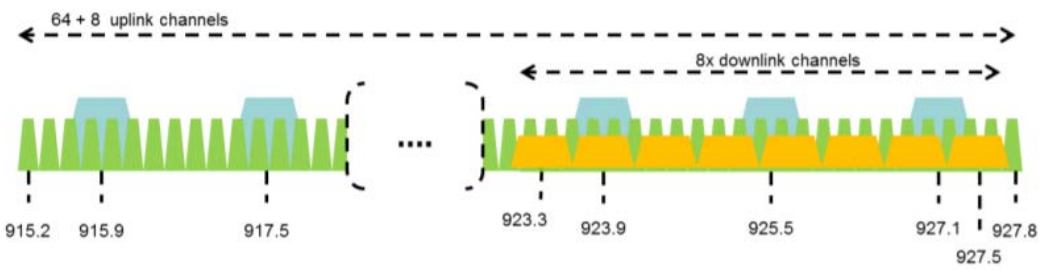

Figura 7 - Distribución de canales en el espectro para AU915 [18]

2.2.2. LoRaWAN

LoRaWAN [17] es un protocolo de Control de Acceso al Medio (MAC), es decir que se encuentra dentro de la capa de Enlace en el modelo OSI (Ver Figura 8), por encima y utilizando los recursos de la modulación LoRa que se encuentra en la capa física. LoRaWAN define en una capa de software cómo utilizar el hardware de LoRa, el formato de los mensajes, cuándo y cómo transmitir, etc.

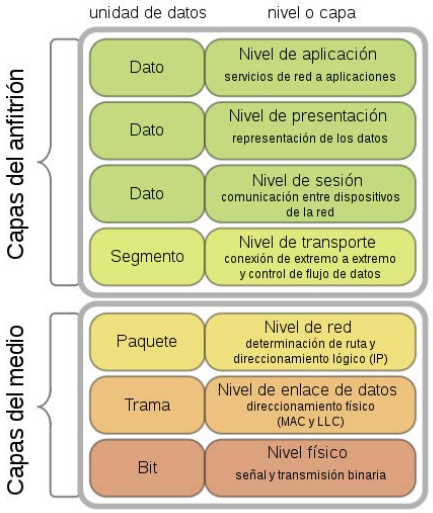

Figura 8 - Modelo OSI [19]

2.2.2.1. Ventajas de LoRaWAN

La gran mayoría de las ventajas que presenta LoRaWAN coinciden con las ya expuestas para LoRa, a continuación se detallan una por una:

- **Bajo consumo:** Los dispositivos finales que utilizan esta tecnología son optimizados para ir en línea con la baja potencia de comunicación que implica el protocolo, por lo que, dependiendo del sistema, pueden llegar a durar hasta 10 años con una sola pila de botón (la utilizada en los relojes).
- **Largo alcance:** Los gateways LoRaWAN pueden enviar y recibir mensajes a más de 10km en zonas rurales y hasta 3km en zonas urbanas de alta densidad.
- **Penetración profunda en interiores:** Las redes LoRaWAN pueden dar cobertura a edificios de muchos pisos traspasando paredes, pisos y techos.

- **Espectro sin licencia:** LoRaWAN trabaja en frecuencias de uso libre, por lo que no es necesario comprar licencias de espectro para poder utilizarla.
- **Alta tolerancia a las interferencias:** Gracias a que está basado en modulación Chirp.
- **Geolocalización:** En redes de complejidad moderada, se puede ubicar los nodos por medio de triangulación si al menos 3 gateways reciben su señal.
- **Alta capacidad:** Los servidores de red de LoRaWAN pueden procesar millones de mensajes provenientes de miles de gateways.
- **Seguridad:** Conexiones bidireccionales encriptadas de extremo a extremo.
- **Interoperabilidad:** Al ser una tecnología de uso libre y no un proveedor de redes IoT (como lo es Sigfox por ejemplo) es fácil interoperar o desarrollar redes LoRaWAN públicas o privadas utilizando el mismo hardware y el mismo software para gateways, dispositivos finales y servidores.
- **Roaming:** Los dispositivos finales tienen la capacidad de realizar traspasos de conexión de una red a otra manteniendo la aplicación.
- **Actualizaciones OTA (Over The Air):** Es posible enviar actualizaciones de firmware a uno o a un determinado grupo de dispositivos finales.
- **Bajo costo:** Debido a los dispositivos finales económicos, el software de código abierto y la poca infraestructura requerida para el armado de las redes.
- **Certificaciones y Ecosistema:** Al ser una tecnología abierta y estandarizada, hay una gran oferta de fabricantes de dispositivos finales, gateways y antenas, así como de proveedores de servicios de red y de aplicaciones. Además, todos estos elementos son certificados por LoRa Alliance, por lo que no hay que preocuparse por la compatibilidad con las especificaciones del protocolo.

2.2.2.2. LoRa Alliance

Es una asociación sin fines de lucro establecida en 2015 [16] con el objetivo de asegurar la interoperabilidad de los dispositivos y las tecnologías LoRaWAN, y continuar con el desarrollo del protocolo. Es abierta y cuenta con más de 500 miembros alrededor del mundo. Es la encargada de brindar las certificaciones para los fabricantes de dispositivos

finales y gateways, asegurando la compatibilidad de los dispositivos con las especificaciones de LoRaWAN. Estas certificaciones están disponibles solamente para los miembros de la asociación, los cuales una vez certificados pueden utilizar la marca LoRaWAN en sus productos.

2.2.2.3. Arquitectura

Las redes LoRaWAN generalmente se organizan en una topología de estrella en la que los gateways transmiten mensajes entre dispositivos finales (nodos) y el servidor de red en el backend, que a su vez se comunica con el servidor de aplicación. Los gateways están conectados al servidor de red a través de enlaces IP mientras que los nodos usan LoRaWAN. La comunicación es generalmente bidireccional. Las redes LoRaWAN utilizan el principio del protocolo ALOHA [20], es decir que no hay "pairing" entre los dispositivos finales y los gateways. No es una comunicación "peer to peer" sino que todos pueden hablar entre todos mientras estén dentro del rango de cobertura. En la Figura 9 se puede apreciar un esquema genérico de una red LoRaWAN, cuyos elementos se detallan a continuación.

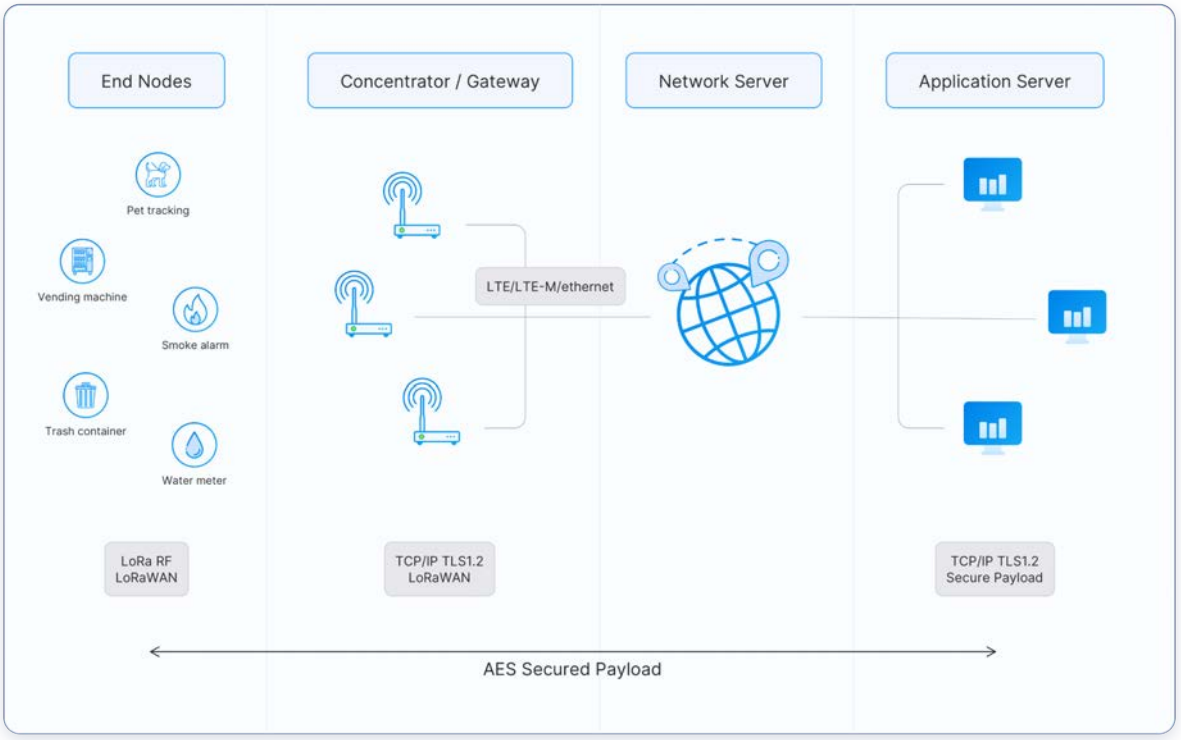

Figura 9 - Arquitectura LoRaWAN [17]

End Nodes: Los dispositivos finales, también llamados nodos, son objetos con un módulo de comunicación LoRaWAN integrado. Pueden involucrar sensores, actuadores, o ambos, y en general son alimentados mediante baterías.

Gateway: También llamado concentrador, es el nexo entre los dispositivos finales y los servidores. Es el elemento que conoce los dos mundos y habla los dos idiomas. Se comunica con los dispositivos finales mediante LoRaWAN y con los servidores mediante comunicación IP. Se pueden separar en dos categorías: indoor (picocell) y outdoor (macrocell). Como sugieren sus nombres, outdoor se utiliza para exteriores, ya sea en zonas rurales o urbanas, mientras que indoor es más adecuado para cubrir espacios interiores, o para mejorar su cobertura. La huella de los outdoor es macrocell, significativamente más grande que la de los indoor que es picocell.

Network Server: El servidor de red es el que maneja y administra la red. Es el nexo entre los gateways y el servidor de aplicación. Cuando le llega un mensaje, decodifica el encabezado y lo redirige al servidor de aplicación correspondiente.

Application Server: El servidor de aplicación es el otro extremo de la red. Recibe los mensajes del servidor de red y los decodifica mediante para obtener el payload. Es el encargado de asignar los identificadores y las llaves de sesión a los dispositivos finales al usar OTAA.

2.2.3. OTAA vs ABP

Existen dos modos de activación para los dispositivos finales, uno es OTAA (Over-The-Air Activation) y el otro es ABP (Activation By Personalization) [21]. El proceso de activación no solamente tiene que ver con la asignación de un identificador en la red para los nodos (DevAddr), sino que también infiere en la seguridad de los datos y de la red en general. Al estar hablando de un protocolo de comunicación inalámbrica, la seguridad cobra un valor importante, ya que los paquetes pueden ser recibidos por cualquier dispositivo compatible con el protocolo que se encuentre dentro del área de cobertura, la cual en el caso de LoRaWAN es muy extensa [22]. Como una primera aproximación a estos dos métodos se puede asociar OTAA con DHCP (Dynamic Host Configuration Protocol) para IP, una

configuración dinámica de ID para los dispositivos finales, en la cual el nodo envía un mensaje de *join-request* para unirse a la red y, si es aceptado, el servidor de red le devuelve un mensaje de *join-accept* de donde obtiene su nueva DevAddr y sus llaves de sesión. Cabe destacar que este proceso se vuelve a realizar cada vez que el nodo inicia una nueva sesión. Por otro lado, en esta analogía, ABP se asociaría a una configuración manual de IP, la DevAddr y las llaves de sesión se setean en el firmware de los nodos previo a su instalación.

Antes de entrar en los detalles de cada uno y en qué se basó la elección del modo en este proyecto, es conveniente precisar un glosario con los términos a utilizar, para comprender de manera más directa el funcionamiento de ambos métodos [23][24].

- **EUI** (Extended Unique Identifier): refiere a un identificador único global.
- **DevEUI** (Device EUI): ID global único de 64 bits (8 bytes) para cada dispositivo, seteado por el fabricante, se puede asociar con la MAC Address de las tarjetas o dispositivos de red.
- **AppEUI** (Application EUI): Es un identificador de la aplicación final.
- **App Key** (Application Key): Es una llave de cifrado en formato AES-128 bit (AES refiere a Advanced Encription Standard)
- **DevAddr** (Device Address): Es un identificador del dispositivo en su red particular, de 32 bits, se puede asociar con la IP local de un dispositivo en una LAN.
- **NwkSKey** (Network Session Key): Es la llave que encripta la metadata de los paquetes desde los dispositivos finales hasta el Servidor de Red.
- **AppSKey** (Application Session Key): Es la llave que encripta el payload de los paquetes desde los dispositivos finales hasta el Servidor de Aplicación.
- **NetID** (Network Identifier): Es un identificador único para la red.
- **Nonce**: Es un número que se usa una única vez en los mensajes criptográficos.
- **DevNonce:** Es un Nonce de 2 bytes, único y aleatorio, que se utiliza en OTAA. Es generado y enviado desde el dispositivo final al servidor de aplicación.

- **AppNonce**: Es un Nonce que se utiliza en OTAA, es generado y enviado desde el servidor de aplicación a los dispositivos finales.
- **Network Server**: Es el servidor de la red, su tarea es concentrar la información recibida por los gateways, descartar mensajes duplicados, y decodificar el encabezado de los paquetes para redirigirlos al Application Server correspondiente.
- **Application Server**: Es el servidor de la aplicación, decodifica el payload de los paquetes y realiza las funciones específicas de la aplicación.
- **MIC** (Message Integrity Code): Es similar a un checksum, que además evita la manipulación maliciosa de los mensajes mediante el algoritmo de cifrado AES-CMAC.

2.2.3.1. OTAA

Este proceso comienza en el borde de la red, el primer paso es previo a la instalación y es la asignación a los dispositivos finales de una AppKey y una AppEUI en el firmware. Los nodos son los que inician la interacción con el Application Server mediante el mensaje de *join-request* (Ver Figura 10).

8 bytes	8 bytes	2 bytes	4 bytes
AppEUI	DevEUI	DevNonce	MIC

Figura 10 - Estructura del join-request

El mensaje de *join-request* no está cifrado, y puede ser visible por cualquiera que lo reciba. Éste contiene el DevEUI único de la placa LoRaWAN del dispositivo, la AppEUI presente en el firmware del mismo, un DevNonce y un MIC. El DevNonce es utilizado por el Network Server para hacer un seguimiento de todos los dispositivos finales que se registran en la red. Si un nodo intenta unirse a la red mediante un *join-request* cuyo DevNonce ya fue utilizado, el Network Server rechazará la solicitud. Esto sirve para evitar los ataques de replay (estos ataques consisten en capturar mensajes y retransmitirlos numerosas veces con el objetivo de saturar la red) [25]. Por último, el MIC se genera a partir de los otros tres campos del mensaje y de la AppKey presente en el firmware del nodo. Se utiliza para asegurar la

integridad del mensaje y para que el Application Server pueda corroborar que la Appkey es la correcta. Es importante que la misma Appkey sea configurada en la red para que el MIC coincida en ambos extremos y el nodo sea aceptado para ingresar a la misma. En esta instancia, el dispositivo es agnóstico de hacia dónde está enviando el mensaje, no conoce ningún gateway ni el Application Server o Network Server. Cualquier gateway de la red que reciba este mensaje lo reconoce por la AppEUI, y lo redirige al Application Server por medio del Network Server.

A partir de este momento el recorrido es el inverso. El Application Server recibe el *join-request* y verifica primero que el DevNonce no haya sido utilizado previamente por un nodo y luego, mediante el MIC y la AppKey, la integridad del mensaje, lo que infiere que el dispositivo final contiene la AppKey correcta y por lo tanto está habilitado para unirse a la red. Dado todo esto, procede a enviar el *join-accept*. Si por el contrario alguna de estas premisas no se verifica, simplemente descarta el mensaje. El mensaje de *join-accept* está compuesto como indica la Figura 11.

3 bytes	3 bytes	4 bytes	1 bytes	1 bytes	16 bytes	4 bytes
AppNonce	NetID	DevAddr	DLSettings	RXDelay	CFList	MIC

Figura 11 - Estructura del join-accept [17]

El AppNonce es un valor aleatorio y único, a partir del cuál el dispositivo final genera las llaves de sesión (AppSKey y NetSKey). El NetID es el identificador de la red, mientras que el DevAddr será el identificador, único en la red, del nodo al cuál se le envía el *join-accept*. DLSettings y RXDelay son parámetros de comunicación para el dispositivo final, el primero indica la configuración de downlink que deberá utilizar, y el segundo indica el delay que debe utilizar entre RX (Recepción de datos) y TX (Envío de datos). CFList contiene la lista de frecuencias que puede utilizar el nodo. Por último, se vuelve a agregar un MIC para verificar la integridad del mensaje. A diferencia del *join-request*, el *join-accept* sí está cifrado. Se encripta con la AppKey, que a esta altura del proceso ya se verificó ser idéntico en el

dispositivo final y en el Application Server. De este modo el nodo puede desencriptar el mensaje y almacenar su DevAddr junto con las configuraciones de comunicación, y generar el AppSKey y el NetSKey. Tanto las llaves de sesión como el DevAddr son únicos para el nodo en la sesión actual. Si el dispositivo final fuera a establecer una nueva sesión se repite todo el proceso y se generan unas nuevas llaves de sesión y un nuevo DevAddr. En la Figura 12 se puede apreciar un esquema del proceso OTAA.

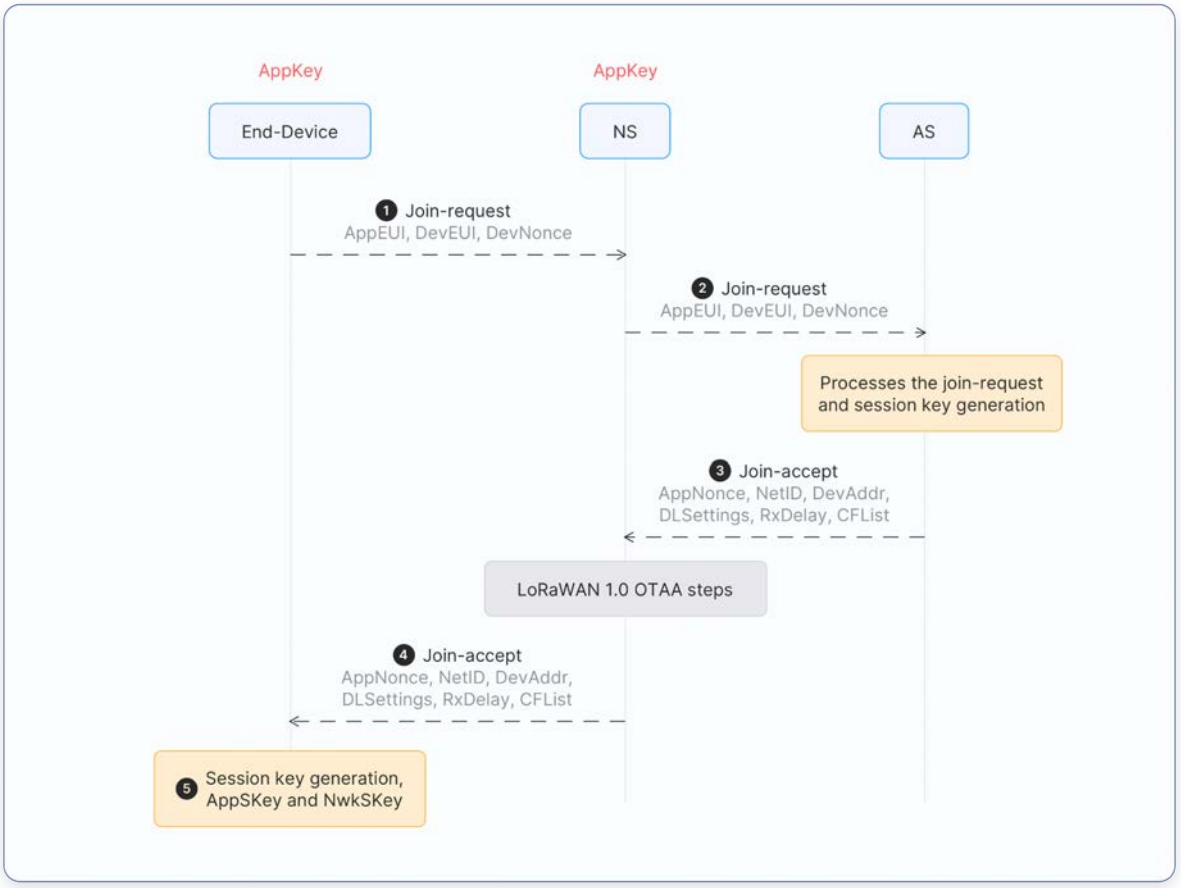

Figura 12 - Proceso OTAA [23]

2.2.3.2. ABP

Este método de activación es mucho más sencillo que el OTAA, por lo que no es casualidad que sea menos robusto en cuanto a la seguridad. El proceso se realiza previo a la instalación del nodo, y consiste en el seteo manual de DevAddr, NetSKey y AppSKey, las cuales van "harcodeadadas" en el firmware. A su vez, se debe setear las mismas en los servidores para

que la comunicación sea posible. En el Network Server se almacenan el DevAddr y la NetSKey, mientras que en el Application Server se almacena la AppSKey (Ver Figura 13).

Esto implica que la red debe ser pre-seleccionada, quedando vinculado el dispositivo a esa red definitivamente con las mismas llaves de sesión, hasta que se reprograme manualmente.

Figura 13- ABP - datos en cada elemento [23]

2.2.3.3. ¿Qué método es conveniente utilizar?

En primera instancia dependerá del caso, sin embargo en general lo más recomendable es utilizar OTAA. Hay dos factores fundamentales que inclinan la balanza hacia este método: flexibilidad y seguridad.

En cuanto a la flexibilidad, al utilizar ABP el nodo queda con una DevAddr fija, y por lo tanto también queda fijo a la red que tenga seteada, y es en la única que puede funcionar excepto que se modifique el firmware. Con OTAA, el nodo se encuentra fijo a la aplicación, pero es independiente de la red. Si se cambia de red no es necesario modificar el dispositivo final, simplemente vuelve a cursar el proceso de activación, se inicia una nueva sesión y se obtienen las nuevas DevAddr y llaves de sesión. Por otro lado, en ABP se deben fijar los parámetros de comunicación, los cuales también quedarán fijos, pudiendo generar problemas si están mal configurados. En cambio, en OTAA estos parámetros se establecen de forma correcta en cada inicio de sesión.

La falta de flexibilidad vuelve a ser desventaja para ABP en el apartado de la Seguridad. Al estar las llaves de sesión “hardcodeadas” en el firmware, en el caso en que se vean vulneradas las mismas, no hay posibilidad de cambiarlas sin reprogramar el nodo manualmente. En cambio, al utilizar OTAA las llaves de sesión son renegociadas cada vez que se establece una nueva sesión. Algo similar pasa con los frame counters.

Los frame counters (*FCntUp* y *FCntDown*) [22] son dos contadores cuyo objetivo es bloquear los ataques de replay. Cuando un dispositivo final es activado ambos son seteados en cero, luego cada vez que el nodo envía un mensaje el *FCntUP* es incrementado, y cada vez que recibe un mensaje de la red el *FCntDown* es incrementado. Si el dispositivo final o la red recibe un mensaje con un frame counter menor que el del último recibido, descarta el mensaje. Si se utiliza ABP hay dos posibles inconvenientes con los frame counters. El primero es que si el nodo se reinicia, ya sea por un cambio de batería o por un cambio en el firmware, los frame counters vuelven a cero, y al tener la red guardado el último *FCntUP*, todos los mensajes enviados por el nodo serán descartados hasta que el *FCntUP* alcance el que estaba almacenado en la red. Esto se soluciona volviendo a registrar el dispositivo final en el Application Server cada vez que se resetea, para que vuelvan todos los frame counters a cero. El otro inconveniente es que los nodos se queden sin frame counters, es decir, que el contador llegue a su límite de memoria, y en este caso todos sus siguientes mensajes serán descartados. Ambos problemas se solucionan sencillamente en OTAA ya que en cada nuevo establecimiento de sesión se resetean a cero los frame counters de ese nodo tanto del lado de la red como del lado del nodo.

La ventaja más grande que presenta ABP sobre OTAA es su sencillez, por lo que para prototipos, redes pequeñas y entornos controlados, puede ser conveniente inclinarse por ABP. Además, el procedimiento de activación de OTAA requiere una capacidad de recursos, por parte del nodo, mayor que la activación por ABP redundando en mayor consumo energético.

En este proyecto se eligió utilizar ABP por ser un entorno controlado, una red de un solo nodo y un solo gateway, en un sistema sin riesgo de ataque. Luego de probar ambos métodos, en estado de prototipo, se verificó una mejor performance de ABP.

Capítulo 3 - Elección de los componentes de la solución

3.1. Elección de la placa de desarrollo

Se elige la placa de desarrollo de bajo consumo Arduino MKR WAN 1310 [26] con comunicación LoRa por ser la que más se adecúa a las necesidades y alcance del proyecto (Ver Figura 14). Además por la gran cantidad de documentación, tutoriales, librerías y casos de uso que ofrece Arduino y su versatilidad en cuanto integración de sensores, haciéndola una solución muy amigable y de relativo fácil desarrollo. En este modelo se resuelven algunas ineficiencias de consumo energético de su antecesora MKR WAN 1300.

Esta placa se basa en el procesador de bajo consumo Microchip SAMD21 y el módulo de conectividad LoRa Murata CMWX1ZZABZ, pudiendo trabajar en la frecuencia de 915MHz. Permite utilizar tanto ABP y OTA para la activación LoRa y posee un chip de cifrado ECC508 lo que agrega más seguridad al almacenar credenciales y certificados.

Se puede alimentar con 5V via USB o bien con una batería de 3.7V. El consumo de la placa alimentada por USB es muy superior al alimentarse a través de batería [26]. Se reduce aún más el consumo si se alimentan directamente con 3.3V el microprocesador y el módulo de

comunicación. En la página del fabricante se dice que se pueden alcanzar consumos tan bajos como 104µA.

Cuenta con entradas digitales y analógicas, con un conversor A/D de 10 bits, lo que brinda versatilidad en la integración de sensores y una resolución aceptable.

La placa viene con una antena de dipolo simple para la trasmisión y recepción de datos, la misma consiste en dos conductores rectilíneos colineales de igual longitud.

Arduino cuenta con un IDE (Integrated Development Environment) simplificando la escritura del código y la subida del mismo a la placa a través del puerto USB. Cuenta con herramientas de “debuggeo” que facilitan a la hora del desarrollo de las partes del código. Se hablará del mismo más adelante en el documento.

En el Anexo C se muestran las especificaciones técnicas completas.

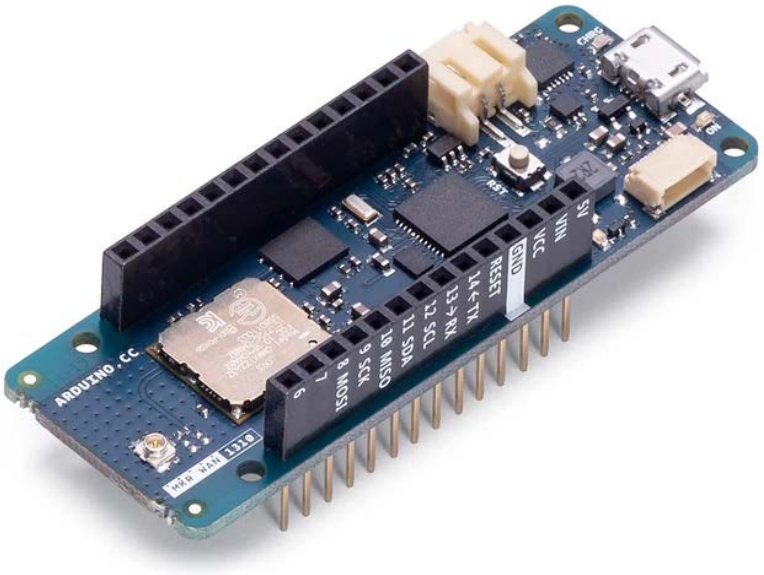

Figura 14 - Arduino MKR WAN 1310 [26]

3.2. Elección del gateway

Se selecciona el gateway de interior LoRaWAN Dragino LPS8 915MHz (Ver Figura 15) [27] el cual permite el pasaje de LoRa a IP a través de WiFi (2.4G WiFi - 802.11 bgn) o Ethernet (puerto RJ45 10M/100M). Es de procedencia china y de bajo costo. Éste utiliza un módulo de

manejo de paquetes Semtech y un concentrador LoRa SX1308 que es totalmente compatible con el protocolo LoRaWAN.

Conectándose al access point WiFi que emite el gateway, y a una determinada IP, se accede al servidor web con la interfaz de configuración del mismo, donde se configuran las bandas de frecuencia a utilizar, se conecta con el servidor LoRaWAN The Things Network (explicado en el capítulo 5) y se pueden visualizar los logs y paquetes recibidos y enviados, facilitando las pruebas de conexión con la Arduino. En el Anexo D se muestran sus especificaciones.

Figura 15 - Gateway LoRaWAN Dragino LPS8

3.3. Investigación y elección de sensores

Para la aplicación para la cual se pretende utilizar esta solución, el dato de humedad de suelo es el más importante. Sin embargo, los cuatro datos (temperatura y humedad en el aire, y temperatura y humedad en el suelo) aportan para la toma de decisiones de riego.

Para la elección de estos sensores se tuvo en cuenta su costo, que tengan protección para exterior, al menos IP65 (hermético al polvo y protegido contra el agua proyectada, de

manera de poder permanecer a la intemperie), así como el voltaje de alimentación sea los que entrega la Arduino (3.3 V o 5 V) para evitar una batería adicional que lo alimente.

3.3.1 Humedad del suelo

Primero es pertinente definir qué se entiende por humedad del suelo. La humedad en el suelo generalmente se refiere al contenido volumétrico de agua, que señala la cantidad o porcentaje de agua en el suelo (Ver Figura 16) [28].

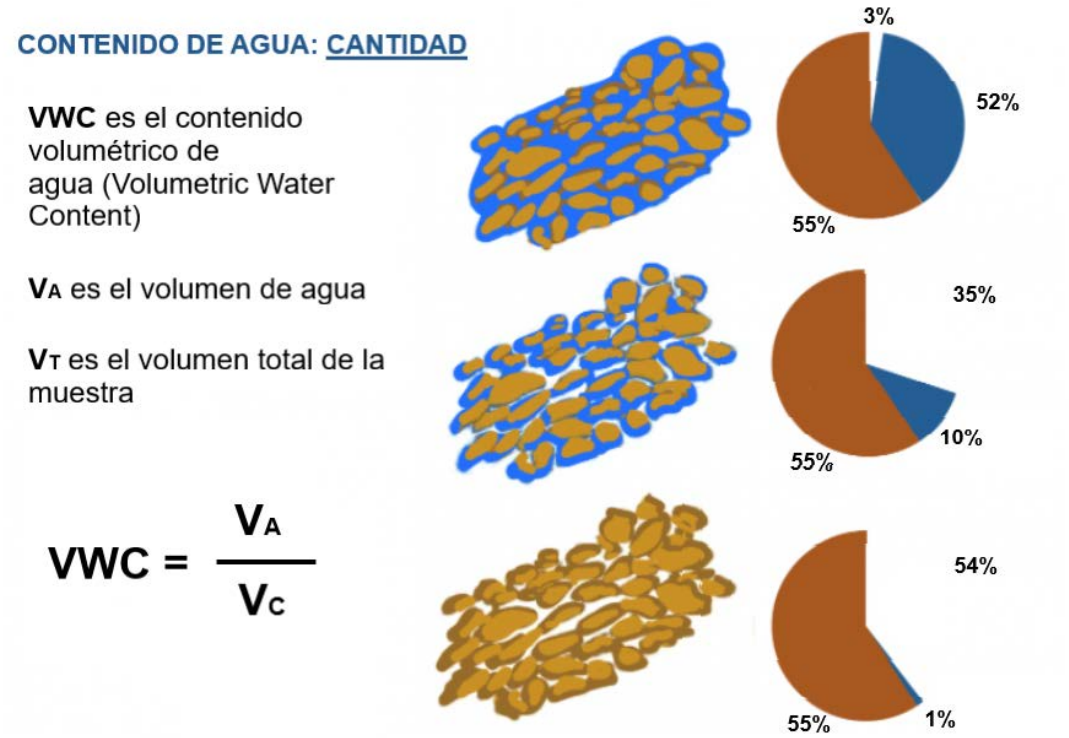

Figura 16 - Contenido volumétrico de agua en el suelo [28]

[Imagen traducida al español. Las gráficas a la derecha muestran los porcentajes de: Minerales del suelo (marrón), Aire (blanco), y Agua (azul)]

El suelo (composición, estructura, caracterización y propiedades) en sí, es algo muy complejo y su estudio excede al alcance de este proyecto. Sin embargo, se ha investigado al respecto y asistido a webinars sobre el tema, como por ejemplo: “El idioma del suelo” de Kilimo [29], donde se explicaron algunos aspectos básicos de esta temática y su relación

directa con la oferta de agua, ya sea en riego, precipitaciones y napas subterráneas así como las pérdidas por evapotranspiración y escurrimiento.

Podemos decir que el suelo está compuesto de 4 grandes ingredientes: materia orgánica, minerales, agua y aire. Las proporciones entre estos ingredientes y tipo de minerales (limo, arcilla, arena) definen al suelo que es único en cada ecosistema y lugar geográfico, así como su capacidad de retención del agua.

Vamos a observar a continuación que una de las técnicas para medir la humedad en el suelo tiene que ver directamente con la utilización de un sensor dieléctrico: cada uno de los ingredientes del suelo tiene una constante dieléctrica diferente y, generalmente, todos tienen constantes dieléctricas bajas en comparación con el agua. Por lo tanto, cuando la capacidad de almacenamiento de carga eléctrica del suelo se mide con un sensor dieléctrico, el agua y el aire son los únicos elementos que cambian significativamente por volumen, por lo que se los puede relacionar directamente con el contenido volumétrico de agua [28].

También en él conviven todo tipo de organismos, vegetales, animales y hongos. Es común que en aplicaciones de sensores en el suelo, animales como algunos roedores, desentierren los cables e incluso los corten mordisqueandolos.

3.3.2. Sensores resistivos, capacitivos, FDR (Frequency Domain Reflectometry) y TDR (Time Domain Reflectometry)

Resistivos: Se mide la variación de la resistencia eléctrica del suelo. En general están compuestos de dos sondas recubiertas de un metal conductor (electrodos) y miden la resistencia que ofrece el suelo (Ver Figura 17). Como nos dice la intuición, a mayor cantidad de agua presente en el suelo, mayor es la cantidad de sales disueltas y menor resistencia presenta. En un suelo seco la resistencia es mucho mayor.

Figura 17 - Ejemplo sensor de humedad de suelo tipo resistivo

Capacitivos: Se mide la variación de la capacitancia del suelo. El dieléctrico entre los conductores es el suelo (Ver Figura 18) y la permitividad eléctrica cambia con el cambio de humedad [28], por lo tanto, se calibra la relación entre la capacidad de almacenamiento de carga eléctrica del suelo y el contenido volumétrico. A mayor humedad mayor capacidad.

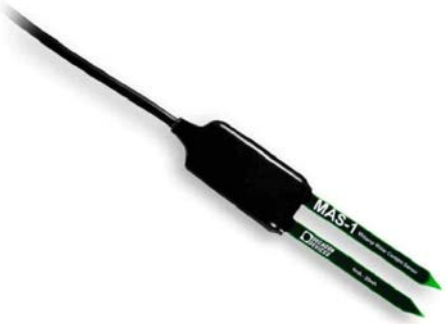

Figura 18 - Ejemplo sensor de humedad de suelo tipo capacitivo

FDR: también utiliza el suelo como un condensador para medir la frecuencia de resonancia máxima en el circuito eléctrico y relacionar así la frecuencia de resonancia con el contenido volumétrico de agua.

TDR: Se mide el tiempo necesario, en nanosegundos, para que una señal electromagnética se propague a lo largo de una guía de ondas rodeada de suelo. El tiempo de viaje, o la velocidad, de este pulso es afectado por la constante dieléctrica (K_a) del suelo. Un suelo más húmedo con una constante dieléctrica más alta produce un pulso de velocidad más lento [28].

El TDR mide la cantidad volumétrica del suelo (VSW%) independientemente de todas las demás variables del suelo, incluida la densidad, textura, temperatura y salinidad.

En la Tabla 2 la comparativa de los métodos:

	Resistivo	**TDR**	**Capacitance**
Precio	Bajo	Moderado a alto	Bajo a moderado
Precisión	Baja	Alta (con calibración suelo específico)	Alta (con calibración suelo específico)
Complejidad	Fácil	Fácil a intermedio	Fácil
Uso energético	Bajo	Moderado a alto	Bajo a moderado
Sensibilidad a la Salinidad	Extrema	Ninguna en salinidad baja. Si en salinidad alta	Ninguna en salinidad baja. Si en salinidad alta
Durabilidad	Baja	Alta	Alta
Volumen de influencia	Pequeño entre filo A y filo B	0.25 a 2 L dependiendo en el tamaño y largo del sensor y de la forma del campo electromagnético	0.25 a 2 L dependiendo en el tamaño y largo del sensor y de la forma del campo electromagnético

Tabla 2 - Comparativa de métodos de sensado de humedad del suelo [28]

El sensor elegido funciona mediante TDR y se detalla en el siguiente apartado.

3.3.3. Sensor de humedad de suelo Vegetronix VH400

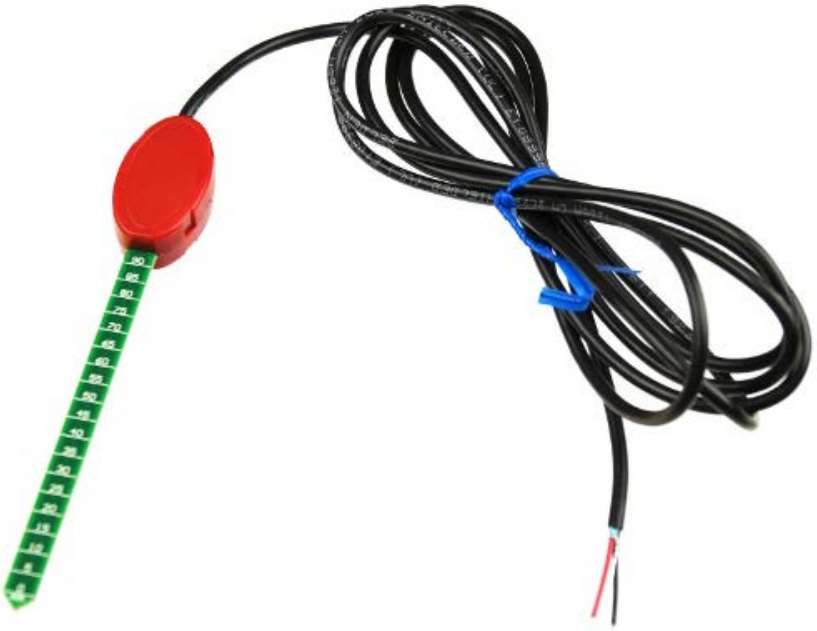

Figura 19 - Sensor de humedad de suelo

De procedencia americana y con un costo de $39.95 [30] (con dos metros de cable), este sensor de relativo (Ver Figura 19) bajo consumo (12mA cuando se activa) tiene como salida una señal analógica de voltaje de 0 a 3V relacionada directamente con el VWC del suelo, permitiendo ser entrada analógica de la Arduino sin necesidad de un divisor resistivo. La Arduino seleccionada cuenta con un conversor A/D de 10 bits (0-1024) y se entiende que no es necesario incluir un módulo conversor de mayor resolución. En la Tabla 3 se pueden apreciar sus especificaciones

Sensor VH400	
Consumo	< 13mA
Voltaje de operación	3.5V a 20 VDC
Power on para salida estable	400 ms
Impedancia de salida	10Kohms
Temperatura de operación	-40 to 85 °C
Precisión a 25°C	2%
Salida	0 to 3V

Tabla 3- Especificaciones sensor VH400

Para su calibración el fabricante proporciona un código que relaciona el voltaje con el VWC obtenido experimentalmente en tierra para macetas (Ver Figura 20).

Ecuación de rango de voltaje [31]:

- 0 a 1.1 V $VWC = 10 * V - 1$
- 1.1V a 1.3V $VWC = 25 * V - 17.5$
- 1.3V a 1.82V $VWC = 48.08 * V - 47.5$
- 1.82 V a 2.2 V $VWC = 26.32 * V - 7.89$

La gran mayoría de los suelos tienen una capacidad de retención de agua de menos del 50%, por lo que las curvas se detienen en 2,2 V, lo que representa el 50% del VWC. Por encima del 50%, se utiliza la siguiente aproximación:

- 2.2V a 3.0V $VWC = 62.5 * V - 87.5$

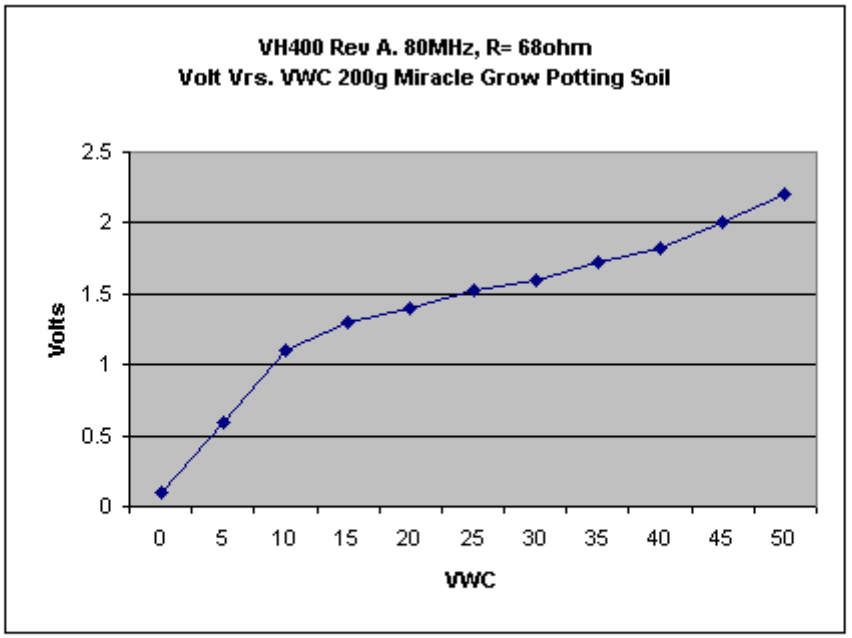

Figura 20 - Curva de VWC a voltaje para un contenedor con 200g de tierra para macetas [30]

El Vegetronix VH400 también puede ser utilizado como sensor de nivel de agua (por eso las marcas de división cada 5mm en su hoja), siendo el voltaje proporcional al nivel.

3.3.4. Sensor de temperatura del suelo DS18B20

Figura 21 - Sensor de temperatura de suelo

El DS18B20 [32] es un termómetro digital que permite medir temperaturas con precisión en ambientes húmedos (Ver Figura 21). Ofrece la posibilidad de configurar la resolución de las lecturas entre 9-bit y 12-bit. Utiliza la tecnología 1-Wire®, que consiste en una única interfaz de comunicación. Por lo tanto solamente requiere de un pin de comunicación con el microprocesador, y utiliza los otros dos cables para alimentación y tierra, obteniendo la referencia para brindar los datos acertadamente. Mide temperaturas entre -55°C y +125°C con una precisión de ±0.5°C entre -10°C y +85°C.

3.3.5. Sensor de temperatura y humedad relativa del aire DHT21 (AM2301)

Figura 22 - Sensor de temperatura y humedad relativa del aire

El DHT21 [33] es un sensor digital de temperatura y humedad relativa (Ver Figura 22). Contiene un sensor de humedad capacitivo, un termistor y un microcontrolador, el cual se encarga de convertir las medidas analógicas a lecturas digitales. Al igual que el DS18B20 utiliza 1-Wire®, y las conexiones son las mismas para ambos sensores. Tanto la temperatura como la humedad tienen una resolución fija de 16-bits. Mide temperaturas entre -40°C y +80°C con una precisión de ±0.5°C, y humedades relativas entre 0 y 100% RH con una precisión de ±3% RH. Es un sensor de empaque robusto y es por eso que dentro de su gama es el recomendado para exteriores.

3.3.6. Integración de los sensores digitales

Los DS18B20 y DHT21 son sensores estándar sumamente utilizados, lo que simplifica su integración al sistema ya que se puede encontrar numerosas librerías para facilitar su incorporación en el firmware, así como también tutoriales para su anexado a nivel de hardware (implica resistencias eléctricas)[34][35].

Capítulo 4 - Nodo integrado

4.1. Consumo energético

En la gran mayoría de los dispositivos IoT, un factor determinante es la optimización del consumo energético para lograr una máxima duración en la batería. El bajo consumo es clave más aún cuando el dispositivo es instalado en el campo donde no hay tendidos eléctricos o tomacorrientes cercanos hasta los puntos de medida.

Para lograr una solución de bajo consumo es necesario contar con el hardware adecuado, un firmware que optimice los procesos y contar con un modo “sleep”, para cuando el dispositivo se encuentra inactivo entre medida y medida.

En una primera instancia, y observando soluciones como la de Libelium [36], se pensó en utilizar un panel solar de baja potencia, en conjunto con un regulador de carga, para que cargue la batería. Según lo investigado y según comentarios sobre experiencias en trabajos dentro de la Universidad, el mantenimiento del panel, su correcta limpieza y direccionamiento, hacen que esta alternativa sea compleja y costosa. Además, las especificaciones de consumo de la Arduino elegida, con corrientes en el orden de los µA, prometen lograr un dispositivo que su batería dure años con una batería de algunos Ah.

La Arduino se puede alimentar de varias maneras: a través del puerto USB con 5V, mismo puerto con el que se programa y se utiliza la comunicación serial, a través de una batería estándar Li-Po o Li-Ion de 3.7V, para la cual la placa cuenta con un puerto JST de dos pines, o bien alimentando directamente el micro y el módulo de comunicación con 3.3V. Esta última opción es la más eficiente energéticamente pero implica intervenir la placa cortando un jumper soldado.

Se decide utilizar una batería de 3.7V estándar 18650, con un portapilas y conector JST. Se comprobó experimentalmente la importancia de utilizar baterías de alta calidad, aunque esto implique elevar un poco el costo, frente a baterías de baja calidad, como la Ultrafire Li-Po de 8000mAh, que no tienen la carga que dicen en la etiqueta. Se consiguió una batería Li-Ion Samsung recargable de 3000mAh, de notoria mejor calidad que las Ultrafire.

Se investigó sobre las funciones de bajo consumo que ofrecen las librerías Arduino, en especial para la MKR WAN 1310 [37] con el micro (MCU) SAMD21.

El Power Manager (PM) de la MCU es el responsable de distribuir los relojes, supervisando así los modos de bajo consumo y también administrar los reinicios de la MCU [38]. El mismo está compuesto de tres módulos: el controlador de reloj síncrono, el controlador de suspensión y el controlador de reinicio. Los modos de bajo consumo para el SAMD21 son: IDLE y STANDBY.

STANDBY es el modo de menor consumo de energía: los relojes del sistema se desactivan y los reguladores de voltaje son seteados para funcionar en modo de bajo consumo. Se entra en este modo a través de software ejecutándose la instrucción Wait For Interrupt (WFI) mientras el flag SCR.DEEPSLEEP está puesto en “1”.

IDLE, en cambio, detiene el reloj de la MCU pero deja los periféricos en funcionamiento. Existen tres niveles de modo IDLE en el SAMD21, éstos apagan selectivamente los relojes que se consideran innecesarios para ahorrar energía. Se entra en este modo al ejecutar una instrucción WFI con el flag SCR.DEEPSLEEP en “0” y el nivel de suspensión en SLEEP.IDLE.

A continuación en la Tabla 4 se aprecian los detalles de los tres modos de IDLE y el modo STANDBY y sus implicancias a nivel de hardware [38]:

Sleep Mode	CPU Clock	AHB Clock	APB Clock	Oscillators				Main Clock	Regulator Mode	RAM Mode
				ONDEMAND = 0		ONDEMAND = 1				
				RUNSTDBY=0	RUNSTDBY=1	RUNSTDBY=0	RUNSTDBY=1			
Idle 0	Stop	Run	Run	Run	Run	Run if requested	Run if requested	Run	Normal	Normal
Idle 1	Stop	Stop	Run	Run	Run	Run if requested	Run if requested	Run	Normal	Normal
Idle 2	Stop	Stop	Stop	Run	Run	Run if requested	Run if requested	Run	Normal	Normal
Standby	Stop	Stop	Stop	Stop	Run	Stop	Run if requested	Stop	Low power	Low power

Tabla 4- Modos de sleep en SAMD21 [38]

La librería ArduinoLowPower.h fue la utilizada ya que ésta provee las funciones de bajo consumo para placas con el microcontrolador SAMD21. En particular se utilizó la función LowPower.deepSleep() [37], el modo de mayor optimización de energía pero con el tiempo de activación más lento. Todos los periféricos están detenidos. Se obtiene una corriente de sleep de 210 µA, medida con un multímetro de precisión de los laboratorios de la UCU.

Existen dos maneras para "despertar" a la Arduino del modo de bajo consumo o pasar al modo ACTIVE nuevamente: interrupciones por hardware externo o timers y WDT (watch dog). Si por alguna razón (por algún error en el código) la Arduino permanece dormida para siempre, se debe resetear manualmente desde el botón físico de reset, vaciando la memoria flash y reprogramando. En el caso de la función utilizada, la MCU puede activarse sólo mediante RTC interno (no es el caso de la MKR WAN 1310 ya que no lo tiene) o mediante interrupción externa en algún pin previamente seteado.

Para obtener una interrupción externa periódica se incluye un módulo de reloj RTC (Real Time Clock) DS3231 [39] muy preciso y de bajo costo (Ver Figura 23). El mismo cuenta con un oscilador de cristal integrado con compensación de temperatura (TCXO) e incorpora una batería, manteniendo el cronometraje preciso cuando la alimentación principal del dispositivo es interrumpida. El RTC lleva la cuenta de segundos, minutos, horas, día, fecha e información de mes y año. Para la aplicación nos interesan las alarmas programables y la salida de onda cuadrada como señal de interrupción en el pin digital previamente configurado en la Arduino. La dirección y los datos se transfieren en serie a través de un bus bidireccional digital I2C.

Adicionalmente, se le extraen los LEDs del módulo, para esta aplicación innecesarios, de forma de optimizar el consumo energético (cada LED consume 2mA).

Figura 23- Módulo RTC DS3231

Otro factor clave para optimizar el consumo, es poder controlar el encendido y apagado de los sensores que están directamente conectados a la batería. Sin esta función, el consumo sería constante, por ejemplo, el sensor de humedad de suelo Vegetronix (alimentado directamente con la batería y el que tiene consumo más alto) siempre va a transmitir la señal analógica de voltaje en su cable de salida. Para resolver esto, se decidió integrar al circuito un transistor MOSFET y utilizarlo tipo llave, abriendo y cerrando a demanda, por software, el circuito de alimentación del sensor.

Para que este componente sea controlable se eligió el MOSFET Fairchild IRF540N (33A, 100V, 0.040 Ohm, N-Channel) [40], cuyo voltaje de threshold entre Gate y Source es en el entorno de [2, 4] V. El mismo es controlado desde la Arduino levantando un pin digital a 3.3V, previo a la lectura del sensor y bajando a 0 V una vez que se obtuvo la medida. Así se logra controlar el consumo del sensor, alimentándose únicamente cuando es necesaria su lectura. Se le agrega un delay de 400ms (dado por el fabricante) luego de ser alimentado para estabilizar la medida.

Para conocer el nivel de batería del dispositivo es necesario saber el voltaje en bornes de la misma y enviarlo como parte del payload. Como el límite de voltaje en los pines analógicos de la Arduino es de 3.3V se agrega un divisor resistivo para reducir el voltaje y luego convertirlo al valor real vía software. Se elige una relación entre las resistencias (se muestra en la Figura 24) para lograr una corriente de 2µA, despreciable frente al consumo del

dispositivo. La lectura del voltaje de la batería se calibra vía software, previamente midiendo el voltaje de la batería con un multímetro.

4.2. Autonomía del nodo

A través de una planilla dinámica (Anexo G) se realiza el ejercicio teórico de calcular y estimar la autonomía del nodo. Conociendo las corrientes máximas del ciclo de funcionamiento del nodo: 30mA en la toma de medidas y envío, 210µA en modo sleep, se puede variar la frecuencia de envío así como la carga de la batería y obtener la autonomía del nodo. En el caso implementado con una frecuencia de medida y envío cada 30 minutos y una batería de 3Ah se logra una autonomía teórica de 2.2 años.

4.3. Armado del nodo

A continuación se presenta un esquemático del conexionado. El mismo se realizó en una protoboard soldando con estaño cada uno de los componentes y se coloca dentro de una caja estanco de dimensiones 115x115x81mm con protección IP65. La misma es agujereada y a través de un prensacable, también de exterior, se enhebran los cables de los sensores y de la antena (reforzada con silicona a la entrada del cable). Se logra un dispositivo compacto, liviano y con la protección necesaria para exterior.

En las Figuras 25, 26 y 27 se puede apreciar un registro fotográfico de la integración y el armado del nodo a caja estanco abierta, y en la Figura 28 con la caja estanco cerrada.

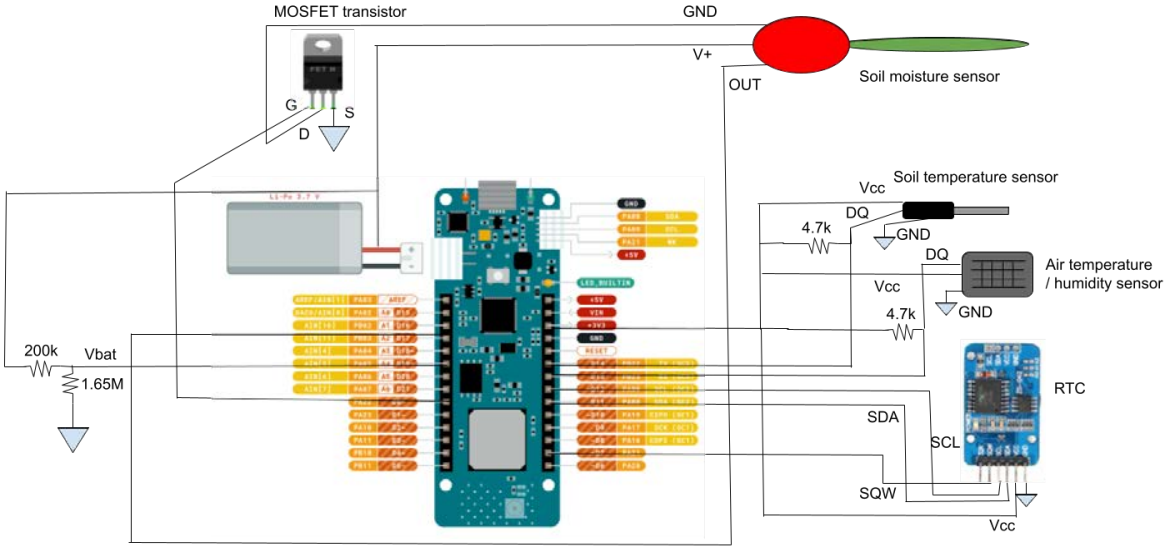

Figura 24 - Esquemático del sistema

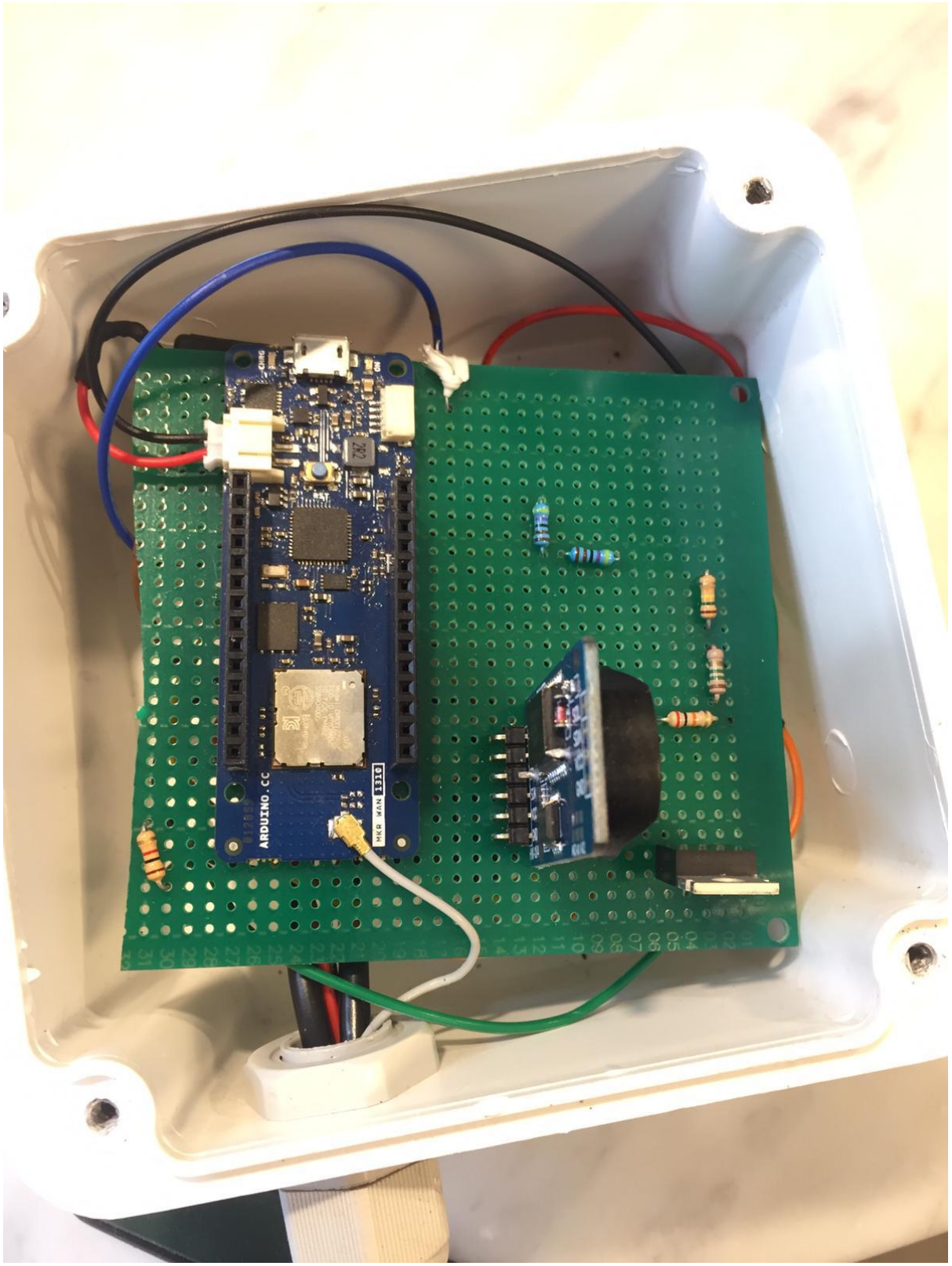

Figura 25 - Sistema integrado

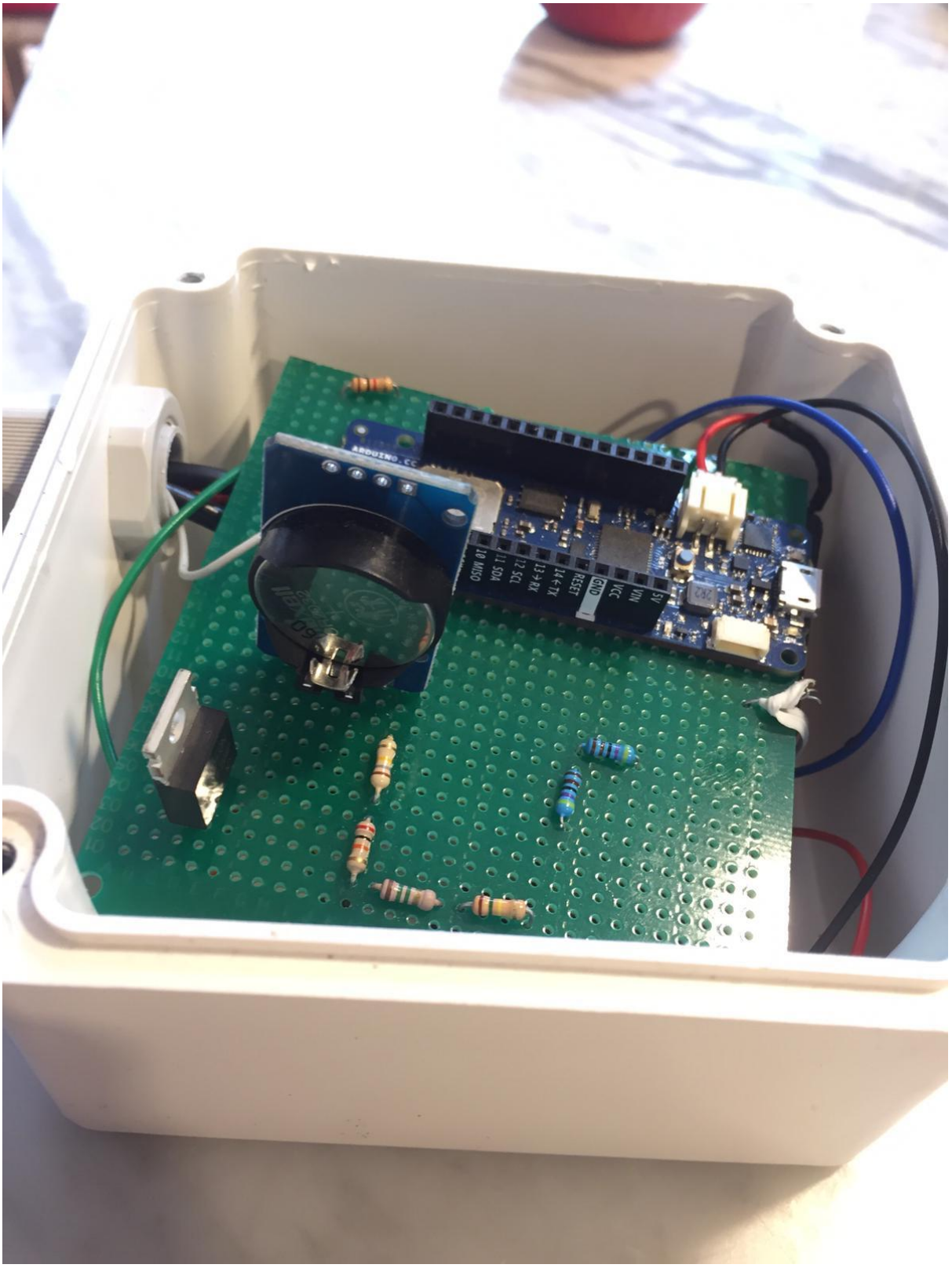

Figura 26 - Sistema integrado

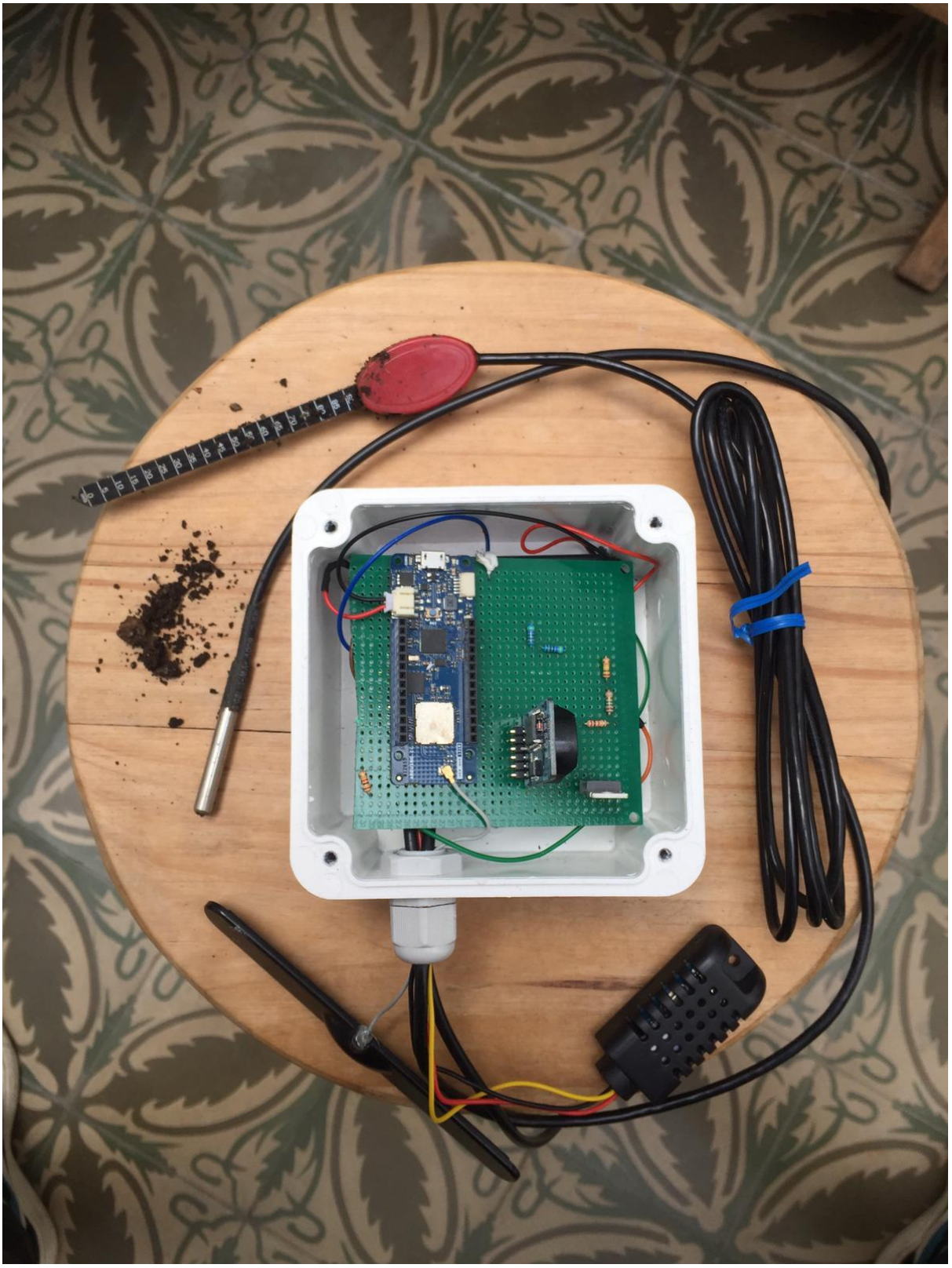

Figura 27 - Sistema integrado

Figura 28 - Nodo cerrado

4.4 Costos

A continuación se presenta una planilla con los costos reales asociados. No está incluido el costo de envío en el caso de las compras en el exterior, ni mano de obra. La UCU cubrió los costos de la placa Arduino, el gateway LoRaWAN Dragino y el sensor de humedad del suelo Vegetronix, los únicos ítems comprados en el exterior (Estados Unidos). El resto de los ítems se consiguieron en plaza en tiendas de electrónica locales.

	Pesos uruguayos	USD (42.7)
Arduino MKR WAN 1310	1622.6	38.00
Caja estanco IP65	365	8.55
Modulo RTC	231	5.41
Pila Modulo RTC	50	1.17
Batería Samsung 3.7V 3000mAh	896.7	21.00
Resistencias y cables	50	1.17
Transistor MOSFET IRF540N	50	1.17
TOTAL NODO	**3265.3**	**76.47**
Soil moisture sensor Vegetronix	1705.865	39.95
Sensor temperatura DS18B20	165	3.86
Sensor temp y humedad aire	315	7.38
Dragino LPS8 LoRaWAN gateway	4184.6	98
TOTAL 1 nodo + sensores + gateway	9636	225.67

Tabla 6 - Costos

4.5. Comparación con otras soluciones en el mercado

En primer lugar, se reconoce que no existen productos estandarizados de sensores inalámbricos para esta aplicación en plaza, en Uruguay. Sí hay empresas locales, como Nettra, que pueden desarrollar a medida soluciones para lograr una agricultura de precisión [41].

Se realiza una búsqueda de soluciones de sensado mediante tecnología LoRaWAN en el exterior para hacer una comparativa de funcionalidades, rendimientos, dificultad de implementación y precios.

- Libelium Smart Agriculture:

Libelium [42] es una reconocida compañía española de desarrollo de tecnología IoT para diversas aplicaciones como industria, smart cities, monitoreo de agua, retail, agricultura y más. Los nodos que ofrecen son personalizables en cuanto a conectividad y sensores que se integran. Dentro de los productos que ofrecen para agricultura, el Smart Agriculture (Ver Figura 29) es el que más se ajusta a nuestra solución. Se pidió cotización con tecnología LoRaWAN y sensor resistivo de humedad del suelo, temperatura de suelo, temperatura y humedad del aire, así como el gateway LoRaWAN, al cual ellos llaman Meshlium. No recibimos el costo específico pero al investigar en la web se puede encontrar que el precio de la solución está en el entorno de €2500 [43], siendo enormemente más cara que la solución llevada a cabo en este proyecto.

La batería se puede cargar con un panel solar opcional.

Figura 29 - Nodo Libelium

- Nodo LoRa Dragino LSN50V2:

Dragino, de procedencia china, es el mismo proveedor que el gateway elegido para este proyecto, ofrece un nodo de exterior (Ver Figura 30), LoRa (bandas 433/868/915/920 MHz) de bajo costo [44], con batería Li/SOCI2 desechable de 4000 o 8500 mAh, una corriente de sleep de 2µA, de código abierto para integración de sensores. Cuenta con 18 entradas digitales y dos entradas analógicas. El precio del mismo es de 56,29 USD sin costo de envío, por lo que es una solución más económica que la nuestra (unos 25 USD de diferencia) y también con un mayor rendimiento de la batería. Se puede ver por los manuales que la integración de los sensores implica una dificultad mucho mayor que al programar en Arduino, así como la posibilidad de Arduino de integrar distintos shields para la comunicación de los sensores por ejemplo RS232 o RS485 muy común en diversos sensores.

Figura 30 - Nodo Dragino

- Moko Smart LW002-TH - LoRaWAN-Based Temperature & Humidity Sensor:

Moko Smart [45] es una empresa china que se dedica a fabricar dispositivos inteligentes de variada índole. El LW002-TH es un nodo de sensado completo (Ver Figura 31), incluye el sensor de temperatura y humedad del aire, la capacidad de integrar más sensores externos, así como la comunicación LoRaWAN. La batería es de 12000 mAh no recargable, y la corriente en sleep es menor a 20µA. El sensor de humedad es capacitivo y los intervalos de medida son configurables. El precio ronda los 65 USD en China, lo cuál lo hace una solución más económica que la presentada en este proyecto y los mismos comentarios en cuanto a la dificultad de integración de los sensores.

Figura 31 - Sensor Moko Smart LW002-TH

En base a esta breve investigación sobre soluciones similares podemos concluir que nuestro desarrollo con fines académicos no se aleja de la media en cuanto a costos, siendo un poco más cara que los nodos LoRaWAN de procedencia china, pero mucho más económica que el paquete ofrecido por Libelium. Otro aspecto a destacar es que nuestro nodo al ser una Arduino permite mucho más flexibilidad y versatilidad, con un entorno de desarrollo más amigable en cuanto a firmware y funcionalidades a agregar.

En cuanto al consumo y autonomía del nodo, la corriente de sleep conseguida de 200µA es uno y dos órdenes de magnitud mayor que los nodos Dragino y Moko respectivamente. Entendemos que esto se debe a que la electrónica de estos últimos fue diseñada específicamente para optimizar la corriente y conseguir dispositivos de ultra bajo consumo. La Arduino es utilizada, en general, para prototipado y está "sobredimensionada" para la aplicación desarrollada. Para abaratar la solución se podría diseñar la electrónica específica pero esto excede al alcance del proyecto.

Capítulo 5 - Desarrollo y plataformas

5.1. Software

A través del IDE (Integrated Development Environment) de Arduino se escribe el software, se compila y se programa la placa. Arduino cuenta con un lenguaje propio de programación basado en Wiring, que esté a su vez se basa en C y C++., al cual con los conocimientos de programación adquiridos en la carrera logramos familiarizarnos rápidamente y desarrollar el programa de una manera relativamente fácil.

Los programas Arduino constan de dos grandes funciones: "setup", donde se inicializan algunos parámetros variables, se asignan los pines de entrada, salida e interrupción, se inicializa el módulo de comunicación y la función "loop" que va ser la rutina que se ejecuta

repetitivamente. Ésta última se basa en la siguiente secuencia: obtener los datos de interés (lectura de los sensores y voltaje en batería), enviarlos y entrar en modo sleep. Se determina en el programa la frecuencia de este ciclo, es decir, cada cuanto se obtienen los datos y son enviados, y el dispositivo es despertado mediante la interrupción externa del RTC que cuenta con batería propia.

Las lecturas de los sensores al ser números con cifras decimales, son variables de tipo *float* de 4 bytes de tamaño. Podrían ser de tipo *double (8 bytes)* para una mayor precisión pero no se considera necesario, además que para la comunicación LoRa se busca minimizar el tamaño del payload. Para el armado del mismo con la información de interés, se utiliza un array de 20 bytes incluyendo los cinco *floats*: humedad del suelo, temperatura del suelo, temperatura del aire, humedad del aire y voltaje en la batería. El fragmento de código desarrollado para codificar cada una de las lecturas y armar el payload está en el Anexo E.

Se utilizan varias librerías que proveen funciones ya desarrolladas que facilitan mucho el desarrollo del código: bajo consumo, lectura de los sensores, configuración del módulo RTC, comunicación LoRa y comunicación I2C.

En el archivo de comunicación LoRa se setean por código los parámetros para la activación ABP (Ver Anexo H).

Estas direcciones se eligen aleatoriamente pero deben coincidir con lo configurado en TTN, servidor LoRaWAN que será explicado en el siguiente punto.

Para este proyecto sólo se utilizan mensajes de uplink. Es decir, el dispositivo nunca recibe mensajes de la red, mensajes downlink, comúnmente utilizados para realizar configuraciones remotamente o bien activar actuadores.

Otro parámetro a configurar para la comunicación LoRa es el Spreading Factor (SF). El SF decide cuántos chirps se envían por segundo, así como también la cantidad de bits/chirp. Se decide el SF (entre 7-12) de la red en función de las condiciones de propagación y las ambientales entre el nodo y el gateway.

Un SF más bajo significa que se envían más chirps por segundo, es decir, se puede codificar más datos por segundo. Un SF más alto implica menos chirps por segundo, por lo tanto, hay menos datos para codificar por segundo y se necesita más tiempo de transmisión, conocido

como Time On Air (ToA), lo que implica que el módem está funcionando durante más tiempo y consume más energía (Ver Figura 32).

Como contrapartida, la ventaja de un SF alto es que un ToA mayor le da al receptor más oportunidades para muestrear la señal, resultando en una mejor sensibilidad. Esto, a su vez, significa que puede recibir la señal más lejos, por lo que se obtiene una mejor cobertura.

También, en función del tamaño del payload y resultados experimentales, se setea el SF (Spreading Factor) en 9, con la función DataRate(3), un ancho de banda de 125(kHz)

DataRate	Modulation	SF	BW (kHz)	bps
0	LoRa	12	125	250
1	LoRa	11	125	440
2	LoRa	10	125	980
3	LoRa	9	125	1760
4	LoRa	8	125	3125
5	LoRa	7	125	5470
6	LoRa	7	250	11000

Tabla 5 - modos DataRate Arduino MKR WAN 1310

Se configura con *modem.dataRate(3)* ya que con el valor por defecto (4) no se recibían los datos correctamente en la plataforma de visualización.

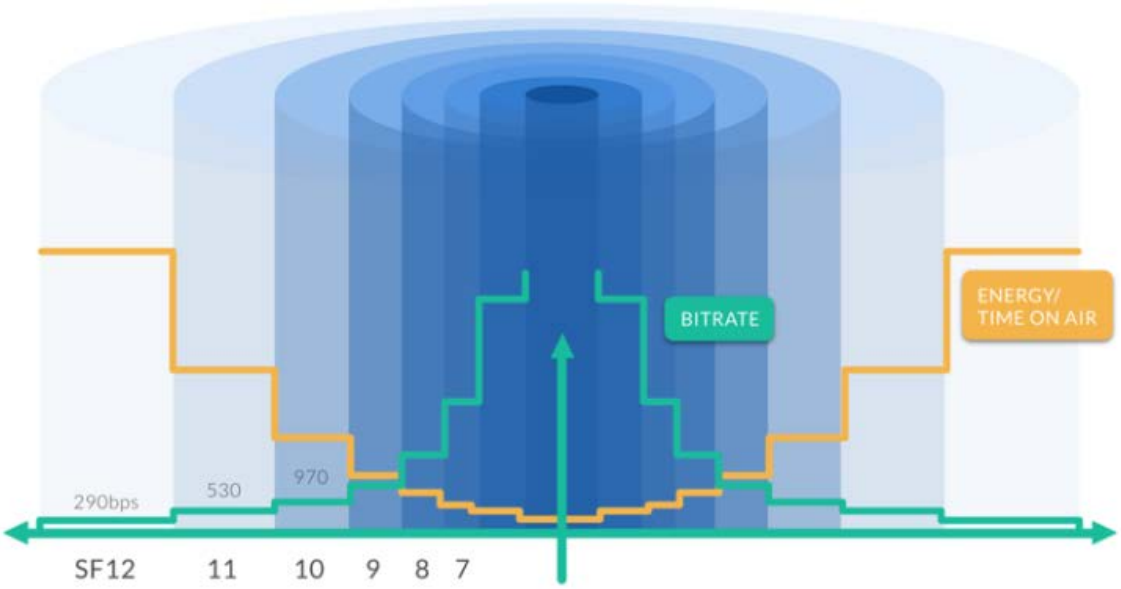

Figura 32 - Relación entre el Spreading Factor y el Time On Air para la modulación LoRa [48]

Existen calculadoras online para calcular el ToA según el SF, ancho de banda y el tamaño del payload [49]. Con el payload de 20 bytes y SF=9 se obtiene un ToA de 246.78 ms y un duty cycle de 25 s.

5.2. The Things Network

The Things Network (TTN) [47] es un ecosistema IoT global y colaborativo, que sirve para crear redes LoRaWAN, integrando gateways y dispositivos finales con sus servidores de red y de aplicación. Es una potente herramienta para construir soluciones IoT a bajo costo, con un uso abierto de sus redes de servidores e integraciones, permitiendo crear redes con máxima seguridad, altamente escalables. Además, brinda una extensa documentación en su plataforma The Things Stack, acerca de todas las aristas que componen una solución LoRaWAN, devices, gateways, guías de construcción de redes, integraciones con terceros, etc. Está presente en 151 países, con más de 150 mil miembros y más de 22 mil gateways conectados a sus servidores [46]. Se creó una cuenta gratuita en TTN para poder utilizarlo.

A través de la configuración de nuestro gateway, que se realiza mediante un servicio web del equipo, se selecciona la frecuencia a utilizar y se “apunta” la información recibida por el

gateway al servidor TTNv3. Posteriormente, se agrega nuestro gateway a la consola TTN, registrándose con su identificador.

Se podría haber prescindido del uso de TTN y enviar los mensajes modulados LoRa a nuestro gateway, previamente activando el dispositivo mediante ABP, y reenviar los datos mediante MQTT (Message Query Telemetry Transport), protocolo muy utilizado en IoT que funciona sobre TCP/IP, de tamaño muy pequeño. No se tomó esta vía dadas las ventajas de simplicidad que ofrece utilizar TTN detalladas a continuación.

El dispositivo final envía una cadena de bytes, el payload, con la información de los sensores en cada una de sus entradas. A pesar de que la información llegue a TTN, a priori, es una información ilegible para un humano. Por lo que es necesario utilizar una función que reciba, ordene y traduzca (decodifique) esa cadena de bytes en los parámetros de nuestro interés: humedad y temperatura del suelo y del aire así como el voltaje de la batería (Ver Figura 33). TTN ofrece una herramienta de Payload Formatter para esto, por lo que se crea una función en JavaScript que recibe los bytes, los convierte a float y los reconoce según sus posiciones en el array como las variables esperadas. El código desarrollado se encuentra en Anexos F.

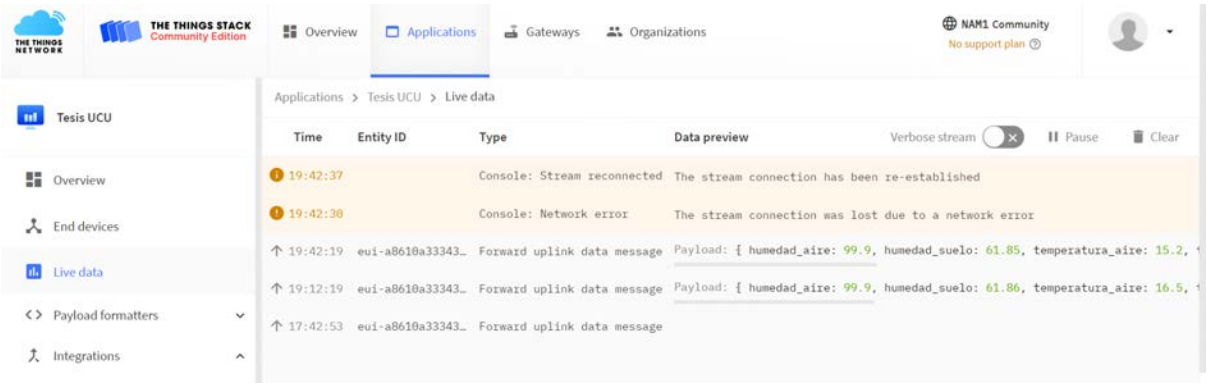

Figura 33 - Pantalla de recepción del payload TTN

Una vez recibidos y decodificados los datos en TTN se realiza la integración con la plataforma de visualización seleccionada, TagoIO [50] la cual se va a detallar en el siguiente apartado. La integración fue muy fácil, a través de una serie de parámetros:

- Process ID: nombre alfanumérico en minúsculas único para la aplicación.
- Access key: clave de acceso.

- Authorization: clave de seguridad única generada por TTN que debe coincidir con la de TagoIO.

5.3. TagoIO

TagoIO [50] es una plataforma web que combina un manejo avanzado de dispositivos, servicios, almacenamiento de datos, visualización y analíticas, basado en una tecnología cloud con una amplia gama de funcionalidades de fácil implementación y altamente customizables.

Puede ser empleada para una infinidad de soluciones IoT, siendo en general la última etapa de las soluciones al brindar una interfaz de usuario que despliega mediante dashboards personalizables toda la información recolectada por el "backend", que son las redes de servidores, gateways y dispositivos finales. En el otro extremo de la solución, se encuentran los dispositivos finales o nodos, que pueden utilizar variedad de tecnologías como LoRaWAN, Sigfox, Zigbee, WiFi, Narrowband LTE, entre otras. Estos recolectan datos mediante sensores y los envían a gateways o servidores que componen el core de las redes.

A nivel de TagoIO, se selecciona un template y se integran las soluciones para traer los datos de interés y desplegarlos en el dashboard creado. Esta plataforma también brinda la posibilidad de crear y administrar distintos usuarios, que podrán tener permisos de editor o simplemente poder monitorear el sistema [51].

Los dashboard pueden incluir:

- Datos en tiempo real, que pueden ser desplegados en distintos widgets
- Gráficos con los históricos de datos
    - Curvas en 2 ejes
    - Circulares o "de torta"
    - Verticales de columnas u horizontales de barras
    - Área bajo la curva
- Imágenes

- Videos
- Tablas
- etc.

Los datos almacenados pueden ser exportados en formato .XLSX o .CSV para su análisis. A su vez, también se pueden crear reglas que disparen alarmas vía correo electrónico o SMS ante algún evento, o bien, si una variable determinada sube o baja de un umbral configurable. Se crean alarmas vía email ante dos eventos: batería baja y suelo muy seco (Ver Figura 36 y 37).
En nuestra aplicación se crea un dashboard (Ver Figura 35), con cada una de las medidas “en tiempo real” (las últimas muestras) y dos gráficas históricas de las condiciones del suelo y del aire donde se puede visualizar claramente cuando se efectuó el riego, o cómo se va secando el suelo a medida que pasa el tiempo. El tiempo o cantidad de muestras desplegadas es configurable. También se agrega una imagen satelital, meramente ilustrativa, con la ubicación exacta del lugar donde fue instalado el nodo. TagoIO no admite función de mapa, por ejemplo, tener varios nodos desplegados en el campo e ir seleccionando de a uno para ver las lecturas, lo que consideramos como una pequeña desventaja si nuestra solución implicase más de un nodo.
Se agrega el voltaje de la batería así como el RSSI (Indicador de fuerza de la señal recibida). Éste último parámetro es agregado en el gateway y es la potencia de la señal recibida en mW medido en dBm negativo. Este valor sirve como medida de qué tan bien un receptor puede "escuchar" la señal de un emisor.
Los valores típicos de LoRa RSSI son:

- RSSI mínimo aceptable= -120 dBm.
- Si RSSI = -30dBm: la señal es fuerte.
- Si RSSI = -120dBm: la señal es débil.

Se obtuvo el código QR de la dirección web del dashboard de la aplicación para facilitar el acceso al mismo (Ver Figura 34).

Figura 34 - Código QR del dashboard público TagoIO

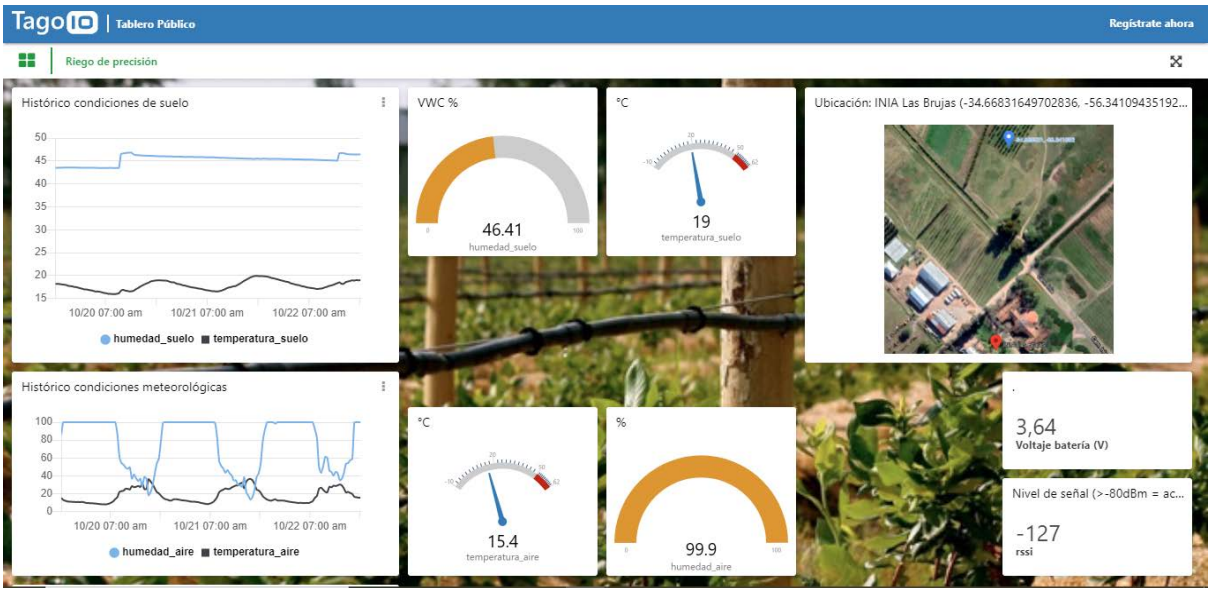

Figura 35 - Dashboard TagoIO

Figura 36 - Alarma vía email de batería baja

Figura 37 - Alarma vía email de suelo seco

Capítulo 6 - Implementación, resultados obtenidos y conclusiones

6.1. Pruebas de campo

Se realizan las primeras pruebas en interior, instalando el dispositivo en una maceta (Ver Figura 38), bajo una claraboya y a unos pocos metros del gateway. Se verifica que llegan los datos a la plataforma, sin pérdidas, y que las lecturas de humedad del suelo coinciden con el riego.

Figura 38 - Nodo midiendo en entorno controlado

Luego de esta primera prueba, nos pusimos en contacto nuevamente con el INIA para comentarle sobre los avances del proyecto y que ya disponemos del dispositivo listo para dejarlo funcionando en sus instalaciones de Las Brujas. Se coordina una fecha y hora para realizar una presentación del proyecto e instalar el nodo donde ellos dispongan.

Nos recibe el Ing. Agrónomo (Dr.) Claudio García, Investigador Principal en las instalaciones de la Estación Experimental "Wilson Ferreira Aldunate", INIA Las Brujas, ubicadas al suroeste del departamento de Canelones, a unos kilómetros del centro de Montevideo. Ésta es una zona de mucha plantación frutícola y hortícola. En el predio del INIA se realizan diversas tareas de investigación, ensayos, contando con laboratorios, para diversos tipos de cultivo.

Luego de la presentación sobre el proyecto y su proceso, en primer lugar se instala el gateway LoRaWAN en el interior de una oficina del edificio central a través de un cable Ethernet a uno de los routers que proveen de internet al predio. Luego, con el objetivo de exigir y poner a prueba la comunicación inalámbrica, se decide instalar el nodo a más de 300 metros del gateway (Ver Figura 39), en un bajo y con grandes obstáculos, como las paredes del edificio central y árboles de gran tamaño.

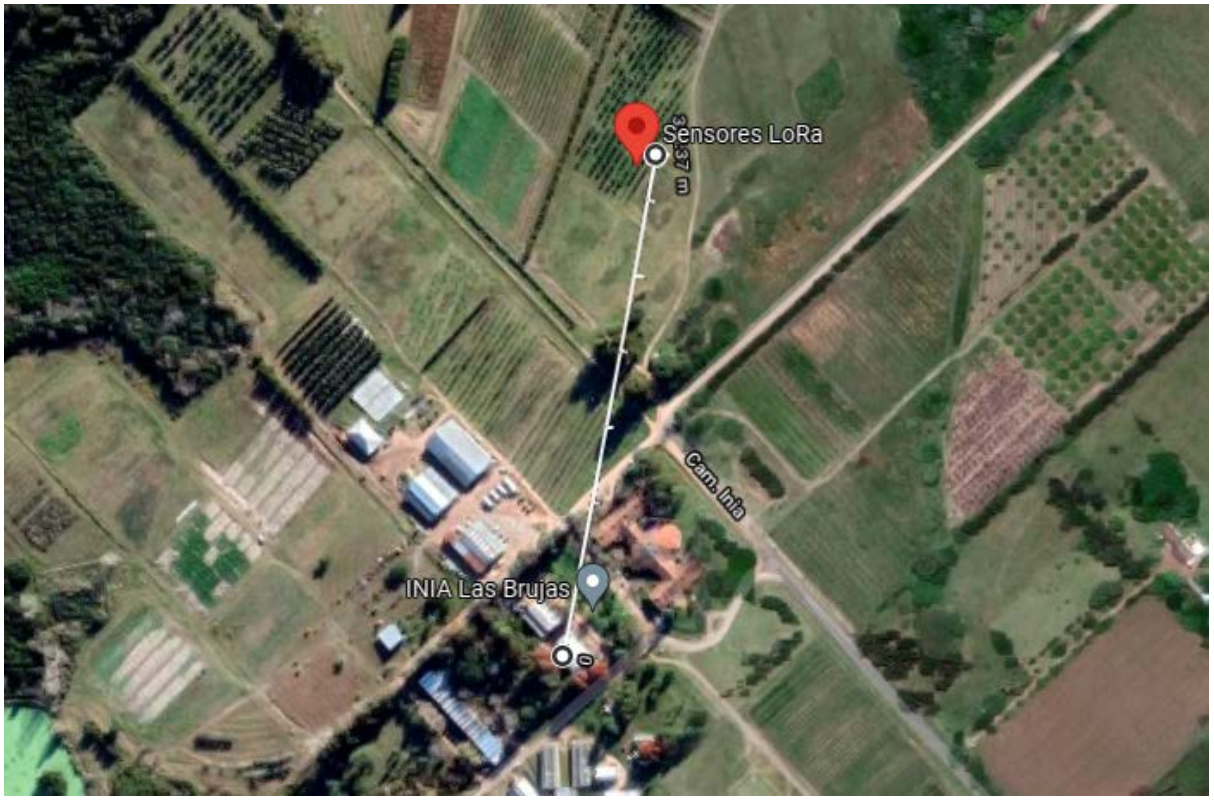

Figura 39 - Imágen satelital INIA Las Brujas, distancia entre el nodo y el gateway

El cultivo elegido fue unos ensayos de manzanos, ya en flor, con cosecha estimada para enero de 2022 donde se riega por goteo a demanda. Se observó la válvula por la que se activa el riego, la cual tenía integrado un caudalímetro mecánico para conocer el volumen de agua suministrado al cultivo. Se comentó que podría integrarse un nodo adicional para obtener las lecturas de un caudalímetro eléctrico así como una electroválvula (actuador) para activar el riego. Ésto último implicaría que el nodo reciba mensajes de downstream por lo que se necesitaría de otro desarrollo de firmware.

El proceso de instalación fue sencillo: con una pala se hizo un pequeño pozo, a unos 25 cm se clavó en la tierra horizontalmente el sensor de humedad del suelo y a unos 15 cm el sensor de temperatura del suelo. Se observa, y nos comenta el Dr. Garcia , que es un suelo más bien arcilloso, con gran capacidad de retención de agua.

Previamente, a través de una varilla metálica tipo estaca, se le dió estabilidad al dispositivo, sujetando la caja estanco a la misma. La antena y el sensor de humedad y temperatura del aire quedaron sujetados con precintos plásticos a la varilla.

Se verificó mediante la plataforma en el celular que las medidas llegaban correctamente. El dispositivo se configuró para enviar las lecturas cada 30 minutos.

A continuación en las Figuras 40, 41, 42, 43 y 44 un registro fotográfico de la instalación y ubicación de los sensores:

Figura 40 - Instalación del nodo en cultivo de manzanas

Figura 41 - Instalación de los sensores en el suelo

Figura 42 - Instalación del nodo en cultivo de manzanas

Figura 43 - Instalación del nodo en cultivo de manzanas

Figura 44 - Instalación del nodo en cultivo de manzanas

6.2. Resultados obtenidos de la prueba de campo

Lo primero a destacar como resultado es que desde el INIA se mostraron muy conformes desde un principio con la solución. En comparación con los sensores utilizados por ellos, la primera gran ventaja es que el nodo desarrollado en este proyecto envía los datos constantemente y es totalmente independiente. Mientras que los que se utilizan generalmente en el instituto almacenan los datos en una memoria local, por lo que es necesario que el usuario vaya al sitio y recoja los sensores para poder obtener las lecturas. Los sensores se conectan a una interfaz que mediante USB se conecta a una PC y con un software levanta los datos. Todo este proceso es bastante tedioso para los investigadores, ya que implica un traslado hasta el sitio y un proceso más bien manual de obtención de datos, los cuales se reciben "crudos" en planillas de Excel. Nuestra solución parece ser ampliamente superior en este apartado, ya que en ningún momento es necesario un movimiento por parte de los investigadores. Los datos se reportan automáticamente en el intervalo de tiempo que se desee, y se despliegan en un dashboard mucho más intuitivo que una planilla de Excel, que además puede ser accedido en tiempo real desde cualquier dispositivo y ubicación donde se cuente con conexión a internet. Además de tener datos en “tiempo real”, lo que es inviable con la solución actual utilizada en INIA, debido al proceso de obtención de datos ya mencionado, cabe destacar que en dicho proceso se deja de obtener medidas. Todos estos inconvenientes están resueltos más acertadamente en la solución presentada en este proyecto, lo que decanta en una mejor experiencia de usuario, y más actualizada en cuánto a lo que es la tecnología en los tiempos que corren.

La comunicación inalámbrica del nodo, desde que permaneció instalado, fue ininterrumpida. Es decir, no se perdió ni un sólo envío desde el momento de la instalación. Se obtuvo un mes de lecturas cada 30 minutos hasta que la batería Sanyo de 3000mAh, utilizada en un primer momento, de procedencia y tiempo de uso desconocido, tuvo que ser sustituída (Ver Figura 45) por una nueva de marca Samsung de la misma carga. La misma, continúa funcionando en el dispositivo al momento de la entrega de este documento.

Figura 45 - Cambio de batería en el nodo

En cuanto a la autonomía energética, la batería Sanyo usada, duró unos 60 días desde la carga y puesta en funcionamiento con lecturas cada 30 minutos. Por lo mencionado anteriormente, no es referencia para saber realmente la autonomía del dispositivo. Si lo es la batería Samsung nueva que continúa activa.

Sabemos que para lograr una mayor autonomía, las lecturas podrían estar más separadas en el tiempo. Es decir, la frecuencia de lectura y envío podría ser menor, por ejemplo cada 1 o 2 horas, logrando así períodos de sleep más largos con la corriente de 210uA.

En cuanto a la coherencia de los datos obtenidos, los datos de humedad y temperatura del aire fueron validados por el Dr. García , quien los comparó con una estación meteorológica ubicada junto al edificio principal del INIA Las Brujas. Los datos en general coinciden, y las diferencias, según el Dr. García , se deben a la distinta ubicación de la estación, su altitud y que los sensores se encuentran recubiertos según ciertos estándares. Los valores de humedad del suelo, desde un primer momento, no coinciden con lo que a simple vista es el estado del suelo. Los mismos fueron siempre unos puntos porcentuales por encima de lo observado. Esto se debe a la calibración del sensor, hecha por software según las curvas que brinda el fabricante Vegetronix, que dependen del suelo en donde se está midiendo. Esto, si bien nos presentó un desafío, es solucionable mediante el ajuste de la relación entre voltaje de salida del sensor y el contenido volumétrico de agua. La calibración se hizo comparando contra un lector de humedad del suelo FDR, tipo bastón (ver Figura 46), ya calibrado que da la lectura en un display. Se compararon las lecturas de ambos sensores, con una diferencia de 10 puntos porcentuales y se hizo un ajuste desde la plataforma para hacer coincidir los valores. Ésto fue suficiente para obtener valores consistentes y fue validado por el Dr. García.

Figura 46 - Calibración del sensor de humedad

Algo que nos llamó la atención es el RSSI obtenido a los poco más de 300 metros que fue instalado el nodo, teniendo en cuenta el alcance teórico de LoRaWAN. El mismo oscila en el entorno de -110 dBm, lo que es muy cercano al mínimo aceptable para evitar pérdidas (-120 dBm), por lo que la comunicación inalámbrica estaría siendo exigida al máximo. Entendemos que para acercarse a los valores teóricos de alcance se debería utilizar un gateway de mayor calidad, que además debería estar instalado en exterior para evitar las pérdidas de dBm por penetración de las paredes, y a una mayor altura que ayude a tener línea directa de vista con el nodo. Como el objetivo de este trabajo no era poner a prueba el alcance de LoRaWAN, se consideró aceptable y suficiente (no hubo pérdida de un sólo mensaje) para los objetivos puestos en primer lugar.

En cuanto a la cantidad de nodos por unidad de área necesarios para tener el monitoreo del cultivo entero, va a depender mucho del tipo de suelo, relieve (curvas de nivel) del mismo, separación entre las líneas y plantas, etc. Para definir esto es necesario realizar un relevamiento previo en cada cultivo específico. En el caso del escenario donde fue instalado el nodo, lote de 500m^2 (0.05 hectárea) sin pendiente pronunciada, se podría decir que las medidas de un único nodo son representativas del lote entero.

6.3. Conclusiones

En primer lugar, concluimos que los objetivos planteados al inicio de este proyecto fueron ampliamente cumplidos. Se transitó un camino de investigación y desarrollo para hacer esto, logrando aprendizajes muy valiosos en cuanto a estudio de mercado, estándares de protocolos de comunicación inalámbrica, integración de componentes electrónicos, programación de firmware, desarrollo de prototipos para uso exterior y conocimientos de agricultura en general.

Es sumamente satisfactoria la respuesta positiva del organismo nacional más especializado en tecnología para la actividad agropecuaria.

Entendemos que la solución obtenida, así como las demás encontradas en el mercado, permiten un uso más eficiente de un recurso natural tan valioso como es el agua. Ésto nos parece esencial para un desarrollo sostenible de la agricultura a nivel mundial para el futuro, en un mundo donde la población incrementa cada vez más, y con ella la necesidad de grandes cultivos para su alimentación. Este progreso va de la mano con una tendencia global que apunta a la conciencia sobre la utilización de los recursos del planeta, alineándose a uno de los Objetivos de Desarrollo Sostenible 2030 de la ONU (Organización de Naciones Unidas) "Objetivo 12: Garantizar modalidades de consumo y producción sostenibles" [52]. El dato de que la agricultura demanda el 70% del agua extraída en el mundo [2] y de esta agua, aproximadamente un 44% se desperdicia merece tomar medidas urgentes y la tecnología es el camino. Con soluciones de este tipo se mejora tanto el uso del agua, como los costos asociados al riego y también se simplifica el proceso para la toma de decisiones.

Si bien se logró e implementó una solución, se amplió la visión sobre qué tanto se puede seguir mejorando y agregando funcionalidades. Además de las lecturas de los sensores, a través de avances en big data e inteligencia artificial, se podría lograr una toma de decisión de riego añadiendo datos meteorológicos predictivos e imágenes satelitales que evalúan la sanidad de los cultivos mediante el estudio de su color (índice de verde). Está claro que cuanto más información se disponga, mayor será la precisión en las prácticas agrícolas.

Otro aspecto interesante es el hecho de poder anexar cualquier tipo de sensor al nodo desarrollado. Existen otros parámetros de interés en frutihorticultura, como el diámetro del

fruto (utilizando un dendrómetro) o la humedad de la superficie foliar, de las hojas, para prevenir la aparición de plagas u hongos que afecten negativamente la producción.

En cuanto a las soluciones en el mercado exterior, como fue analizado en el capítulo 4, entendemos que nuestro desarrollo, a diferencia de las soluciones más competentes en el apartado del precio, es “llave en mano”, es decir, que el productor no tiene que tener conocimientos de tecnología ni programación para integrar sensores y hacer uso de la información recogida. De esta forma, se crea una solución más accesible desde todo punto de vista para los productores, lo que interpretamos como un pequeño paso que nos acerca al objetivo de lograr una mejor relación con nuestro planeta.

Bibliografía

Capítulo 1:

1. Beecham Research Ltd. [2016], UK, *"The Future of Farming through IoT Perspective"*.
2. Sitio web Banco Mundial, 12 Julio 2017 [online] disponible: https://www.bancomundial.org/es/topic/water-in-agriculture#1 [accedido en Septiembre de 2021]
3. Sitio web Academia de Riego Kilimo, sin fecha [online] disponible: https://academiaderiego.kilimoagtech.com/huella-hidrica-en-la-produccion-de-cultivos-bajo-riego-todo-lo-que-necesitas-saber [accedido en Marzo de 2021]
4. Sitio web Kilimo, sin fecha [online] disponible: https://www.kilimo.com.arCórdoba, Argentina. Empresa recomendaciones de riego [accedido en Marzo de 2021]
5. Sitio web de INIA, sin fecha [online] disponible: http://www.inia.uy/ [accedido en Diciembre de 2020]
6. Sitio web GRAS, sin fecha [online] disponible: http://www.inia.org.uy/disciplinas/agroclima/gras_pres.htm [accedido en Diciembre de 2020]
7. Sitio web TagoIO, sin fecha [online] disponible: https://tago.io/ [accedido en Junio de 2021]

Capítulo 2:

8. Internet Engineering Task Force (IETF), RFC 8376 “Low-Power Wide Area Network”, Mayo 2018 [online] disponible: https://tools.ietf.org/html/rfc8376 [accedido en Mayo de 2021]
9. Sitio web Signals IoT, “IoT Celular: 85% del mercado en manos de diez operadores”, 24 de Julio de 2020 [online] disponible: https://signalsiot.com/iot-celular-85-del-mercado-en-manos-de-diez-operadores/ [accedido en Junio de 2021]

10. Sitio web Antel, ”Conectividad IoT”, sin fecha [online] disponible: https://www.antel.com.uy/web/iot/conectividad/conectividad-iot [accedido en Junio de 2021]
11. Sitio web Antel, “Cobertura”, sin fecha [online] disponible: https://www.antel.com.uy/personas-y-hogares/movil/cobertura [accedido en Junio de 2021]
12. Sitio web de Sigfox, “What is sigfox?”, sin fecha [online] disponible: https://www.sigfox.com/en [accedido en Noviembre de 2021]
13. Sitio web de DSET Energy, “Tecnología Sigfox: Internet de las cosas”, sin fecha [online] disponible: http://productos-iot.com/sigfox-3/ [accedido en Noviembre de 2021]

LoRa y LoRaWAN:

14. Sitio web Semtech, sin fecha [online] disponible: https://www.semtech.com/lora [accedido en Noviembre de 2020]
15. Sitio web CatSensors, “Tecnología LoRa y LoRaWAN”, sin fecha [online] disponible: https://www.catsensors.com/es/lorawan/tecnologia-lora-y-lorawan [accedido en Noviembre de 2020]
16. Sitio web LoRa Alliance, sin fecha [online] disponible:: https://lora-alliance.org/ [accedido en Abril de 2021]
17. Sitio web TTN, “What are LoRa and LoRaWAN?”, sin fecha [online] disponible: https://www.thethingsnetwork.org/docs/lorawan/ [accedido en Abril de 2021]
18. Sitio web LoRa Alliance “Regional Parameters”, sin fecha [online] disponible: https://lora-alliance.org/resource_hub/rp2-1-0-3-lorawan-regional-parameters/ [accedido en Abril de 2021]
19. Sitio web Wikipedia, “Modelo OSI” sin fecha [online] disponible: https://es.wikipedia.org/wiki/Modelo_OSI [accedido en Marzo de 2021]
20. Sitio web Wikipedia “ALOHA” sin fecha [online] disponible: https://es.wikipedia.org/wiki/ALOHAnet [accedido en Marzo de 2021]

OTAA vs ABP:

Sitio web, ”ABP vs OTA” sin fecha [online] disponible: https://www.thethingsindustries.com/docs/devices/abp-vs-otaa/ [accedido en Agosto de 2021]

21. Sitio web, sin fecha [online] disponible: https://www.thethingsnetwork.org/docs/lorawan/security/ [accedido en Agosto de 2021]
22. Sitio web, “End Device Activation” sin fecha [online] disponible: https://www.thethingsnetwork.org/docs/lorawan/end-device-activation/ [accedido en Agosto de 2021]
23. Web Article, “LoRaWAN - OTA or ABP?” sin fecha [online] disponible: https://static1.squarespace.com/static/560cc2c2e4b01e842d9fac18/t/5a938d38ec212d9451fbecf8/1519619387035/OTAA_or_ABPv3.pdf [accedido en Agosto de 2021]
24. Sitio web Wikipedia, sin fecha [online] disponible: https://es.wikipedia.org/wiki/Ataque_de_REPLAY [accedido en Agosto de 2021]

Placa de desarrollo:

25. Sitio web Arduino, sin fecha [online] disponible: https://store-usa.arduino.cc/products/arduino-mkr-wan-1310 [accedido en Marzo de 2021]

Gateway:

26. Sitio web Dragino, sin fecha [online] disponible: https://www.dragino.com/products/lora-lorawan-gateway/item/148-lps8.html [accedido en Marzo de 2021]

Sensor de humedad del suelo:

27. Sitio web Metergoup, “TDR, FDR and Capacitance Compared” sin fecha [online] disponible: https://www.metergroup.com/environment/articles/tdr-fdr-capacitance-compared/ [accedido en Abril de 2021]

28. Webinar Kilimo “El idioma del suelo”, 21 de septiembre de 2020: https://academiaderiego.kilimoagtech.com/el-idioma-del-suelo-cuando-como-y-cuanto-es-necesario-regar , [participación online en Septiembre de 2020]
29. Sitio web Vegetroix, sin fecha [online] disponible: https://vegetronix.com/Products/VH400/ [accedido en Abril de 2021]
30. Sitio web Vegetroix, curva de calibración, sin fecha [online] disponible: https://vegetronix.com/Products/VH400/VH400-Piecewise-Curve.phtml [accedido en Julio de 2021]

Sensores Digitales:

31. Maxim Integrated DS18B20 Datasheet “DS18B20 Programmable Resolution 1-Wire Digital Thermometer ”,Rev 6; Julio de 2019: https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf [accedido en Julio de 2021]
32. AOSONG Electronics, Datasheet AM2301, “AM2301”: http://www.haoyuelectronics.com/Attachment/AM2301/AM2301.pdf [accedido en Julio de 2021]
33. Sitio web, “DS18B20 sensor de temperatura para líquidos con Arduino” sin fecha [online] disponible: https://programarfacil.com/blog/arduino-blog/ds18b20-sensor-temperatura-arduino/ [accedido en Julio de 2021]
34. Sitio web, “SENSOR DE TEMPERATURA Y HUMEDAD RELATIVA DHT21 (AM2301)” sin fecha [online] disponible: https://naylampmechatronics.com/sensores-temperatura-y-humedad/354-sensor-de-temperatura-y-humedad-relativa-dht21-am2301.html [accedido en Julio de 2021]

Consumo energético:

35. Sitio web Libelium Smart Agricultura, sin fecha [online] disponible, Rev 10; Marzo de 2015: https://www.the-iot-marketplace.com/libelium-smart-agriculture-vertical-kit [accedido en Agosto de 2021]

36. Sitio web librería Arduino, sin fecha [online] disponible: https://www.arduino.cc/en/Reference/LowPowerDeepSleep [accedido en Julio de 2021]
37. Sitio web Microchip, 2021 [online] disponible: https://microchipdeveloper.com/32arm:samd21-pm-overview [accedido en Septiembre de 2021]
38. Hoja de datos RTC DS3231, Maxim Integrated ”Extremely Accurate I2C-Integrated RTC/TCXO/Crystal” : https://datasheets.maximintegrated.com/en/ds/DS3231.pdf [accedido en Agostode 2021]
39. Hoja de datos FAIRCHILD SEMICONDUCTOR Mosfet IRF540N, 2002, “33A, 100V, 0.040 Ohm, N-Channel, Power MOSFET”: https://media.digikey.com/pdf/Data%20Sheets/Fairchild%20PDFs/IRF540N.pdf [accedido en Septiembre de 2021]

Comparación con otras soluciones en el mercado:

40. Sitio web Nettra, “Agricultura de precisión” sin fecha [online] disponible: https://nettra.tech/que-hacemos/agricultura/ [accedido en Octubre de 2021]
41. Sitio web Libelium sin fecha [online] disponible: https://www.libelium.com/ [accedido en Octubre de 2021]
42. Sitio web The IoT marketplace, sin fecha [online] disponible: https://www.the-iot-marketplace.com/libelium-smart-agriculture-vertical-kit [accedido en Octubre de 2021]
43. Sitio web Dragino LSN50, sin fecha [online] disponible: https://www.dragino.com/products/lora-lorawan-end-node/item/155-lsn50-v2.html [accedido en Octubre de 2021]
44. Sitio web Moko, sin fecha [online] disponible: https://www.mokosmart.com/lorawan-based-temperature-humidity-sensor-lw002-th/ [accedido en Octubre de 2021]

The Things Network:

45. Sitio web The Things Network, sin fecha [online] disponible: https://www.thethingsnetwork.org/ [accedido en Abril de 2021]

46. Sitio web The Things Industries, sin fecha [online] disponible: https://www.thethingsindustries.com/docs/getting-started/ttn/ [accedido en Abril de 2021]
47. Sitio web TTN, sin fecha [online] disponible: https://www.thethingsnetwork.org/article/how-spreading-factor-affects-lorawan-device-battery-life [accedido en Agosto de 2021]
48. Sitio web LoRaTools, sin fecha [online] disponible: https://loratools.nl/#/airtime [accedido en Agosto de 2021]

TagoIO:

49. Sitio web TagoIO, sin fecha [online] disponible: https://docs.tago.io/en [accedido en Mayo de 2021]
50. Sitio web Khomp, sin fecha [online] disponible: https://www.khomp.com/es/tagoio-solucion-iot/ [accedido en Mayo de 2021]

Conclusiones:

51. Sitio web, “Garantizar modalidades de consumo y producción sostenibles” sin fecha [online] disponible: https://www.un.org/sustainabledevelopment/es/sustainable-consumption-production/ [accedido en Diciembre de 2021]

Anexos

A- Detalles de los planes de canales dinámicos [18]:

Plan	EU868	CN779	EU433	IN865	KR920	AS923-1	AS923-2	AS923-3	AS923-4	RU864
Default Freq band	863 to 870 MHz	779 to 787 MHz	433 to 434	865 to 867 MHz	920.9 to 923.3 MHz	915 to 928 MHz	915 to 928 MHz	915 to 928 MHz	917 to 920 MHz	864 to 870 MHz
Mandatory Channel Freq (Join Req)	868.10 MHz 868.30 MHz 868.50 MHz	779.5 MHz 779.7 MHz 779.9 MHz	433.175 MHz 433.375 MHz 433.575 MHz	865.0625 MHz 865.4025 MHz 865.985 MHz	922.10 MHz 922.30 MHz 922.50 MHz	923.20 MHz 923.40 MHz	921.4 MHz 921.6 MHz	916.6 MHz 916.8 MHz	917.3 MHz 917.5 MHz	868.9 MHz 869.1 MHz
JoinReq DataRate [MinDR:MaxDR]	[0:5]	[0:5]	[0:5]	[0:5]	[0:5]	[2:5]	[2:5]	[2:5]	[2:5]	[0:5]
CFList Type Supported	0	0	0	0	0	0	0	0	0	0
Mandatory Data Rate [MinDR:MaxDR]	[0:5]	[0:5]	[0:5]	[0:5]	[0:5]	[0:5]	[0:5]	[0:5]	[0:5]	[0:5]
Optional Data Rate [MinDR:MaxDR]	[6:7] | [6:11]	[6:7]	[6:7]	[7]		[6:7]	[6:7]	[6:7]	[6:7]	[6:7]
Number of channels	16	16	16	16	16	16	16	16	16	16
ChMaskCtrl - ChMask	0 -> Channels 0-15 6 -> All channels on	0 -> Channels 0-15 6 -> All channels on	0 -> Channels 0-15 6 -> All channels on	0 -> Channels 0-15 6 -> All channels on	0 -> Channels 0-15 6 -> All channels on	0 -> Channels 0-15 6 -> All channels on	0 -> Channels 0-15 6 -> All channels on	0 -> Channels 0-15 6 -> All channels on	0 -> Channels 0-15 6 -> All channels on	0 -> Channels 0-15 6 -> All channels on
Default channels	[0:2]	[0:2]	[0:2]	[0:2]	[0:2]	[0:1]	[0:1]	[0:1]	[0:1]	[0:1]
Default RX1DRoffset	0	0	0	0	0	0	0	0	0	0
Allowed RX1DRoffset	[0:5]	[0:5]	[0:5]	[0:7]	[0:5]	[0:7]	[0:7]	[0:7]	[0:7]	[0:5]
Duty Cycle	< 1%	< 1%	< 10%		LBT	< 1%	< 1%	< 1%	< 1%	< 1%
Dwell time limitation	No	No	No	No	No	Yes (400ms)	Yes (400ms)	Yes (400ms)	Yes (400ms)	No
TxParamSetupReq support	No	No	No	No	No	Yes	Yes	Yes	Yes	No
Max EIRP (default) - TXPower 0	+16 dBm	+12 dBm	+12dBm	+30 dBm	+14 dBm	+16 dBm	+16 dBm	+16 dBm	+16 dBm	+16 dBm
Default RX2DataRate	DR0	DR0	DR0	DR2	DR0	DR2	DR2	DR2	DR2	DR0
Default RX2 Frequency	869.525 MHz	786.0 MHz	434.665 MHz	866.550 MHz	921.90 MHz	923.2 MHz	921.4 MHz	916.6 MHz	917.3 MHz	869.1 MHz
Class B default Beacon Freq	869.525 MHz	785.0 MHz	434.665 MHz	866.550 MHz	923.1 MHz	923.4 MHz	921.6 MHz	916.8 MHz	917.5 MHz	869.1 MHz
Class B default downlink pingSlot Freq	869.525 MHz	785.0 MHz	434.665 MHz	866.550 MHz	923.1 MHz	923.4 MHz	921.6 MHz	916.8 MHz	917.5 MHz	868.9 MHz

B- Detalles de los planes de canales fijos [18]:

Plan	US915	AU915
Default Freq band	902 to 928 MHz	915 to 928 MHz
Mandatory Channel Freq (Join Req)	upstream: 64 (902.3 to 914.9 [+ by 0.2]) + 8 (903.0 to 914.2 [+ by 1.6]) downstream: 8 (923.3 to 927.5 [+ by 0.6])	upstream: 64 (915.2 to 927.8 [+ by 0.2]) + 8 (915.9 to 927.1 [+ by 1.6]) downstream: 8 (923.3 to 927.5 [+ by 0.6])
JoinReq DataRate [MinDR:MaxDR]	64 (125 kHz channels) using DR0 and 8 (500 kHz channels) using DR4	64 (125 kHz channels) using DR2 and 8 (500 kHz channels) using DR6
CFList Type Supported	1	1
Mandatory Data Rate [MinDR:MaxDR]	[0:4],[8:13]	[0:6],[8:13]
Optional Data Rate [MinDR:MaxDR]	[5:6]	[7]
Number of channels	upstream: 64 (125 kHz) + 8 (500 kHz) downstream: 8 (500 kHz)	upstream: 64 (125 kHz) + 8 (500 kHz) downstream: 8 (500 kHz)
ChMaskCtrl - ChMask	0 -> Channels 0 to 15 1 -> Channels 16 to 31 ... 4 -> Channels 64 to 71 5 -> 8LSBs controls Channel Blocks 0 to 7, 8MSBs are RFU 6 -> All 125 kHz ON, ChMask applies to channels 64 to 71 7 -> All 125 kHz OFF, ChMask applies to channels 64 to 71	0 -> Channels 0 to 15 1 -> Channels 16 to 31 ... 4 -> Channels 64 to 71 5 -> 8LSBs controls Channel Blocks 0 to 7, 8MSBs are RFU 6 -> All 125 kHz ON, ChMask applies to channels 64 to 71 7 -> All 125 kHz OFF, ChMask applies to channels 64 to 71
Default channels	[0:71]	[0:71]
Default RX1DRoffset	0	0
Allowed RX1DRoffset	[0:3]	[0:5]
Duty Cycle	No Limit	No Limit
Dwell time limitation	[0:63] 400ms [64:71] No	[0:63] 400ms (regional dependence) [64:71] No
TxParamSetupReq support	No	Yes
Max EIRP (default) - TXPower 0	+30 dBm	+30 dBm
Default RX2DataRate	DR8	DR8
Default RX2 Frequency	923.3 MHz	923.3 MHz
Class B default Beacon Freq	Hops across all 8 downlink channels	Hops across all 8 downlink channels
Class B default downlink pingSlot Freq	Follows beacon channel	Follows beacon channel

C- *Especificaciones de Arduino MKR WAN 1310 [25]:*

MICROCONTROLLER	SAMD21 Cortex®-M0+ 32bit low power ARM MCU (datasheet)
RADIO MODULE	CMWX1ZZABZ (datasheet)
BOARD POWER SUPPLY (USB/VIN)	5V
SECURE ELEMENT	ATECC508 (datasheet)
SUPPORTED BATTERIES	rechargeable Li-Ion, or Li-Po, 1024 mAh minimum capacity
CIRCUIT OPERATING VOLTAGE	3.3V
DIGITAL I/O PINS	8
PWM PINS	13 (0 .. 8, 10, 12, 18 / A3, 19 / A4)
UART	1
SPI	1
I2C	1
ANALOG INPUT PINS	7 (ADC 8/10/12 bit)
ANALOG OUTPUT PINS	1 (DAC 10 bit)
EXTERNAL INTERRUPTS	8 (0, 1, 4, 5, 6, 7, 8, 16 / A1, 17 / A2)
DC CURRENT PER I/O PIN	7 mA

CPU FLASH MEMORY	256 KB (internal)
QSPI FLASH MEMORY	2MByte (external)
SRAM	32 KB
EEPROM	no
CLOCK SPEED	32.768 kHz (RTC), 48 MHz
LED_BUILTIN	6
USB	Full-Speed USB Device and embedded Host
ANTENNA GAIN	2dB (bundled pentaband antenna)
CARRIER FREQUENCY	433/868/915 MHz
WORKING REGION	EU/US (confirmed) other countries check your region's spectrum availability
LENGTH	67.64 mm
WIDTH	25 mm
WEIGHT	32 gr.

D- Especificaciones del Gateway LoRaWAN Dragino LPS8 [26]:

Specifications

Hardware System:

Linux Part:

- 400Mhz ar9331 processor
- 64MB RAM
- 16MB Flash

Interface:

- 10M/100M RJ45 Ports x 1
- WiFi : 802.11 b/g/n
- LoRaWAN Wireless
- Power Input: 5V DC, 2A, Type C
- USB 2.0 host connector x 1

WiFi Spec:

- IEEE 802.11 b/g/n
- Frequency Band: 2.4 ~ 2.462GHz
- Tx power:
  - ✓ 11n tx power : mcs7/15: 11db mcs0 : 17db
  - ✓ 11b tx power: 18db
  - ✓ 11g 54M tx power: 12db
  - ✓ 11g 6M tx power: 18db
- Wifi Sensitivity
  - ✓ 11g 54M : -71dbm
  - ✓ 11n 20M : -67dbm

LoRa Spec:

- Up to -140 dBm sensitivity with SX1257 Tx/Rx front-end
- 70 dB CW interferer rejection at 1 MHz offset
- Able to operate with negative SNR, CCR up to 9dB
- Emulates 49 x LoRa demodulators and 1 x (G)FSK demodulator
- Dual digital TX & RX radio front-end interfaces
- 10 programmable parallel demodulation paths
- Dynamic data-rate (DDR) adaptation
- True antenna diversity or simultaneous dual-band operation

Max Power: 5v, 1.3A

E- Código desarrollado para codificación del payload en el firmware de Arduino:

```
byte payload[20];
    byte *bVWC= (byte*)&VWC;
    byte *bSoilTemperature= (byte*)&SoilTemperature;
    byte *bAirHumidity= (byte*)&AirHumidity;
    byte *bAirTemperature= (byte*)&AirTemperature;
    byte *bBatVoltage= (byte*)&BatVoltage;

    for (int i=0;i<=4;i++){
      payload[i]=bVWC[i];
      payload[i+4]=bSoilTemperature[i];
      payload[i+8]=bAirHumidity[i];
      payload[i+12]=bAirTemperature[i];
      payload[i+16]=bBatVoltage[i];
    }
```

F- Código desarrollado para decodificación del payload en TTN:

```
function decodeUplink(input) {
  var bytes=input.bytes

  function bytesToFloat(bytes) {
    //LSB Format (least significant byte first).
    var bits = bytes[3]<<24 | bytes[2]<<16 | bytes[1]<<8 | bytes[0];
    var sign = (bits>>>31 === 0) ? 1.0 : -1.0;
    var e = bits>>>23 & 0xff;
    var m = (e === 0) ? (bits & 0x7fffff)<<1 : (bits & 0x7fffff) | 0x800000;
    var f = sign * m * Math.pow(2, e - 150);
    f = f.toFixed(2);
    f=parseFloat(f);

    return f;
  }

  return {
    data: {
      humedad_suelo: bytesToFloat(bytes.slice(0, 4)),
      temperatura_suelo: bytesToFloat(bytes.slice(4, 8)),
      humedad_aire: bytesToFloat(bytes.slice(8, 12)),
      temperatura_aire: bytesToFloat(bytes.slice(12, 16)),
      voltaje_bateria: bytesToFloat(bytes.slice(16, 20))
    },
    warnings: [],
    errors: []
  };
}
```

G - Planilla dinámica para estimación de autonomía del nodo

Frecuencia de medida	30	min			
Corriente sleep	0.0000210	A	1800	0.50	horas
Medida/envío	0.030	A	8	0.0022	horas
Carga batería	3	Ah			
En un cíclo	0.00008	Ah	77.17		
En una hora	0.00015	Ah	154.33		
Duracion en horas	19438.44	h			
Duracion dias	809.94	d			
Duracion años	2.22	a			

H- Parámetros para la activación ABP seteados en el archivo de comunicación LoRa:

```
String devAddr = "26021BD1";

String nwkSKey = "DF924B0388E3C67D91D6500EFAC41DC7";

String appSKey = "8248AFC9B810DB75613513D4C99EC437";
```