Sensores de Calidad de Aire
Taller de Sistemas Ciber Físicos 2021
Facultad de Ingeniería

Joaquín Veirana - joaquin.veirana@gmail.com

Docentes:
Matías Richart - mrichart@fing.edu.uy
Eduardo Grampín - grampin@fing.edu.uy

20 de Noviembre de 2021

Índice

1. Propuesta inicial

El proyecto consistió en el desarrollo de un prototipo capaz de realizar mediciones de calidad de aire con el fin de poder llevar a cabo un control de los niveles de CO2 en el aire al volver a las actividades presenciales en la facultad. El prototipo luego de realizar la medición debe ser capaz de transmitir esos datos mediante el uso de tecnologías inalámbricas. Sumado a esto el proyecto también involucró el desarrollo de una aplicación web capaz recibir los datos sensados y almacenarlos en una base de datos y brindar una interfaz gráfica para poder visualizar estos datos por medio de gráficas.

Era de principal interés poder poner a prueba las distintas tecnologías y hardware disponibles de IoT para poder desempeñar la función.

2. Introducción a tecnologías y herramientas utilizadas

En esta sección se brinda una breve introducción a tecnologías usadas que es conveniente conocer a nivel básico para entender de manera completa el proyecto. Para ir más a fondo en estas tecnologías se dejan referencias que explican en mayor detalle cada punto.

2.1. IoT Internet de las cosas

El internet de las cosas [7] es un paradigma tecnológico en el cual se busca interconectar distintos dispositivos físicos cotidianos a internet. El internet de las cosas tiene gran cantidad de campos de aplicación, siendo de los más populares la domótica en el hogar, la salud, la agricultura o la logística.

En este proyecto se busca realizar mediciones del mundo físico siguiendo un modelo arquitectónico de varios nodos sensores independientes los cuales están constantemente midiendo y enviando información a un nodo central (usualmente llamado gateway) el cual tiene como fin recibir todos estos datos para su posterior procesamiento y/o almacenamiento.

2.2. Lora y LoRaWAN

LoRa [8] (abreviatura de *Long Range*) es una tecnología de transmisión de datos inalámbrica como lo es WiFi pero concebida para el uso en aplicaciones de internet de las cosas. Dentro de las ventajas que ofrece encontramos la posibilidad de transmitir datos a grandes distancias (llegando a kilómetros en condiciones ideales), robustez frente a ruido o interferencias y bajo consumo eléctrico. Todas estas ventajas que ofrece LoRa vienen a cambio del bajo ancho de banda, lo que usualmente en aplicaciones de IoT no es un problema ya que los datos que se suelen enviar consisten en unos pocos bytes de información.

LoraWAN es un protocolo que trabaja con la tecnología LoRa, brinda un estándar de red para las conexiones en dicha tecnología.

2.3. PlatformIO

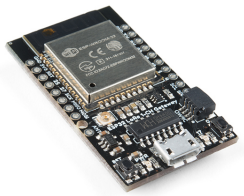
PlatformIO [14] es una herramienta multiplataforma utilizada para programar sistemas embebidos de una manera rápida y sencilla. PlatfomIO puede ser instalado como un complemento en VSCode junto con todas las dependencias y bibliotecas necesarias para programar varias familias de dispositivos, entre ellos los dispositivos esp32 usados en este proyecto y descritos

en detalle más adelante. Sumado a esto, PlatformIO permite agregar bibliotecas externas de terceros por medio de su interfaz gráfica y grabar en los dispositivos el código creado, solamente teniendo configurar un archivo principal de configuración *platformio.ini* en donde se especifica el modelo de la placa a utilizar, el framework de trabajo, el puerto en el que está conectado el dispositivo, los nombres de las bibliotecas externas, entre otros. En la sección *Documentación sobre procedimientos*, más precisamente en la subsección Set-up y programación del dispositivo Sparkfun ESP32 LoRa 1-Channel Gateway con Platform IO se explica más en detalle el uso de la herramienta.

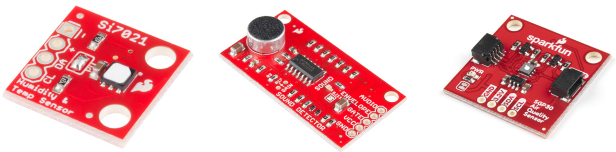
3. Solución realizada

La solución realizada para el proyecto consiste de un dispositivo ESP32 LoRa 1-Channel Gateway [1] del fabricante Sparkfun. Este dispositivo cuenta de fábrica con una antena LoRa modelo RFM95W, antena WiFi y Bluetooth, así como con pines de entrada/salida digital y analógica en los cuales se le conectaron tres sensores de medición también del fabricante Sparkfun, estos sensores se listan a continuación.

- Sensor de temperatura y humedad Si7021 [2]
- Sensor de detección de sonido [3]
- Sensor de calidad de aire (CO_2 y TVOC) SGP30 [4]



ESP32 LoRa 1-Channel Gateway



De izquierda a derecha: Sensor Si7021, Sensor de sonido, Sensor SGP30

El dispositivo ESP32 LoRa 1-Channel Gateway (al cual llamaremos esp32 de ahora en adelante) contiene como su nombre indica, un microcontrolador programable esp32-WROOM-32. Este microcontrolador cuenta con un procesador de dos núcleos y un coprocesador de baja potencia además de los módulos wifi, bluetooth, lora y entrada/salida nombrados anteriormente. Para información detallada del hardware siempre es bueno referirse al datasheet official [5]. Cabe aclarar que el dispositivo no implementa mediante hardware los protocolos LoraWAN necesarios para el envío de información por medio de esta tecnología, por lo que es necesario el uso de una

biblioteca de software específica para la implementación de estos protocolos, la biblioteca utilizada para esto es LMIC [6]. Para el manejo de procesos en el esp32 y para el manejo de asignación de recursos de procesador se utilizó un sistema operativo de tiempo real llamado freeRTOS [9] el cual se encarga del scheduling de tareas (funciones) definidas por el programador, en el caso del proyecto estas tareas consisten en funciones de medición y de envío de datos y de procesamiento de comandos como se verá más en detalle en la siguiente sección.

Para el almacenamiento final de los datos se utilizó Cloudant, una base de datos noSQL en la nube [10] que tiene bibliotecas para ser usada en varios lenguajes de programación [11]. Cloudant funciona bajo el paradigma de documentos por lo que los datos finales se podrán hallar en formato JSON.

Para el envío y procesamiento de datos antes de su almacenamiento en Cloudant se probaron distintas plataformas, la primera fue The Things Network (TTN) [12] y la segunda Orbiwise/-Yeap [13]. Estas plataformas se encuentran disponibles a través de internet y funcionan como intermediarias reenviando los datos que recibe desde nuestro gateway LoRa hasta nuestra aplicación/base de datos. En particular para el envío de datos a la red TTN se dispuso de un gateway del mismo fabricante y para el envío a la red de Orbiwise/Yeap se utilizó una antena LoRa de la empresa Yeap ubicada en la Facultad de Ingeniería. Ambas plataformas y gateways son vistas más en detalle en las próximas secciones.

4. Iteraciones realizadas

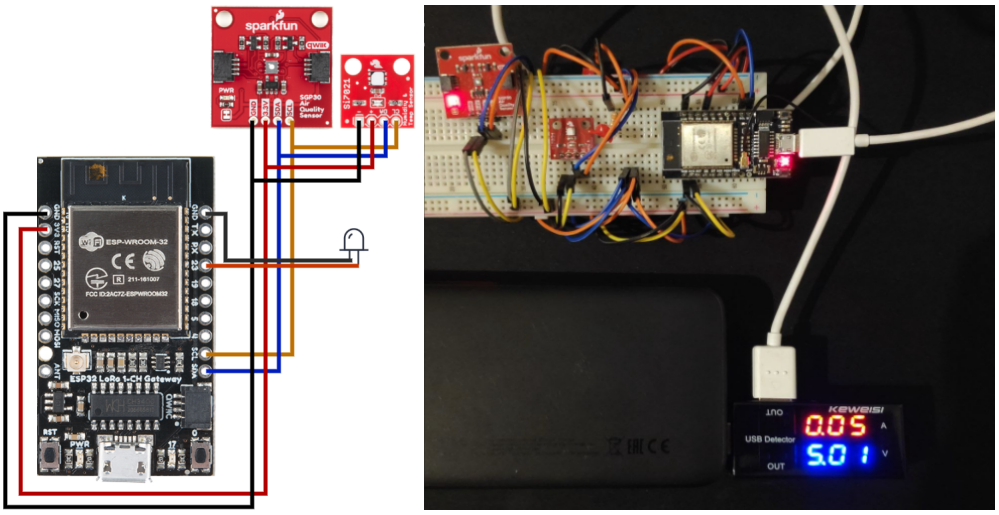
En esta sección se dará una descripción de cada una de las soluciones en las que se iteró a lo largo del proyecto, dando para cada una de ellas una descripción general, la arquitectura de alto nivel así como algunos detalles precisos sobre la implementación.

4.1. Primer Prototipo

El primer prototipo funcional realizado consistió en el dispositivo esp32 conectado a los sensores de clima y de calidad de aire, sin el sensor de sonido. El software grabado en este dispositivo no dispone de freeRTOS por lo que solo contiene un programa principal en loop en el que se miden y envían datos, luego se bloquea el proceso por una cierta cantidad de segundos para posteriormente repetir el ciclo.

El objetivo principal de este prototipo era el de tener un dispositivo capaz de realizar mediciones y almacenarlas para su posterior análisis y comparación con otras mediciones realizadas con otros sensores en la Facultad. Para ver las pruebas realizadas que demuestran el funcionamiento final del dispositivo así como las comparaciones entre el sensor SGP30 y los demás sensores, referirse a la sección de *Pruebas realizadas y resultados obtenidos*.

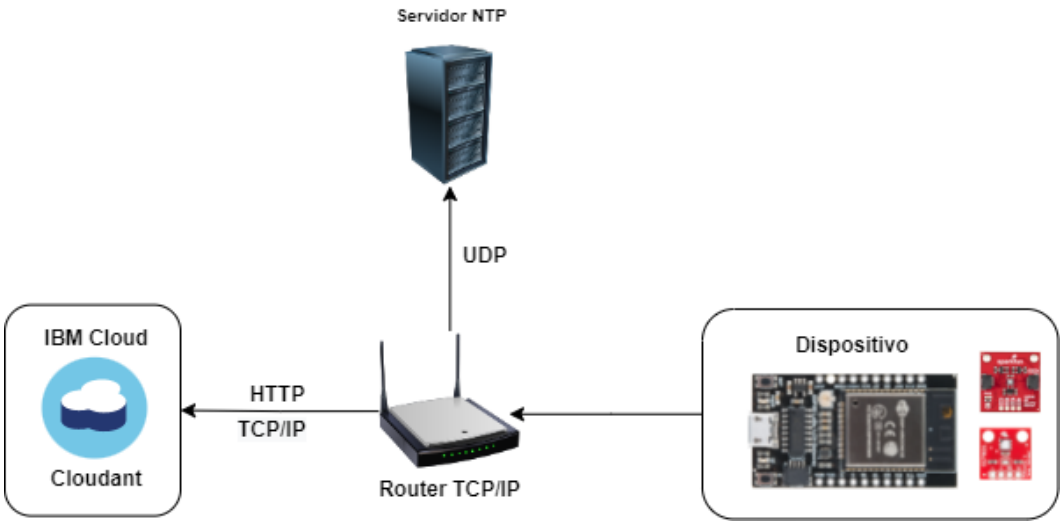
El esquema de conexiones en el dispositivo fue el siguiente:



Esquema de conexiones y foto del protitpo real

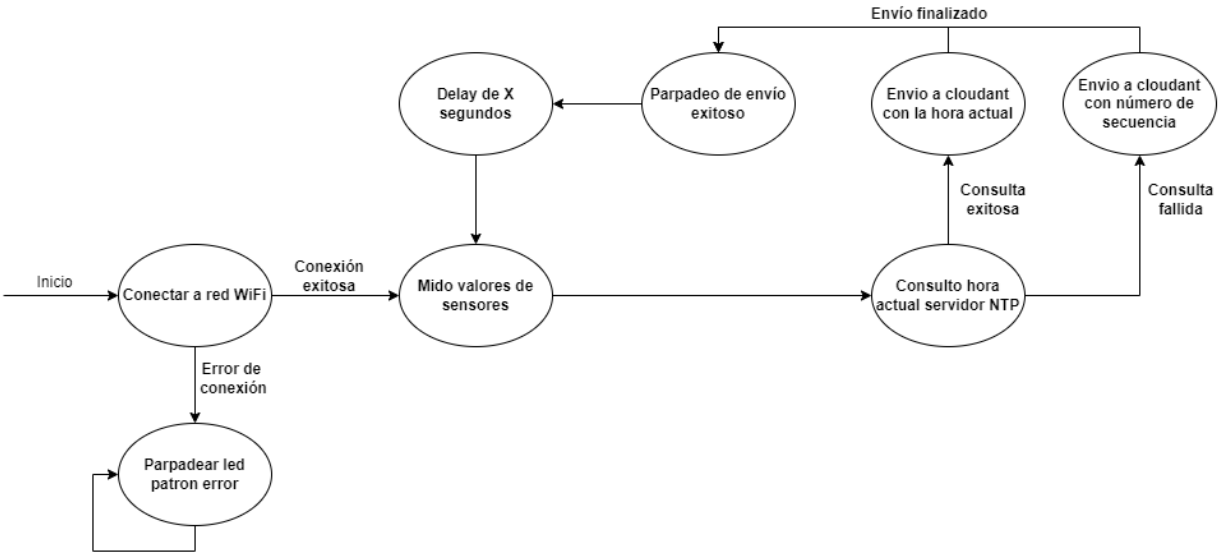
4.1.1. Arquitectura de la solución

La arquitectura de la solución es la siguiente:



Arquitectura del prototipo 1

Este primer prototipo no hace uso de LoRa, realiza los envíos directo a Cloudant por medio de WiFi usando el protocolo HTTP. Como se comentó anteriormente, la lógica de la aplicación es muy sencilla y consiste en un loop infinito de envíos separados por un delay de espera. Para determinar la hora que se le debe asociar a la medición, el dispositivo consulta un servidor Network Time Protocol (NTP) antes de enviar cada medición a la base. Se adjunta una máquina de estados con el comportamiento simplificado del prototipo.



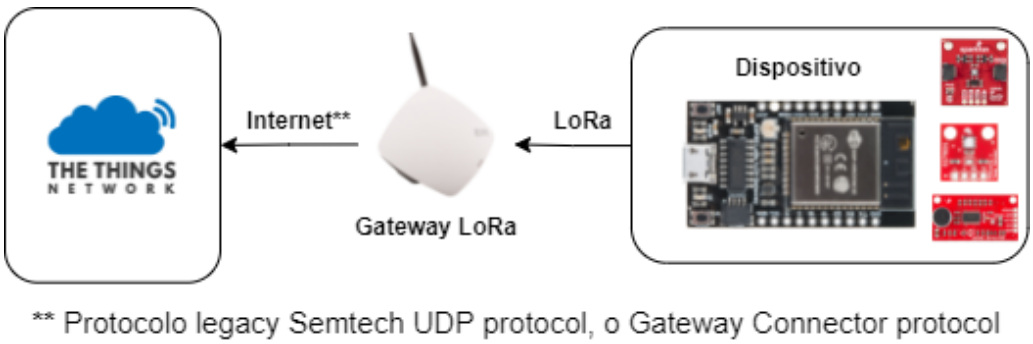
Comportamiento del prototipo 1

Para visualizar el código utilizado de esta primera versión, referirse al proyecto ubicado en la carpeta *Otros/Envio por WiFi*, en este proyecto PlatformIO al igual que en todos los realizados hay un archivo llamado *GlobalData.h* ubicado en la carpeta *Include* el cual contiene las variables de entorno necesarias como por ejemplo el nombre y clave de la red WiFi así como el endpoint y apikey de la base de datos.

4.2. Segundo Prototipo

El segundo prototipo agrega sobre el primero las funcionalidades de envío por LoRa y la conexión con el sensor de sonido, el objetivo principal del prototipo era lograr la conexión LoRa por lo que no se realizó la conexión con Cloudant ni tampoco dispone de freeRTOS. Para el envío de datos por LoRa se realizó una integración con la plataforma The Things Network (TTN).

4.2.1. Arquitectura de la solución



Arquitectura del prototipo 2

Por otra parte, para el envío de datos se utilizaron dos métodos distintos de activación de dispositivos existentes en LoRa, estos métodos son ABP (Activation by personalization) y OTAA (Over the air activation). Al conectarse exitosamente a nuestra red, el dispositivo es asignado con una tira de 32-bit llamada *DevAddr* que identifica al nodo en la red LoRa, este

DevAddr está formado a su vez por la concatenación de dos tiras llamadas NwkAddr (network address) y NwkID (network id). En ABP el dispositivo ya contiene hardcodeado el *DevAddr* así como las keys de sesión, siendo capaz de conectarse de forma directa a la red sin la necesidad de negociar claves. Por un lado este método tiene una activación más sencilla pero tiene limitaciones, como por ejemplo el no poder conectar un dispositivo a distintas redes a la vez y una gran vulnerabilidad en seguridad por el hecho de tener las claves guardadas en el dispositivo. OTAA por otro lado realiza un handshake entre el gateway y el dispositivo sensor al momento de establecer una conexión en donde se intercambian las distintas claves de sesión, haciéndolo un método más seguro. En general no hay motivo para utilizar ABP sobre OTAA, pero OTAA requiere que el dispositivo esté dentro del rango de la red para poder conectarse puesto que necesita poder recibir mensajes de ack (Acknowledgment) desde el gateway.

Para visualizar el código utilizado de esta segunda versión, referirse al proyecto ubicado en la carpeta *Otros/Envio por ABP* u *Otros/Envio por OTAA*.

Para seguir una guía de cómo se llevó a cabo este prototipo referirse a las subsecciones de *Set-up y conexión del gateway a TTN V3* y *Set-up y conexión de un dispositivo sensor a TTN V3* dentro de la sección Documentación de procedimientos.

4.3. Tercer Prototipo (Final)

La última iteración de trabajo realizada añade la posibilidad de enviar datos tanto por WiFi como por LoRa, pudiendo cambiar entre estos de forma dinámica bajo ciertas condiciones. Para llevar a cabo esta funcionalidad se implementaron dos tasks específicas (procesos en FreeRTOS). Además de estas dos tasks de envío de datos (WiFi y LoRa) se implementó una task adicional que funciona como una interfaz de comandos, esta función es capaz de escuchar comandos escritos por el usuario en el puerto serial de la computadora y realizar las respectivas acciones que estos comandos tengan programadas. Esta funcionalidad permite potencialmente realizar configuraciones o cambios sobre el dispositivo sin la necesidad de volver a realizar un flasheo de la placa. Si bien solo se implementaron funciones que cambian parámetros de configuración o realizan pequeños cambios en la lógica del dispositivo, al estar estas funciones creadas manualmente es posible agregar todos los comandos que se crean necesarios, pudiendo hacer cualquier tipo de funciones en estos, más adelante se deja una breve guía para agregar nuevos comandos a la task.

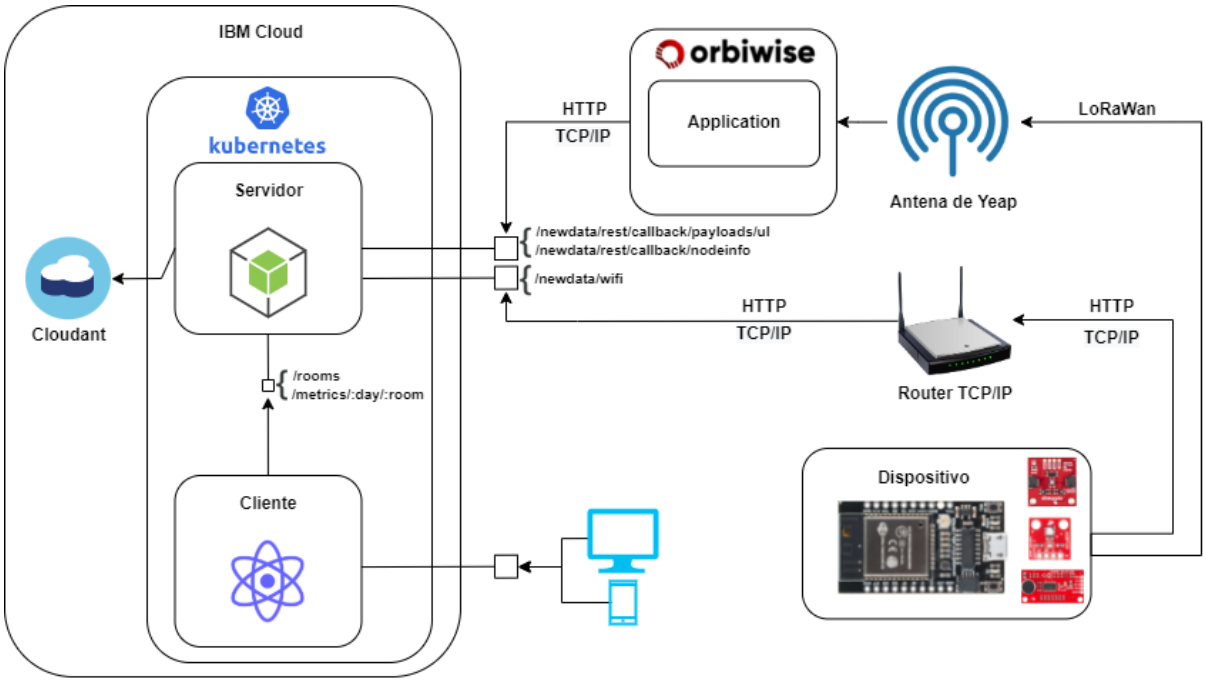
Por otra parte se realizaron las pruebas necesarias para asegurar que el dispositivo pudiera enviar datos a la red Yeap/Orbiwise de manera correcta por medio de la antena presente en la Facultad.

Se implementó además una aplicación web conformada por un cliente (React y Nginx) que permite la visualización de los datos almacenados por medio de gráficas y un servidor (NodeJS) que brinda endpoints, por un lado para que la aplicación cliente consulte datos por medio de queries así como para recibir datos desde Yeap o desde WiFi, procesarlos y enviarlos a Cloudant. En el caso de Yeap la aplicación servidor también tiene un endpoint para recibir metadatos. Estas aplicaciones fueron desplegadas en un cluster de Kubernetes gratuito disponible en IBM Cloud.

Para poder realizar el reenvío de los datos desde la red de Yeap a la aplicación en Kubernetes fue necesario crear endpoints especiales en la aplicación servidor y realizar algunas configuraciones en la plataforma Orbiwise/Yeap. Los detalles en profundidad de cómo fue llevado a cabo este proceso y de cómo son los endpoints de la aplicación se encuentran dentro de la sección de Documentación de procedimientos, más precisamente en la subsección *Set-up y conexión de una aplicación y dispositivo en Yeap*. Por otra parte en la subsección *Despliegue de la aplicación web en Kubernetes en IBM Cloud* se puede encontrar un manual de despliegue de la aplicación web en caso de querer replicar la solución. Por último, en la subsección *Set-up y configuración*

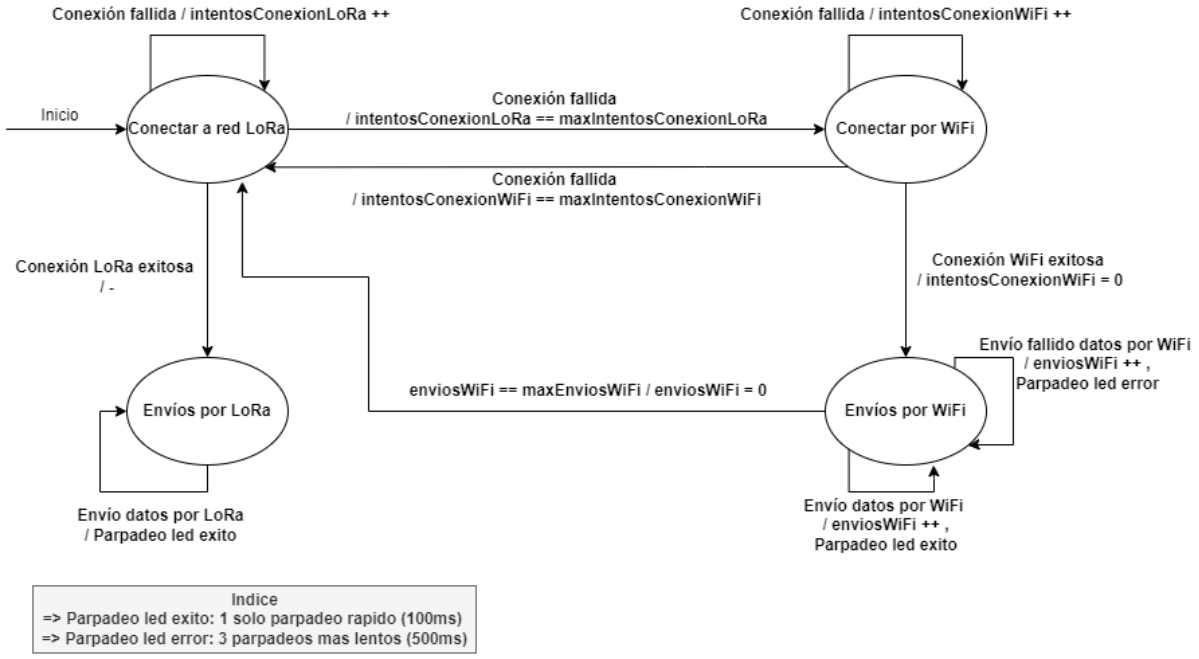
del dispositivo sensor con el código del prototipo final se halla un manual de configuración del dispositivo sensor con el código desarrollado en este tercer prototipo, en esa guía se explica en detalle todos los parámetros de configuración que se encuentran disponibles.

4.3.1. Arquitectura de la solución



Arquitectura del prototipo 3

Como se comentó en la descripción anterior, el dispositivo hace uso de dos tasks de FreeRTOS para el envío de datos, una realiza envíos por WiFi y la otra por LoRa. La lógica utilizada por el programa para decidir cuál de los dos usar se modela con la siguiente máquina de estados.



Comportamiento del prototipo 3

Se puede apreciar en la máquina de estados que el dispositivo le da prioridad al envío por LoRa, es por esto que comienza intentando conectarse a la red de LoRa inicialmente. Si la conexión es exitosa dentro del número de intentos predefinido en una variable de configuración, el dispositivo considera que la conexión con LoRa es exitosa y se quedará enviando datos por este medio indefinidamente. Cabe aclarar que si en algún momento el dispositivo se moviera sacándolo del rango de la antena o si por algún factor externo los mensajes dejasen de llegar al gateway, el dispositivo continuará realizando envíos y no sabrá que se perdió la conexión. Esto sucede por el hecho de no recibir mensajes de acknowledgment (acks) desde el gateway.

Por otra parte si el intento de conexión inicial falla todas las veces definida en la configuración, procederá a intentar conectarse con la red WiFi que tenga guardada en su archivo de configuración. Nuevamente intentará conectarse y de lograrlo, realizará una cierta cantidad de envíos por este medio hasta llegar a la cantidad máxima de envíos por WiFi, también definida en el archivo de configuración. Luego de realizar esta cantidad de envíos el dispositivo le volverá a dar una chance a LoRa de crear una conexión, repitiendo el ciclo anteriormente mencionado.

Si por otro lado la conexión por WiFi también falla en todos los intentos, el dispositivo volverá a intentar conectarse por LoRa sin realizar envíos por WiFi. Se puede dar el escenario en el que el dispositivo intercambie para siempre entre intentos de conexión de WiFi y LoRa si en ninguno de los dos casos logra una conexión exitosa.

4.4. Task de Interfaz de comandos

La task asignada a la interfaz de comandos ejecuta sobre el núcleo 0 a diferencia de las task de envíos. Esta función lee tiras desde el monitor serial y las parsea para ejecutar la función correspondiente.

La función encargada de esto es *leerComandosSerial* y está en el archivo *main.cpp*, cada comando puede recibir un parámetro o ninguno. Para agregar un nuevo comando a la función es necesario editar (agregar cosas) a las siguientes variables en *main.cpp*:

- *std::vector<std::string>INSTRUCCIONES*: en esta lista están los nombres principales de

cada comando, estos nombres deben ser de exactamente 7 caracteres!

- *std::vector<std::string>INSTRUCCIONES_PARAMS*: en esta lista se pone una descripción de la forma o los valores que puede tomar el parámetro del comando
- *std::vector<std::string>INSTRUCCIONES_DESC*: en esta lista se pone una descripción de que hace el comando
- *const int CANT_INSTR*: entero que define la cantidad de comandos existentes (es el largo de las listas anteriores)

Luego dentro de la función *evaluarComando* se debe agregar un condicional que haga un chequeo de igualdad entre de la variable recibida del monitor y el nuevo comando (en la lista *INSTRUCCIONES*) y dentro de este condicional la lógica correspondiente.

4.5. Estructura de los documentos en la base de datos

En esta sección se explica la estructura de los documentos en la base de datos y las formas en que los datos son enviados desde el dispositivo a la aplicación servidor que los procesa.

En el servicio de Cloudant se hallan tres bases de datos con los nombres:

- **datos-sensados**: en esta base de datos se hallan los documentos que contienen los datos medidos de los sensores, el identificador de cada unos de estos documentos consiste en un string de la fecha y hora de la medición en formato ISO (ej.: 2021-09-05T01:29:43). Este formato de id permite realizar queries indexadas de rangos temporales más fácilmente ya que de esta forma se ordenan las fechas también lexicográficamente. Además de los datos relacionados a las mediciones, hay un campo numérico *dev_id* que funciona como identificador del dispositivo que envía esos datos y un campo numérico *alerta* el cual no fue utilizado pero tenía como objetivo indicar si el envío de datos se dio de manera rutinaria o a causa de una medición de alerta.
- **metadata**: en esta base de datos se hallan los documentos de metadatos que genera la plataforma Yeap al recibir transimisones, entre estos metadatos se encuentran variables físicas de LoRa como el Signal-to-Noise Ratio (SNR), Received Signal Strength Indication (RSSI), frecuencia y spread factor utilizado en la transmisión, ubicación de la antena, entre otros.
- **salones**: en esta base se hallan los documentos que buscan asociar un salón a un dispositivo o un conjunto de dispositivos. Estos documentos tienen un número identificador que se mapea al campo dev_id de los documentos en la base *datos-sensados*.

A continuación se muestran imágenes de estas bases de datos y ejemplo de sus documentos:

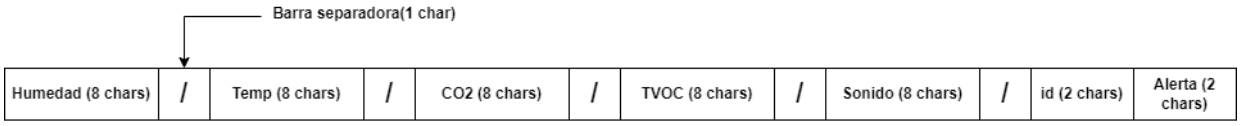
Your Databases

Name	Size	# of Docs
datos-sensados	5.4 MB	81802
metadata	0 bytes	0
salones	91 bytes	1

Bases de datos usadas

Documentos en la base datos-sensados

Los datos enviados por medio de WiFi viajan dentro del body del paquete HTML en formato JSON. Los datos enviados por medio de LoRa van como un arreglo de caracteres de largo 50 con el siguiente formato:

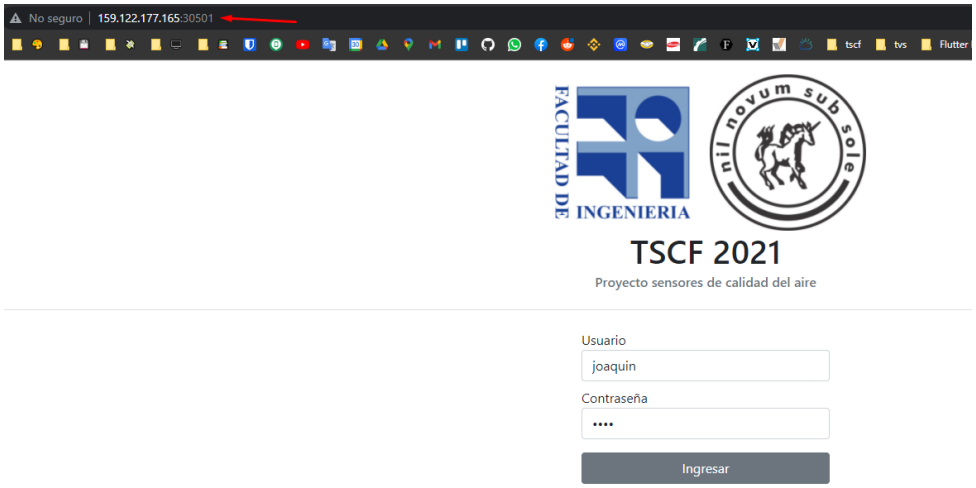


Formato de los datos enviados por LoRa

Es por esto que la aplicación servidor se debe encargar de parsear estos datos y crear el JSON para que pueda ser enviado a Cloudant.

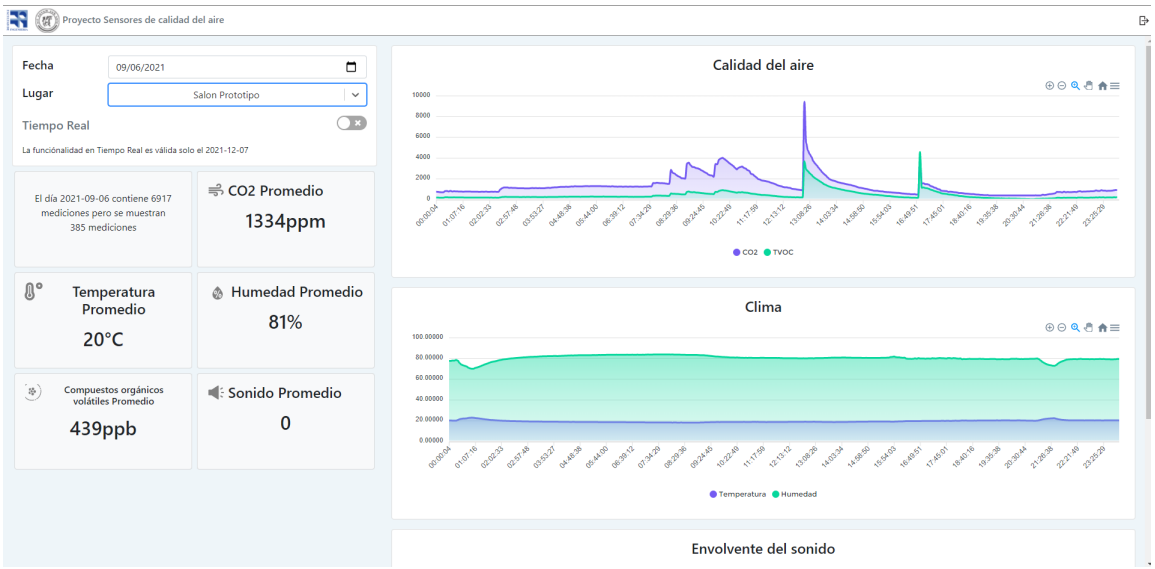
4.6. Aplicación web desarrollada

La aplicacion web tiene una interfaz sencilla donde se tiene una pantalla de login sin logica (el usuario y clave estan hardcodeados). Para ingresar lo hacemos con el usuario **joaquin** y la clave **tscf**

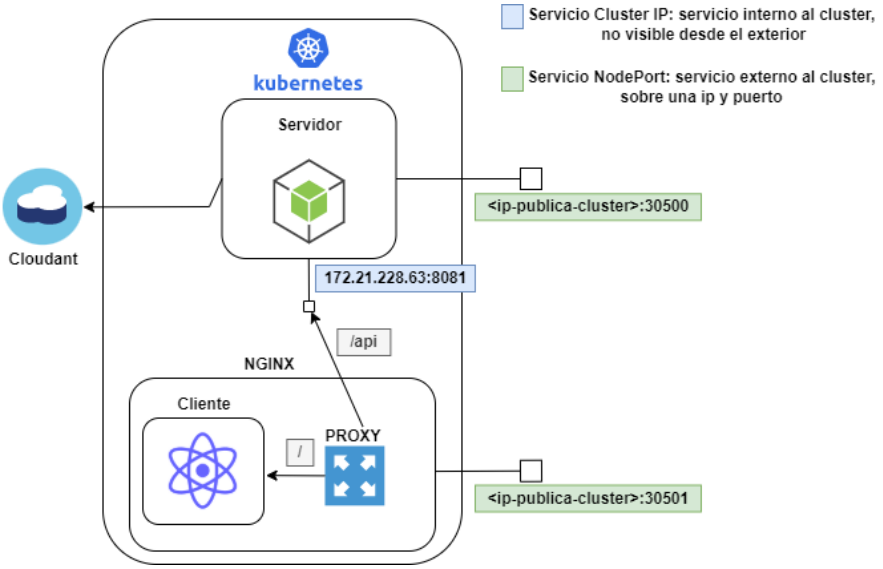


Aplicación web logueo

Una vez dentro se pueden visualizar dos inputs, uno para seleccionar el día y el otro el salón, recordar que los salones son los documentos que se encuentran en la base de datos *salones* donde están los documentos que relacionan mediante el id de dispositivo, los documentos de mediciones que halla en la base *datos-sensados*.



Se presenta la arquitectura de la aplicación a nivel de red y conectividad (IPs y puertos) dentro del cluster:



Se exponen dos servicios externos de tipo NodePort, uno es para el servidor y es usado para recibir los datos por WiFi y el otro es para obtener las páginas del cliente. El cliente es servido por un servidor web Nginx que además se encarga de redirigir los pedidos para el backend. Estos pedidos al backend se redireccionan hacia un endpoint que expone un servicio de tipo Cluster IP, el cual es interno al cluster y no puede accederse desde el exterior. Para ver en detalle estos valores se recomienda ver los archivos de despliegue *.yaml* del cliente y del servidor junto con el archivo de configuración *default.conf* de Nginx.

5. Funcionamiento de FreeRTOS en el dispositivo sensor

En esta sección se busca explicar más en detalle las funciones usadas de la API de FreeRTOS para el manejo de tasks en el procesador del dispositivo. Antes de la explicación detallada de la solución se recomienda referirse a la documentación oficial de la API de FreeRTOS [22] para ver las firmas de las funciones disponibles y la descripción de sus comportamientos de manera precisa. Sumado a esto, se recomienda la visualización de una lista de videos del canal Digi-Key [21] en Youtube el cual explica el funcionamiento general de FreeRTOS en aspectos como el scheduling de tareas, el manejo de memoria, entre otros.

La lógica de la aplicación destinada a FreeRTOS esta toda en el archivo *src/main.cpp*. En este archivo es posible ver las directivas *include "freertos/FreeRTOS.h"* e *include "freertos/tasks.h"* necesarias para tener acceso a los tipos de datos y funciones de la biblioteca. Un poco más abajo, en la parte de definición de variables globales son definidas tres variables de tipo *TaskHandle_t*, estas son usadas para referenciar cada task que defina el programador. Las tasks son procesos dentro del sistema operativo, tienen asociadas un código (una función), parámetros, una cantidad máxima de memoria en el stack asignada por el programador, una prioridad de ejecución y un núcleo asignado en el que ejecutar.

La definición de estas tasks se hacen en el setup() con la función *xTaskCreatePinnedToCore* de la siguiente forma:

```
void setup() {
  static TaskHandle_t handlerTaskEnvioLora = NULL;

  ...

  xTaskCreatePinnedToCore(
      envioLoraOtaa,         // Funcion a llamar
      "EnvioLoraOTAA",       // Nombre de la task
      4096,                  // Tamaño del stack en bytes
      NULL,                  // Parámetros
      prioridadAlta,         // Prioridad
      &handlerTaskEnvioLora, // Handler (TaskHandle_t)
      1);                    // Núcleo en el cual ejecutar (0 o 1)
}
```

Teniendo en cuenta que las tasks de envío de datos por LoRa y por WiFi tienen asociados el mismo núcleo (núcleo 1), fue necesario configurar de qué forma el scheduler iba a asignar tiempo de procesamiento a cada uno.

Inicialmente se quería manejar el scheduling de las tasks por medio de la modificación de sus prioridades con la función *vTaskPrioritySet(handlerTask, nuevaPrioridad)*, seteándole prioridad alta al task de LoRa cuando fuese el turno de enviar por este medio y seteándole prioridad baja al task de WiFi y viceversa. Esto no funcionó ya que ambas funciones de envíos tienen bastantes delays entre medio en donde el proceso queda bloqueado, al ocurrir esto FreeRTOS busca automáticamente otro task asociado a ese núcleo queriendo ejecutar y realiza el cambio de contexto asignando el recurso de procesamiento, debido a esto es que en nuestro caso se intercalaban las funciones de WiFi y LoRa constantemente.

Por la forma en que funciona el scheduler de FreeRTOS, ahora que el proceso de prioridad alta esta bloqueado, el proceso de baja prioridad que no tendría que estar ejecutando va a correr hasta el próximo tick de reloj, en donde el scheduler verifica nuevamente los tasks actuales para darle la CPU al de mayor prioridad. Si el proceso LoRa que teníamos con prioridad alta dejó

de estar bloqueado antes del siguiente tick de reloj, va a figurar como disponible y por ser de mayor prioridad será asignado para seguir ejecutando.

Para solucionar este comportamiento no buscado, se utilizaron las funciones *vTaskSuspend(handlerTask)* y *vTaskResume(handlerTask)*. Como sus nombres indican, son funciones que suspenden o reanudan tasks, la particularidad de esto es que una task suspendida no es tomada en cuenta por el scheduler al momento de tener que buscar a alguien para ejecutar. Al final del día lo que se hace es suspender de forma mutuamente excluyente cada task.

Esta solución deja el núcleo 1 inactivo por mucho tiempo (tiempos en los que las tasks se bloquean mientras están en su turno de ejecución), sin embargo la solución deja lugar a nuevas posibles tasks que se quieran asociar a este núcleo y que puedan ejecutar mientras estas dos tareas principales (envió LoRa y WiFi) están bloqueadas.

5.1. Observación

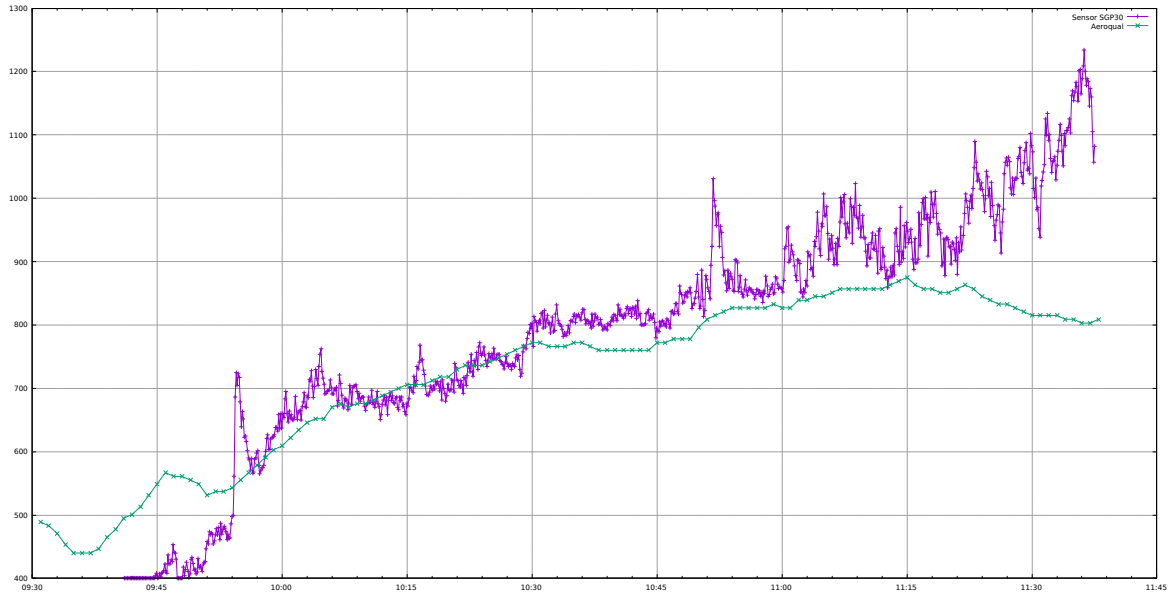
Problemas encontrados con las prioridades de FreeRTOS Cada task es asignada con una prioridad la cual es un valor entero entre 0 y (configMAX_PRIORITIES - 1), el valor de *configMAX_PRIORITIES* está definido en el archivo de configuración *FreeRTOSConfig.h*. Si para una task se elige una prioridad mínima (cero), puede existir la posibilidad tenga problemas al ejecutar debido a que FreeRTOS al momento de iniciar crea dos tasks de prioridad 0: *IDLE_0* e *IDLE_1* para cada núcleo respectivamente las cuales no ejecutan nada y están para ocupar el procesador cuando no hay nada que hacer [23]. Se sugiere manejarse con valores altos (de 10 para arriba) para no tener ningún problema del estilo.

6. Pruebas realizadas y resultados obtenidos

6.1. Pruebas del sensor

El dispositivo fue probado con el fin de determinar dos variables principales, la primera de ellas fue comprobar el funcionamiento del sensor SPG30 y ver como respondía en entornos realistas, como por ejemplo en un salón de clases. A su vez se realizaron mediciones con otros sensores disponibles en Facultad para comparar los resultados de cada uno y ver su exactitud.

Se tomó como medición 'verdadera' la del sensor Aeroqual del IMFIA, a continuación se muestran ambas mediciones contrastadas en una misma gráfica.



Grafica comparativa de las mediciones del SPG30 (violeta) y del Aeroqual (verde)

Como se puede apreciar en la imagen, la medición realizada por el sensor SPG30 sigue a grandes rasgos el patrón correcto, sin embargo se puede notar que tanto en el comienzo como en el final hay una diferencia notoria en la medición. Sumado a esto podría parecer que el sensor SPG30 es más susceptible a cambios drásticos en la medición. Conociendo estos datos, se concluye que es posible que sea necesaria la aplicación de una función de calibración en las mediciones del sensor. Esta queda pendiente como trabajo futuro puesto que se requieren más mediciones comparativas para determinar precisamente como debe ser esta función de calibración.

6.2. Pruebas de conectividad

La segunda variable que fue testeada en el dispositivo fue la conectividad tanto por WiFi como por LoRa, los resultados se presentan a continuación.

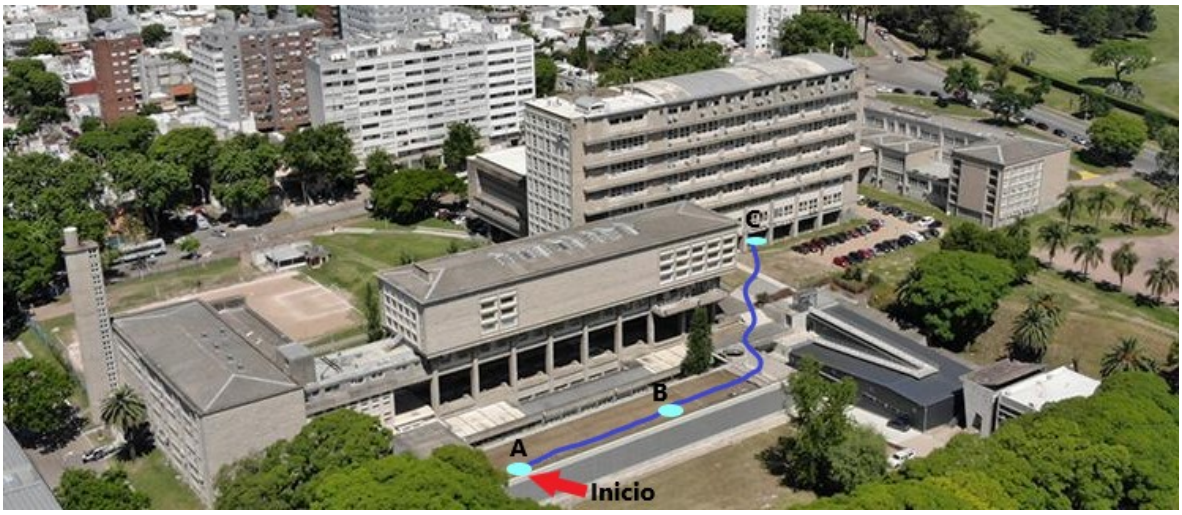
6.2.1. Conectividad con WiFi

- La conexión fue estable y el envío de datos fue constante en el edificio principal
- No se pudo conectar a la red en bandejas, más precisamente dentro del salón que está frente a la sala de máquinas Windows UdelaR A.
- No se pudo conectar a la red en el aulario, más precisamente en el salón B01.

6.2.2. Conectividad con LoRa

Con LoRa se realizó una prueba el día 19 de noviembre recorriendo distintos espacios de la Facultad, dando como resultado final que en todos los lugares se logró (en mayor o menor medida) enviar mensajes. El recorrido realizado comenzó en el techo del InCo desde donde se siguió hasta el bicicletero (debajo del edificio principal), se siguió hasta pasando la estatua de Da Vinci para luego volver hasta la entrada principal e ingresar al edificio. Desde la entrada frente al puesto de fotocopias se recorrió hasta el salón 307 y luego hacia las bandejas yendo por el primer piso. Por último se llegó al aulario donde se midió dentro del salón B01.

Se muestra el recorrido realizado junto con un aproximado de cómo fueron los envíos de datos en cada lugar, el dispositivo estaba configurado para realizar un envío cada 15 segundos.



Primera parte del recorrido

Los resultados de este primer tramo fueron:

- En A (Techo InCo al lado del comedor): el dispositivo no puede conectarse
- En B (Techo InCo más hacia el medio del edificio): el dispositivo logra conectarse y comienza el envío de datos cada 15 segundos.
- En C (Bicicletero): el dispositivo logra enviar datos pero con un poco de irregularidad

Segunda parte del recorrido

Los resultados de este segundo tramo fueron:

- En A (Estatua Da Vinci): el dispositivo envía cada 15 segundos sin problemas
- En B (Hall de entrada principal): el dispositivo tuvo problemas al conectarse, pasados los 5 minutos aproximadamente llegó a mandar un mensaje para luego continuar enviando aproximadamente cada dos minutos
- En C (Salon 307): el dispositivo logra enviar datos pero nuevamente con irregularidad
- En D (Bandejas en el piso superior): el dispositivo se comporta como en el hall principal, demora más de 5 minutos en enviar su primer mensaje y luego logra enviar cada aproximadamente 2 minutos
- En E (Pasillo exterior al lado del MAESO): se realizan envíos sin inconvenientes
- En F (Aulario salón B01): se realizan envíos cada 1 o 2 minutos

Se observó en los metadatos generados por Yeap que los parámetros de frecuencia, Signal-to-Noise Ratio y Spread factor utilizados por la antena van variando a lo largo de los envíos. Sabiendo esto, podría darse la posibilidad que la antena de Yeap tenga algunas frecuencias peores que otras, o que estén recibiendo otros datos externos y no esté atendiendo los nuestros. Por lo tanto como trabajo futuro sería buena idea recopilar estos datos en un periodo largo de tiempo y estudiar si se halla algún patrón de mejores rendimientos (o mayor proporción de datos recibidos) en ciertas combinaciones de frecuencias, spread factor, etc.

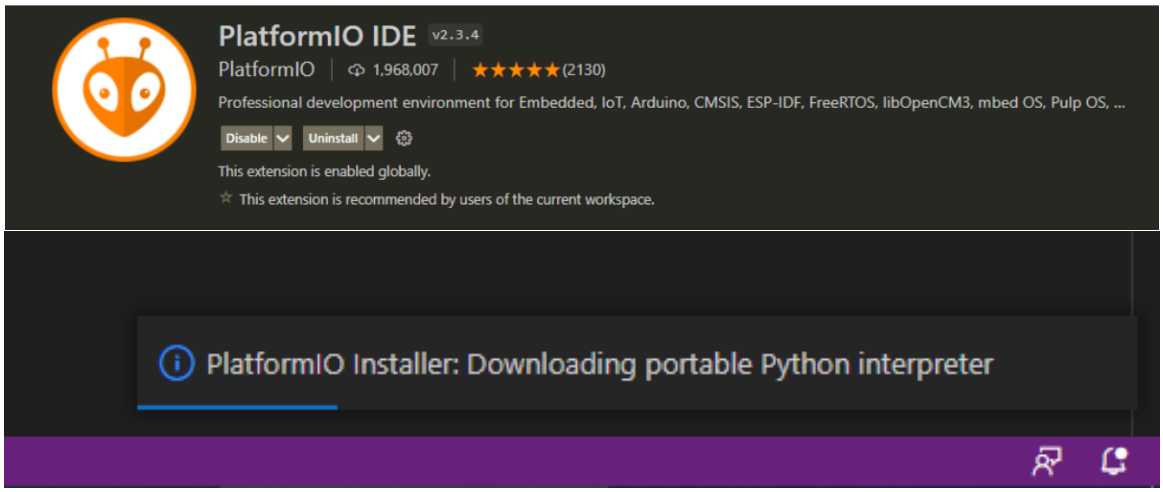
En la sección de Anexos se coloca una tabla que muestra el log de todos los mensajes recibidos en Yeap durante este recorrido.

7. Documentación sobre procedimientos

Esta sección agrupa todas las guías de procedimientos realizadas para facilitar la familiarización a las tecnologías.

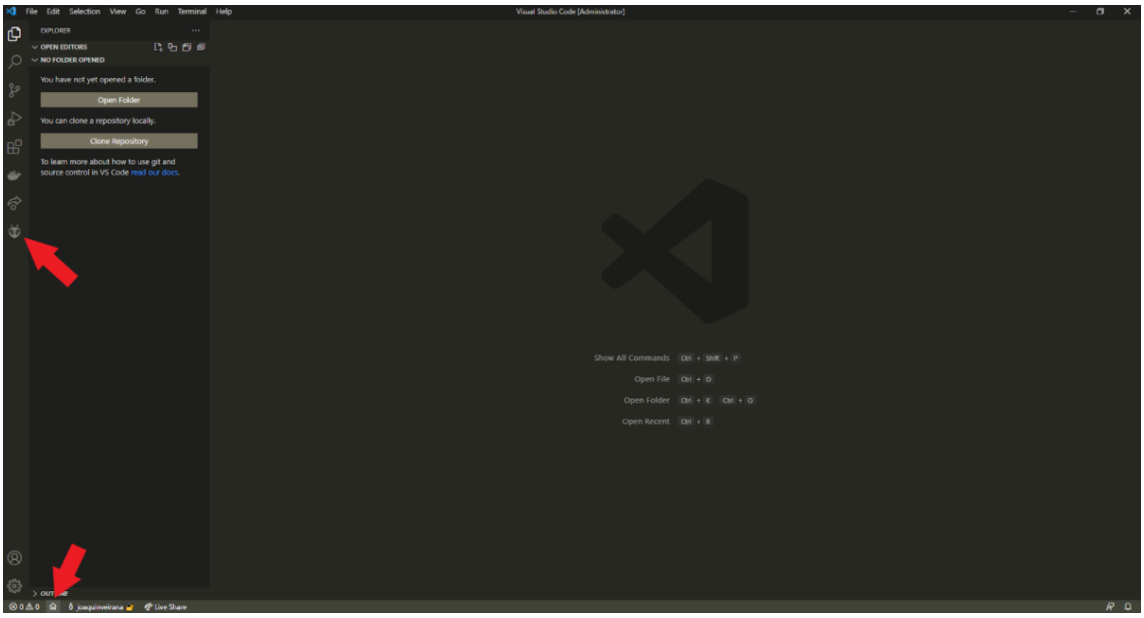
7.1. Set-up y programacion del dispositivo Sparkfun ESP32 LoRa 1-Channel Gateway con Platform IO

Para poder programar el controlador esp32 se comienza instalando PlatformIO como complemento de VScode. PlatformIO permite flashear el controlador con un solo click, este complemento se encarga de descargar e instalar los frameworks necesarios como el de espressif (esp-idf) o Arduino. PlatformIO permite además de esto instalar dependencias de software externas como por ejemplo LMIC.



Instalacion de PIO en VSCode

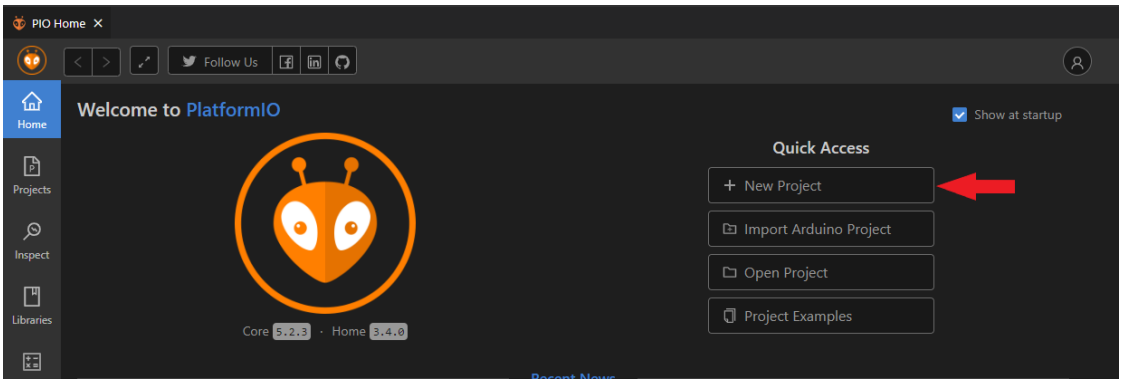
Una vez instalada la extensión, reiniciamos VSCode y ya estamos en condiciones de crear un proyecto para comenzar a desarrollar. Se debe poder visualizar un icono de una casa en la parte inferior izquierda de VScode así como el logo de PlatformIO en la parte izquierda.



Iconos que agrega PIO a VSCode

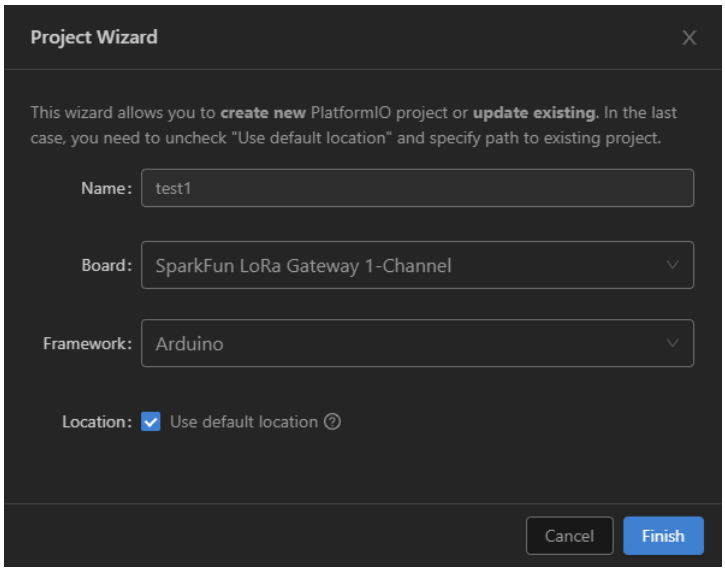
Al clickear en el icono de la casa se accede a la pestaña *PIO Home* en donde se puede crear un nuevo proyecto (*Home*), instalar bibliotecas externas (*Libraries*), ver la lista de dispositivos conectados a la computadora (*Devices*), entre otras opciones.

Para crear un nuevo proyecto se ingresa al menú *Home* en la pestaña *PIO Home*, y se cliquea en *New Project*.



Menu *Home* de *PIO Home*

Un formulario se despliega en el cual se debe ingresar un nombre al proyecto, la placa sobre la cual se va a trabajar (en este caso *Sparkfun ESP32 LoRa 1-Channel Gateway*) y un framework de trabajo (en este caso *Arduino*), opcionalmente se puede elegir la ubicación de guardado del proyecto.



Creación de un nuevo proyecto

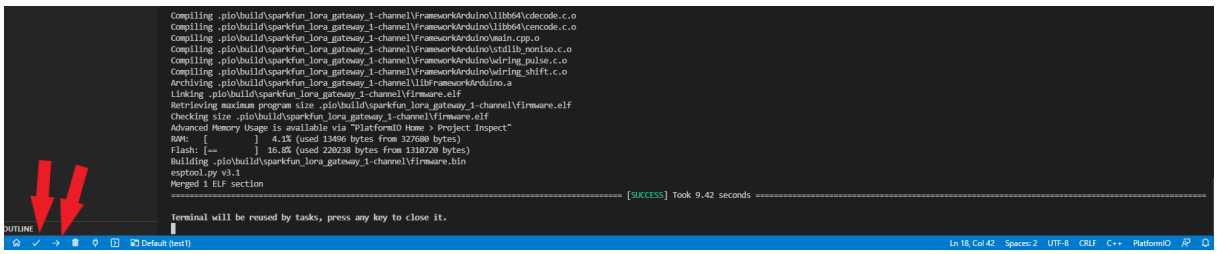
Una vez creado el proyecto se deben modificar dos archivos para poder probar el programa en la placa. El primer archivo es *platformio.ini* en donde se deben agregar las líneas *monitor_port* y *monitor_speed*, las primeras líneas que se muestran a continuación deberían ya estar al abrir el archivo.

```
[env:sparkfun_lora_gateway_1-channel]
platform = espressif32
board = sparkfun_lora_gateway_1-channel
framework = arduino

monitor_port = COM5
monitor_speed = 115200
```

Luego el segundo archivo es el *main.cpp*, en este caso el programa es hacer parpadear un led conectado al pin 23 del dispositivo.

```
#include <Arduino.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "driver/gpio.h"
#include "driver/adc.h"

#define BLINK_GPIO GPIO_NUM_23

void setup() {
  Serial.begin(115200);
  gpio_reset_pin(BLINK_GPIO);
  gpio_set_direction(BLINK_GPIO, GPIO_MODE_OUTPUT);

  while (1) {
    gpio_set_level(BLINK_GPIO, 1);
    vTaskDelay(100 / portTICK_PERIOD_MS);
    gpio_set_level(BLINK_GPIO, 0);
    vTaskDelay(100 / portTICK_PERIOD_MS);
  }
}

void loop() {}
```

Finalmente para hacer el build del proyecto y escribirlo en la placa, se utilizan los iconos de tick y flecha respectivamente que se encuentran a la derecha del icono de la casa en la parte inferior izquierda, como se muestra en la imagen a continuación.

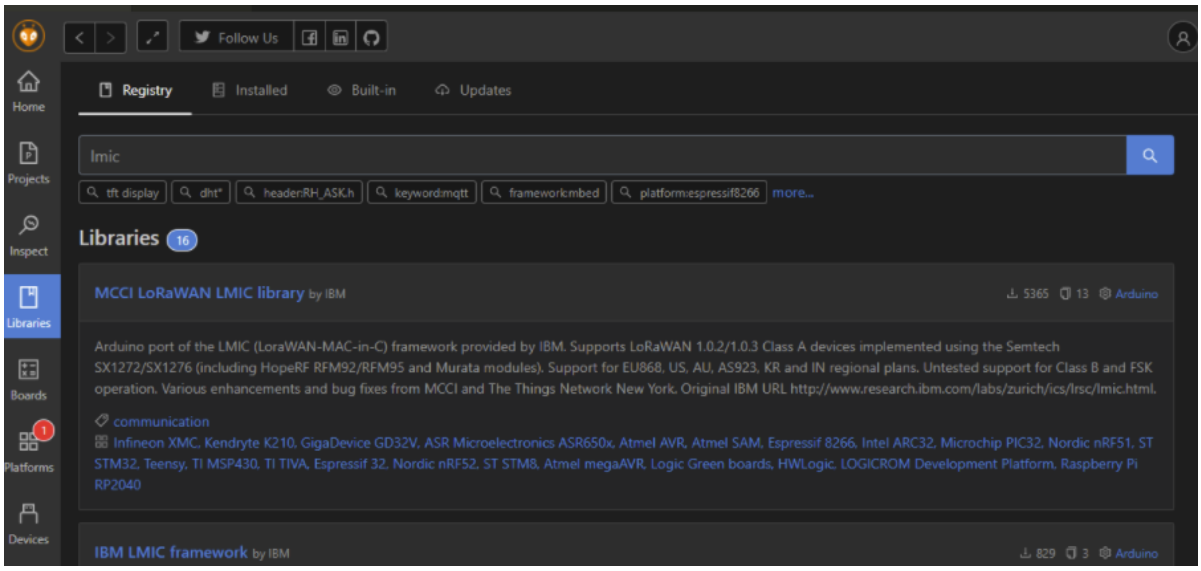


Build del programa con mensaje de Success

7.1.1. Observaciones

1) Monitor serial: El icono de enchufe que se encuentra al lado de los iconos de build y upload es usado para abrir el monitor serial, si el programa realiza prints estos van a visualizarse en el terminal al que se accede con este botón, lo cual es muy útil para debuggear.

2) Instalación de bibliotecas externas: Para instalar bibliotecas externas es necesario ir al menú *Libraries* en la pestaña *PIO Home*, en el menú se puede hacer uso del buscador para encontrar la dependencia (en este ejemplo LMIC). Una vez encontrada la dependencia basta con entrar a ella e instalarla, PlatformIO se encargará del resto.

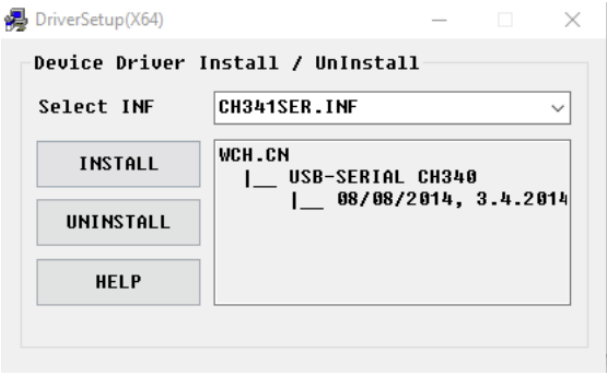


Biblioteca LMIC utilizada

Una vez instalada es posible que sea necesario agregarla al proceso de build de nuestro proyecto, para esto se debe agregar las siguientes líneas en el archivo *platformio.ini*. La biblioteca de LMIC en particular requiere de flags de entrada para la compilación como por ejemplo la frecuencia de trabajo (CFG_au915).

```
[env:sparkfun_lora_gateway_1-channel]
platform = espressif32
board = sparkfun_lora_gateway_1-channel
framework = arduino
monitor_port = COM5
monitor_speed = 115200

lib_deps =
    MCCI LoRaWAN LMIC library
build_flags =
    -D ARDUINO_LMIC_PROJECT_CONFIG_H_SUPPRESS
    -D CFG_au915=1
    -D CFG_sx1276_radio=1
```

3) Drivers USB en Windows: Para que el dispositivo sea reconocido por el sistema operativo es necesario instalar los drivers USB correspondientes [18], en el caso de Windows: CH341SER.EXE.



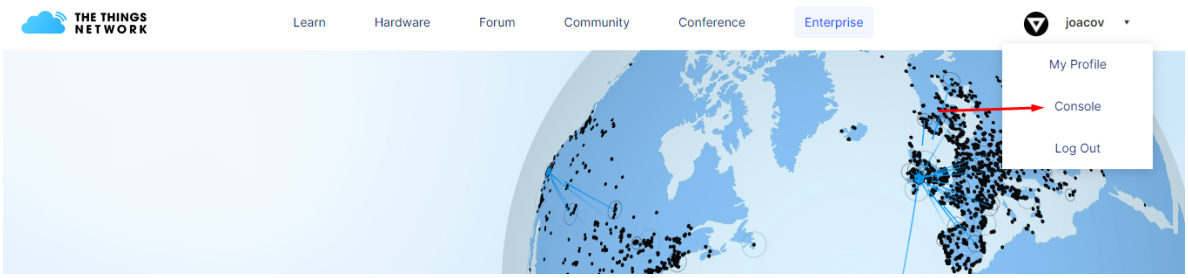
Drivers usados en Windows

7.2. Set-up y conexión del gateway a TTN V3

Para este tutorial se utilizó el gateway *The Things Gateway* fabricado por TTN. Incialmente es recomendado referirse a la guía oficial de TTN [16] así como a la documentación oficial con información extra sobre el dispositivo para facilitar la depuración de problemas [17].

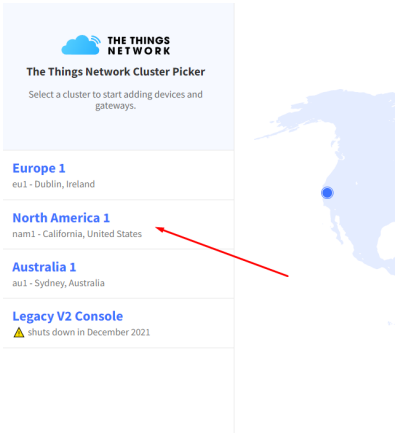
The Things gateway

Paso 1: Creación del gateway en la plataforma de TTN En este paso se crea una instancia del gateway en la consola de TTN, para esto es necesario registrarse en la plataforma e ingresar a la consola como se muestra en las siguientes imágenes.



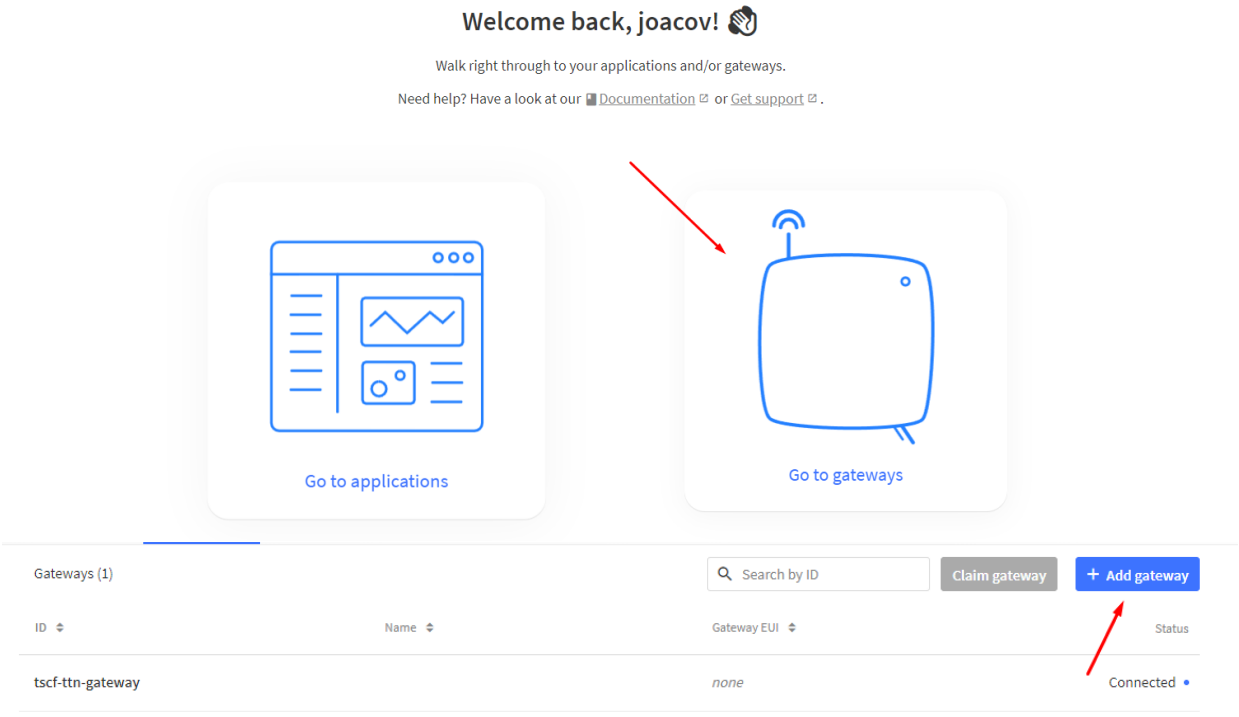
Ingresar a la consola

Una vez allí seleccionamos una zona de trabajo, en este caso se selecciona norteamérica, una vez que se elige una zona se ingresa a la consola.



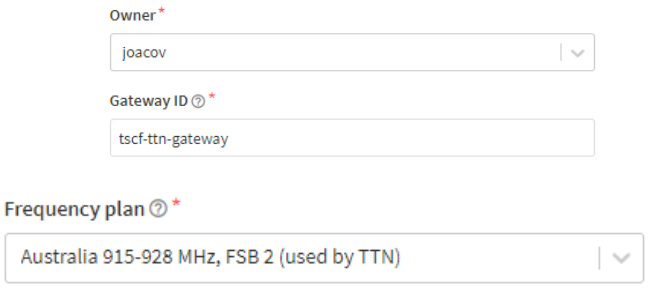
Selección de la zona

Una vez dentro, se ingresa al apartado *go to gateways* y se selecciona *Add gateway* para crear un nuevo gateway.



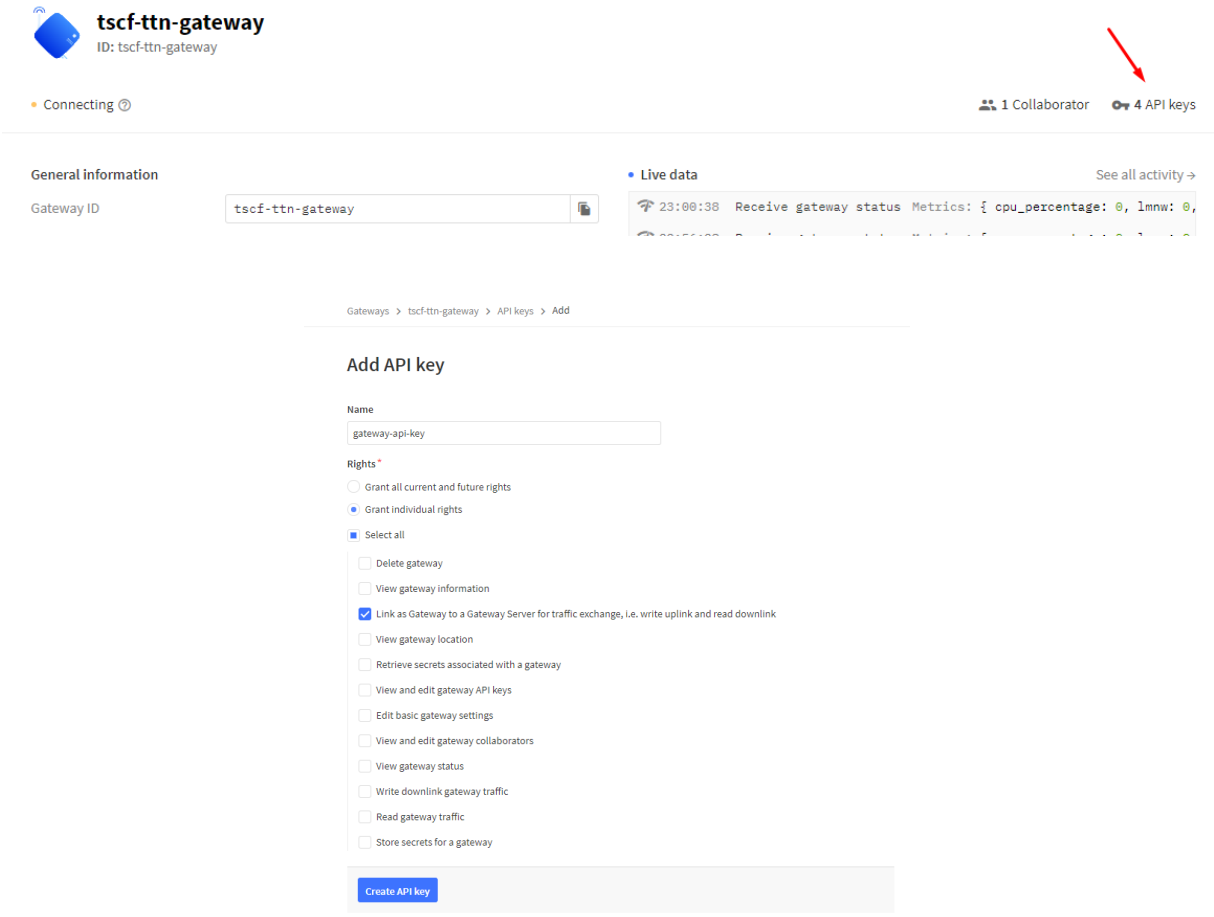
Creación de un nuevo gateway

Para dar de alta el gateway, basta con rellenar el campo id y seleccionar la frecuencia de trabajo de la antena del dispositivo, en este caso es una frecuencia de 915MHz.



Formulario de creación

Por último es necesario crear una api key para el gateway, para esto se debe ingresar a la sección *API Keys* dentro del dashboard del nuevo gateway creado anteriormente. Allí se debe crear una nueva api key y se le deben seleccionar los derechos de forma individual, seleccionando la opción *Link as Gateway to a Gateway Server for traffic exchange, i.e. write uplink and read downlink*. Es importante una vez creada la clave, copiarla y almacenarla en otro archivo ya que una vez creada no puede volver a verse.



Creación de la api key

Paso 2: Conexión del gateway a nuestra red wifi

1. Retirar la cobertura plástica blanca del gateway, dejando visible el botón rosado que hay en el interior

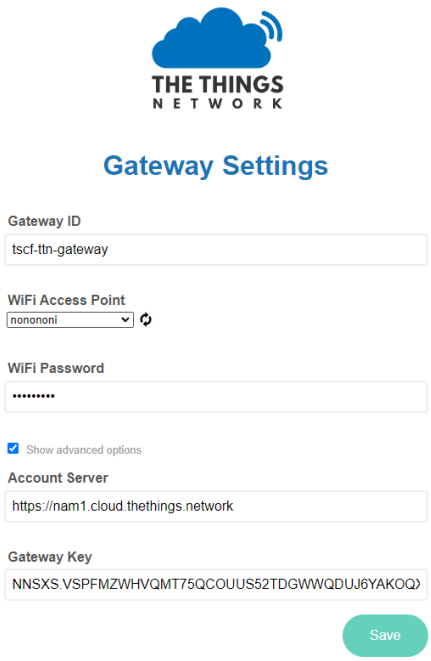
2. Conectar el dispositivo a la corriente, y si es posible conectarlo a la red por ethernet

3. Una vez el dispositivo está encendido, presione el botón rosado por 5 segundos, notar que las luces exteriores van encendiendo de a una hasta llegar a los 5 segundos

4. Acceder mediante una PC con acceso a WiFi a la red que genera el gateway, en mi caso **Things-Gateway-001EC03F986D** la clave para ingresar es **thethings**

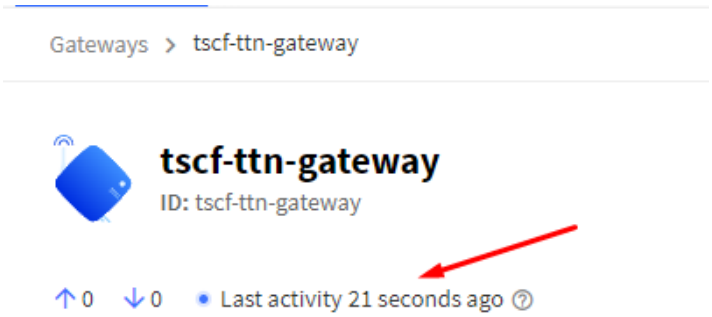
5. Una vez conectados a la red, ingresar en el navegador a la siguiente url **http://192.168.84.1/** en la cual el dispositivo levanta una aplicación web para ingresar los parámetros de activación.

6. Rellenar el formulario con el id de gateway creado en el paso 1, seleccionar nuestra red wifi y poner la clave. Seleccionar el checkbox *Show advanced options*, en el campo *Account Server* ingresar la url **https://nam1.cloud.thethings.network** si fue seleccionada la región norteamericana. Por último copiar la clave (api key) creada en el paso 1 en el campo *Gateway Key*.



Formulario de activación

Paso 3: Verificar que el gateway se conecto exitosamente Ingresar al dashboard del gateway, allí se debería mostrar que él fue visto hace X minutos como se muestra en la siguiente imagen.



Gateway conectado

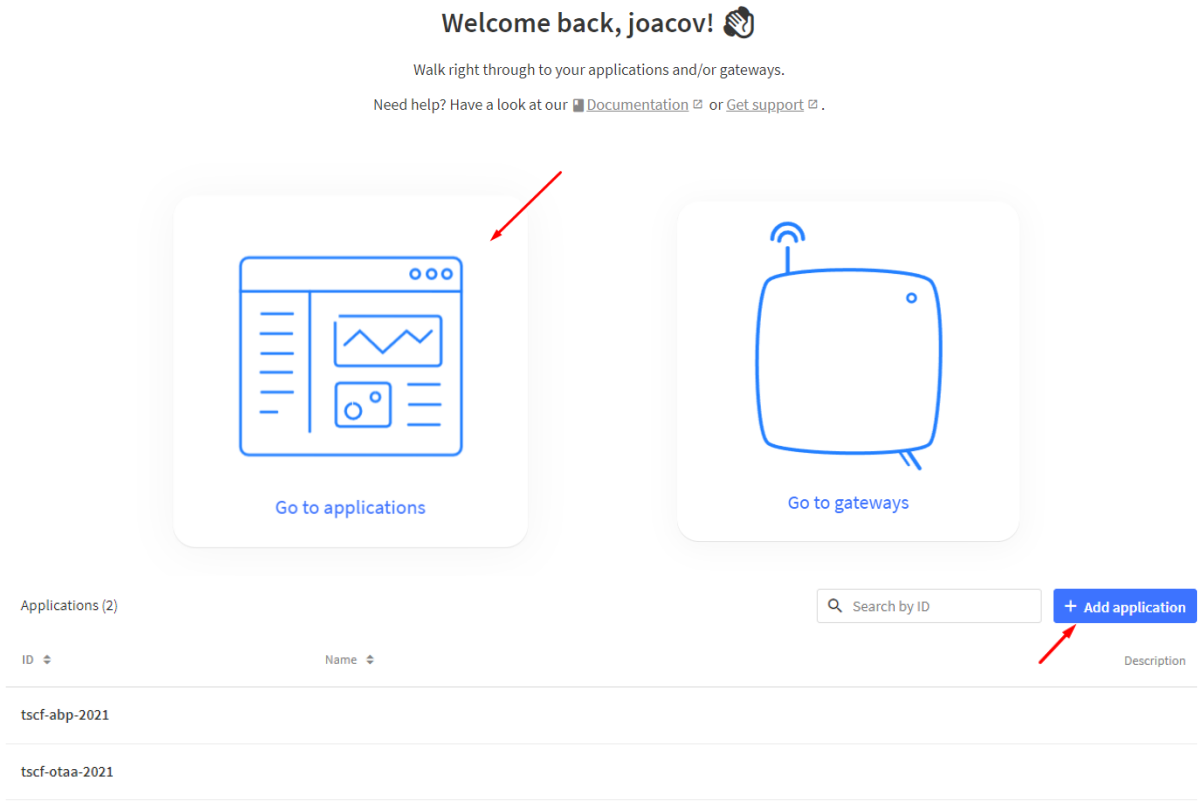
Otro detalle a tener en cuenta es que al estar conectado de manera exitosa, las cinco luces del dispositivo deben estar prendidas completamente sin parpadear. Si hay menos luces prendidas y alguna de ellas está parpadeando, referirse a la documentación [17] para corroborar que causa tiene el comportamiento de las luces. Si el dispositivo no figura conectado en el dashboard conviene volver a revisar todos los pasos en caso de haber cometido un error. También puede ser conveniente ingresar a la url **http://things-gateway.local/info** para poder visualizar información del dispositivo.

7.3. Set-up y conexión de un dispositivo sensor a TTN V3

A continuación se procede a explicar como crear todo lo necesario para que un dispositivo sensor envíe un *Hello World!* por medio de lora a TTN usando la activación OTAA, la activación y envío de datos por ABP es similar, sin embargo se recomienda ver el siguiente video [15] del canal biblioman09 donde se explica el paso a paso de manera precisa. Este programa de envío también funciona para enviar datos a la antena Yeap de la Facultad.

Antes de comenzar es necesario crear y conectar el gateway a la red TTN, ver la guía de esto en Set-up y conexión del gateway a TTN V3. Sumado a esto es necesario ya tener listo el entorno para flashear el dispositivo así como un proyecto armado con LMIC instalado, para ver la guia de esto ver la sección Set-up y programación del dispositivo Sparkfun ESP32 LoRa 1-Channel Gateway con Platform IO.

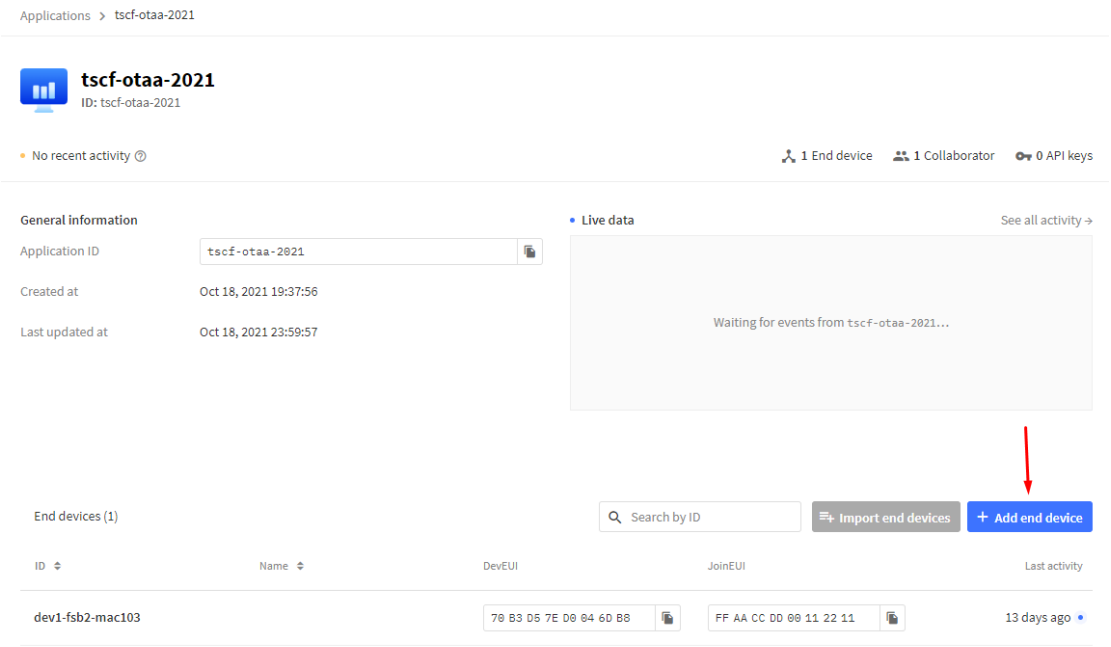
Paso 1: Crear una aplicación en TTN En este paso se crea una aplicación en la plataforma y se le da de alta un dispositivo para el cual es necesario definir ciertos parámetros entre ellos una clave de sesión. Primero se debe ingresar a la aplicación web de TTN y posteriormente en la consola e ir al apartado *go to applications*. Luego se selecciona *Add application* para crear la nueva aplicación.



Creación de una nueva aplicación

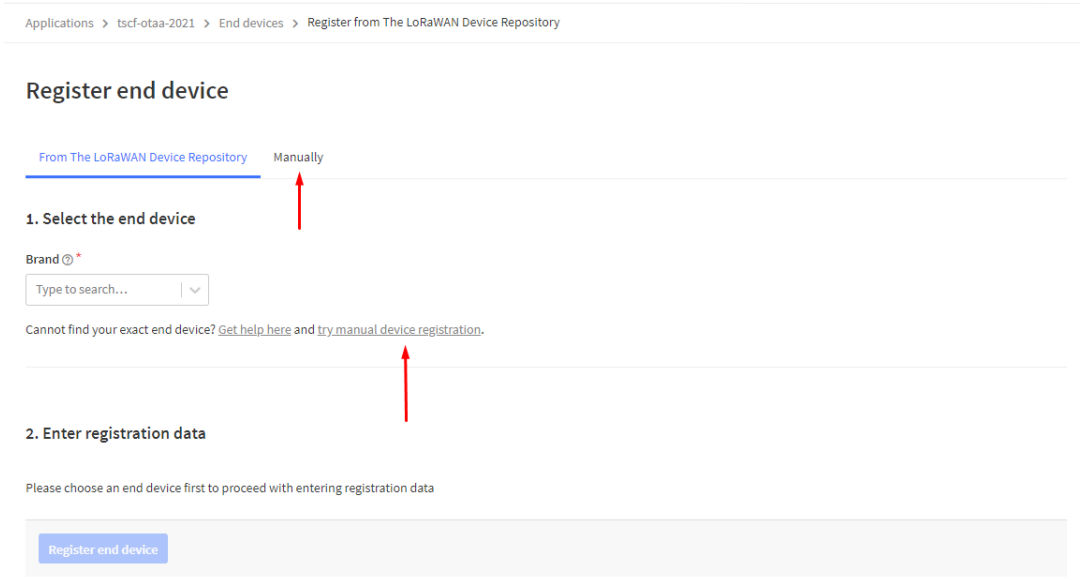
Para crear una nueva aplicación basta con ponerle un id único.

Paso 2: Dar de alta un dispositivo en la aplicación En este paso se crea un dispositivo asociado a la aplicación, para esto se ingresa a *Add end device* dentro de la aplicación.



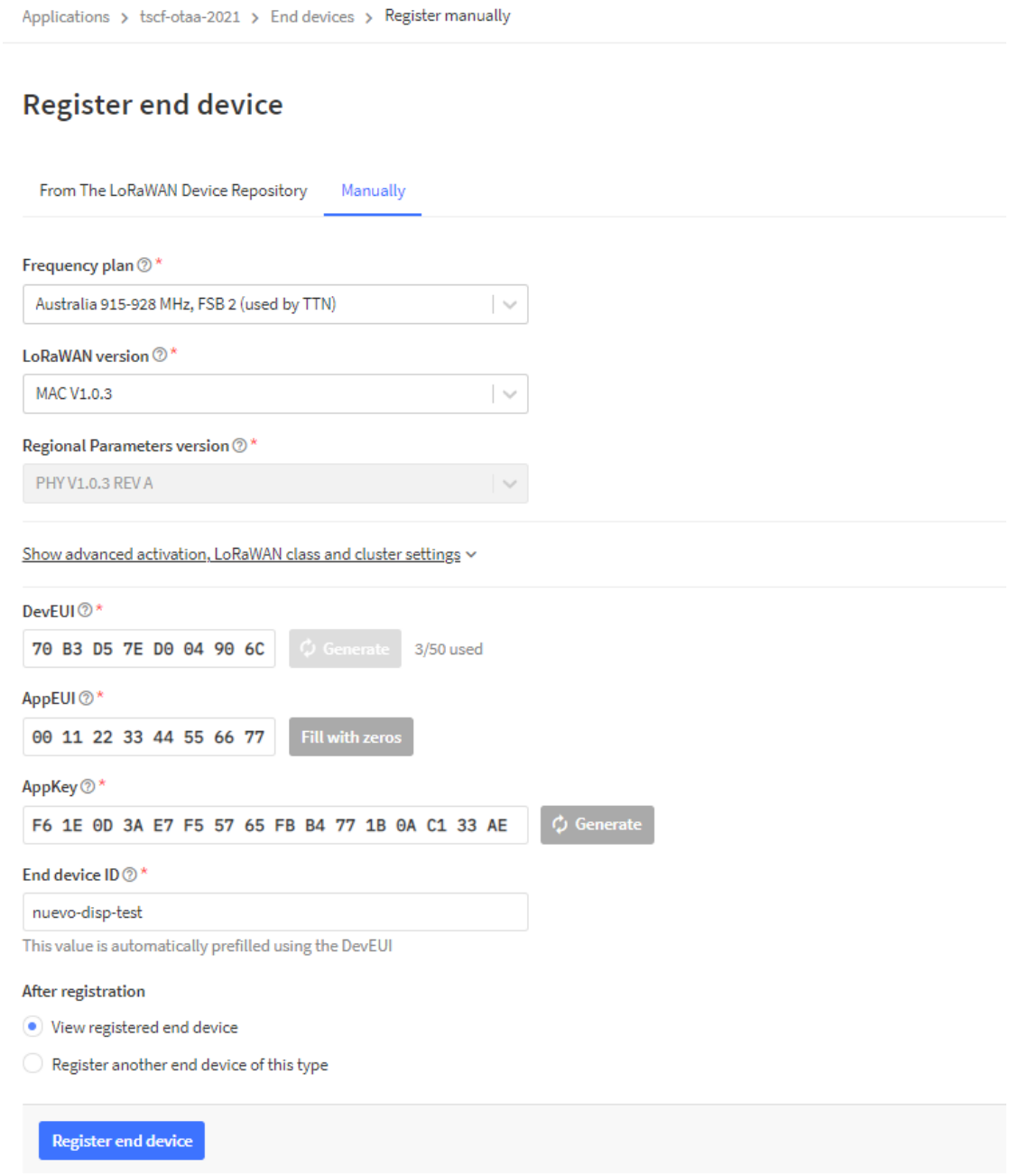
Creación de un nuevo dispositivo

Para crear el dispositivo se debe ingresar a la sección de registro manual como se muestra en la siguiente imagen.



Ingreso a registro manual

La pantalla del registro manual consta de un formulario en donde hay que seleccionar varias cosas, primero que nada la frecuencia, esta debe coincidir con la del gateway. Luego la versión de protocolo LoraWAN cuyo valor puede ser MAC V1.0.3 y por último las credenciales de sesión. En las credenciales se tiene un botón para autogenerarlas, sin embargo en el campo *AppEUI* el botón lo que hace es llenar todo con ceros, aquí nosotros debemos rellenar este campo con valores a elección pero no dejarlo todo en cero. A continuación se deja un ejemplo.



Ejemplo de formulario de creación

Paso 3: Crear el código para el envío y subirlo a la placa Ahora es necesario cargar el código con las funciones de envío en el dispositivo, este código está disponible como ejemplo en el repositorio de LMIC [6] en el archivo *examples/ttn-otaa/ttn-otaa.ino*.

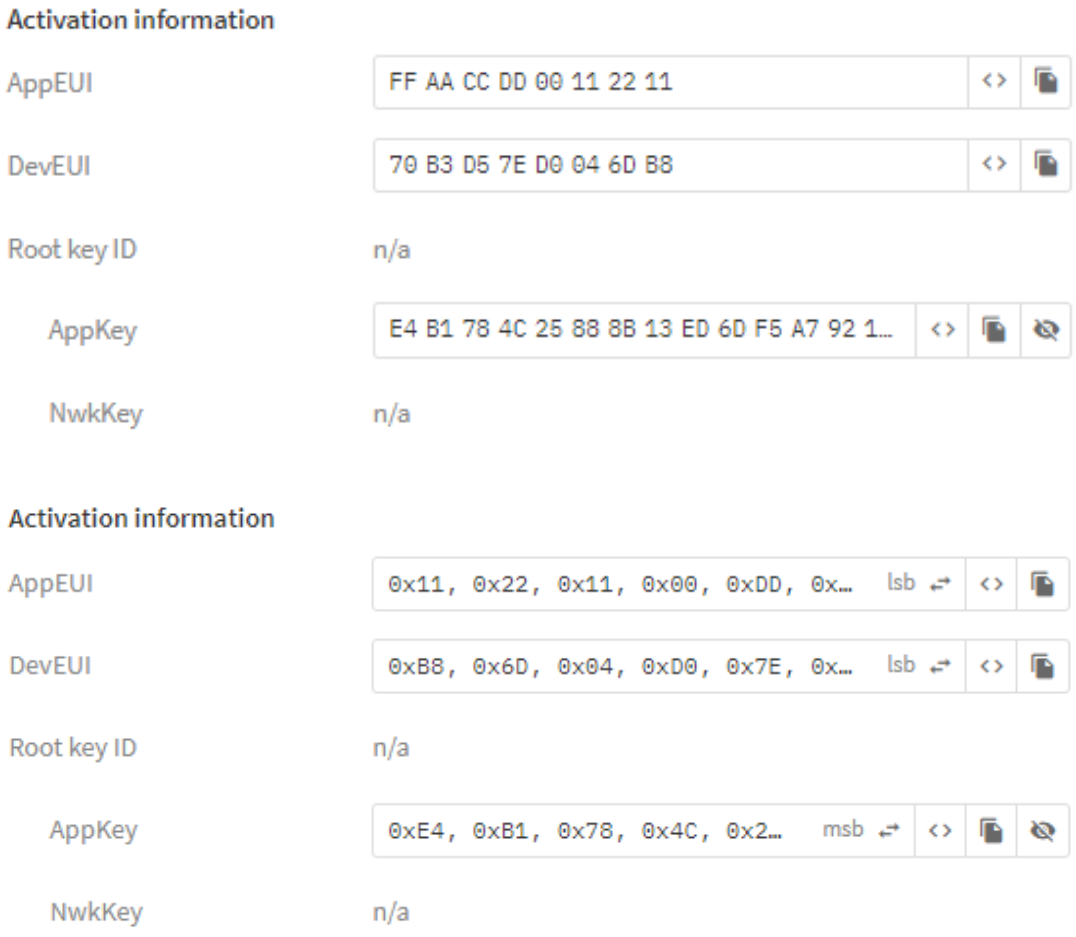
Basta con copiar este código entero y pegarlo en el main.cpp de nuestro proyecto, puede que sea necesario reordenar algunas definiciones de funciones para que no sean invocadas antes de ser declaradas. Lo siguiente es cargar las credenciales de activación de nuestro dispositivo de TTN en el código, estas credenciales deben estar cada una en una cierta endianness (el orden en el que van los bytes al ser almacenado).

Estas credenciales son:

- **APPEUI**: este campo debe estar en little-endian

- **DEVEUI**: este campo debe estar en little-endian
- **APPKEY**: este campo debe estar en big-endian

Ingresando dentro de la página del dispositivo en TTN es posible obtener estos valores ya formateados de la manera necesaria. Si nuestro dispositivo tiene las credenciales de activación de la siguiente imagen, es posible obtener los formatos buscados primero pasandolos a array con el botón <>, y luego utilizando la conversión de endianness (*lsb* es little-endian y *msb* el big-endian).



En este ejemplo el código final debe tener los siguientes valores cargados:

```
static const u1_t PROGMEM APPEUI[8]=
    { 0x11, 0x22, 0x11, 0x00, 0xDD, 0xCC, 0xAA, 0xFF };

static const u1_t PROGMEM DEVEUI[8]=
    {  0xB8, 0x6D, 0x04, 0xD0, 0x7E, 0xD5, 0xB3, 0x70 };

static const u1_t PROGMEM APPKEY[16] =
    { 0xE4, 0xB1, 0x78, 0x4C, 0x25, 0x88, 0x8B, 0x13,
    0xED, 0x6D, 0xF5, 0xA7, 0x92, 0x1F, 0x10, 0xB4 };
```

Lo último a reemplazar es la variable *lmic_pinmap lmic_pins* que contiene el mapeo de pines de la placa. Cada placa tiene su propio mapeo de pines, en este caso la *Sparkfun ESP32 LoRa 1-Channel Gateway* tiene que tener la siguiente variable.

```
const lmic_pinmap lmic_pins = {
    .nss = 16,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = 5,
    .dio = {26, 33, 32},
};
```

Habiendo realizado todo esto ya estaríamos en condiciones de subir el código al dispositivo y comenzar a recibir mensajes.

Paso 3: Verificar que el dispositivo se conectó exitosamente Yendo a la página del dispositivo en TTN es posible ver si se están recibiendo datos, lo primero a tener en cuenta es el estado de conexión el cual indica hace cuanto tiempo se recibió algo por última vez.

Estado de conexión del dispositivo

También es posible ver cuántos paquetes fueron recibidos y enviados en total (en las flechas debajo del nombre del dispositivo). En la pestaña *Live Data* se pueden ver los mensajes.

Por otra lado, es posible ver los prints en el monitor serial de PlatformIO, en donde si el dispositivo está funcionando correctamente debería verse algo similar a esto:

```
------------------------
1981300: [LMIC] EV_JOINED
netid: 19
devaddr: 260CE502
AppSKey: 24-DD-D8-26-B9-9E-4E-3A-11-8A-97-6A-35-6B-A4-25
NwkSKey: E1-A7-0C-E6-2B-4D-8C-A1-BE-A8-62-9C-C8-24-DE-C5
1981585: [LMIC] EV_TXSTART
2343248: [LMIC] EV_TXCOMPLETE (includes waiting for RX windows)
2373885: [LMIC] EV_TXSTART
2758228: [LMIC] EV_TXCOMPLETE (includes waiting for RX windows)
[DATA] Datos sensados en func Lora: 65.10000/27.50000/400.0000/60000.00/0.000000/01/00
4653074: [LMIC] EV_TXSTART
[LMIC] Packet queued
4980033: [LMIC] EV_TXCOMPLETE (includes waiting for RX windows)
5010669: [LMIC] EV_TXSTART
5392114: [LMIC] EV_TXCOMPLETE (includes waiting for RX windows)
[DATA] Datos sensados en func Lora: 65.00000/27.60000/400.0000/60000.00/0.000000/01/00
7286944: [LMIC] EV_TXSTART
```

Logs de conexión (verde) y de envío de datos exitoso (rojo)

7.3.1. Observaciones

Si el dispositivo no logra conectarse con el gateway, es conveniente revisar que las credenciales fueran colocadas en los formatos correctos. También es buena idea ver el estado de conexión del gateway, tanto en la plataforma como en sus luces de conexión (las 5 luces azules deben estar prendidas).

Si en los logs del dispositivo se está haciendo print del mensaje *EV_JOIN_TXCOMPLETE: no JoinAccept* significa que no se está pudiendo hacer el handshake de activación y puede darse porque el dispositivo no tiene señal como para llegar a enviar al gateway o los mensajes de ack del gateway no llegan al dispositivo.

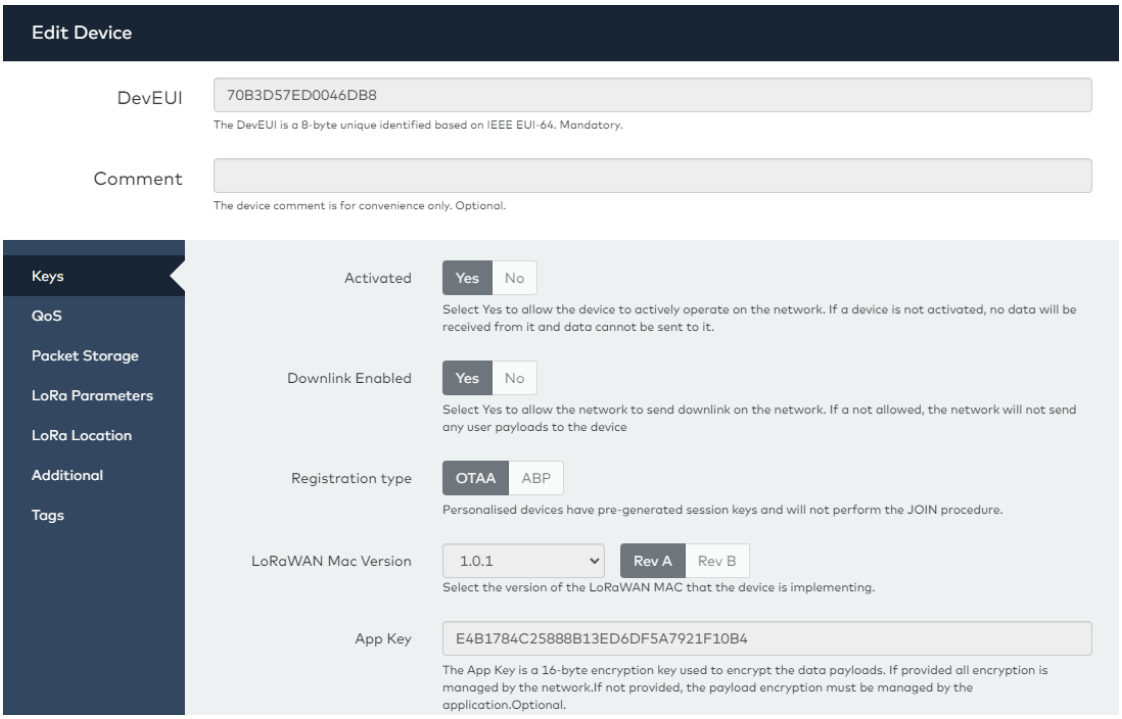
7.4. Set-up y conexión de una aplicación y dispositivo en Yeap

En esta sección se presenta una guía sobre la plataforma Yeap/Orbiwise y sobre cómo configurar el reenvío de datos hacia nuestra aplicación.

Paso 1: Ingresar a la plataforma Se ingresa a la plataforma Orbiwise [13]. El usuario usado en el proyecto fue *fing_uy* y la clave *Tscf_fing2021!*

Menu principal de la plataforma Orbiwise

Paso 2: Crear un dispositivo Bajo la pestaña *Devices/Manage Devices* se ingresa al menú de dispositivos, estos dispositivos funcionan de manera similar a los dispositivos en TTN. Para crear un dispositivo puede hacerse con el botón *Add Device w/o profile*, allí se despliega un formulario en donde hay que ingresar el *Dev EUI*, el *App Key* y la versión de LoRaWAN. A continuación se muestra la aplicación creada para el proyecto.

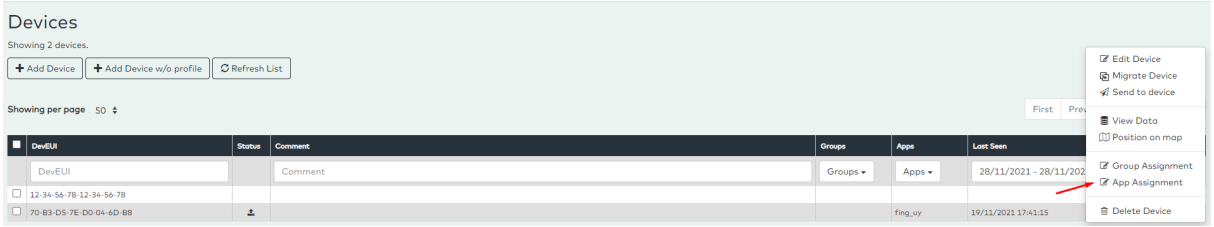


Menu principal de la plataforma Orbiwise

Recordemos que el código en nuestro sensor tiene las siguientes credenciales:

```
static const u1_t PROGMEM APPEUI[8]=
    { 0x11, 0x22, 0x11, 0x00, 0xDD, 0xCC, 0xAA, 0xFF };

static const u1_t PROGMEM DEVEUI[8]=
    {  0xB8, 0x6D, 0x04, 0xD0, 0x7E, 0xD5, 0xB3, 0x70 };

static const u1_t PROGMEM APPKEY[16] =
    { 0xE4, 0xB1, 0x78, 0x4C, 0x25, 0x88, 0x8B, 0x13,
    0xED, 0x6D, 0xF5, 0xA7, 0x92, 0x1F, 0x10, 0xB4 };
```

Paso 3: Crear una aplicación y asociar el dispositivo Una vez creado el nuevo *Device* es necesario asociarlo a una *Application*, para eso ingresamos a la pestaña *Application/Manage Applications* y clickeamos el botón *Add Application*. Una vez creada la aplicación volvemos a *Manage Devices* y en la lista de dispositivos buscamos el indicado. Usando el botón *Action* ingresamos a *App Assignment* y lo procedemos a asignar a la application creada.



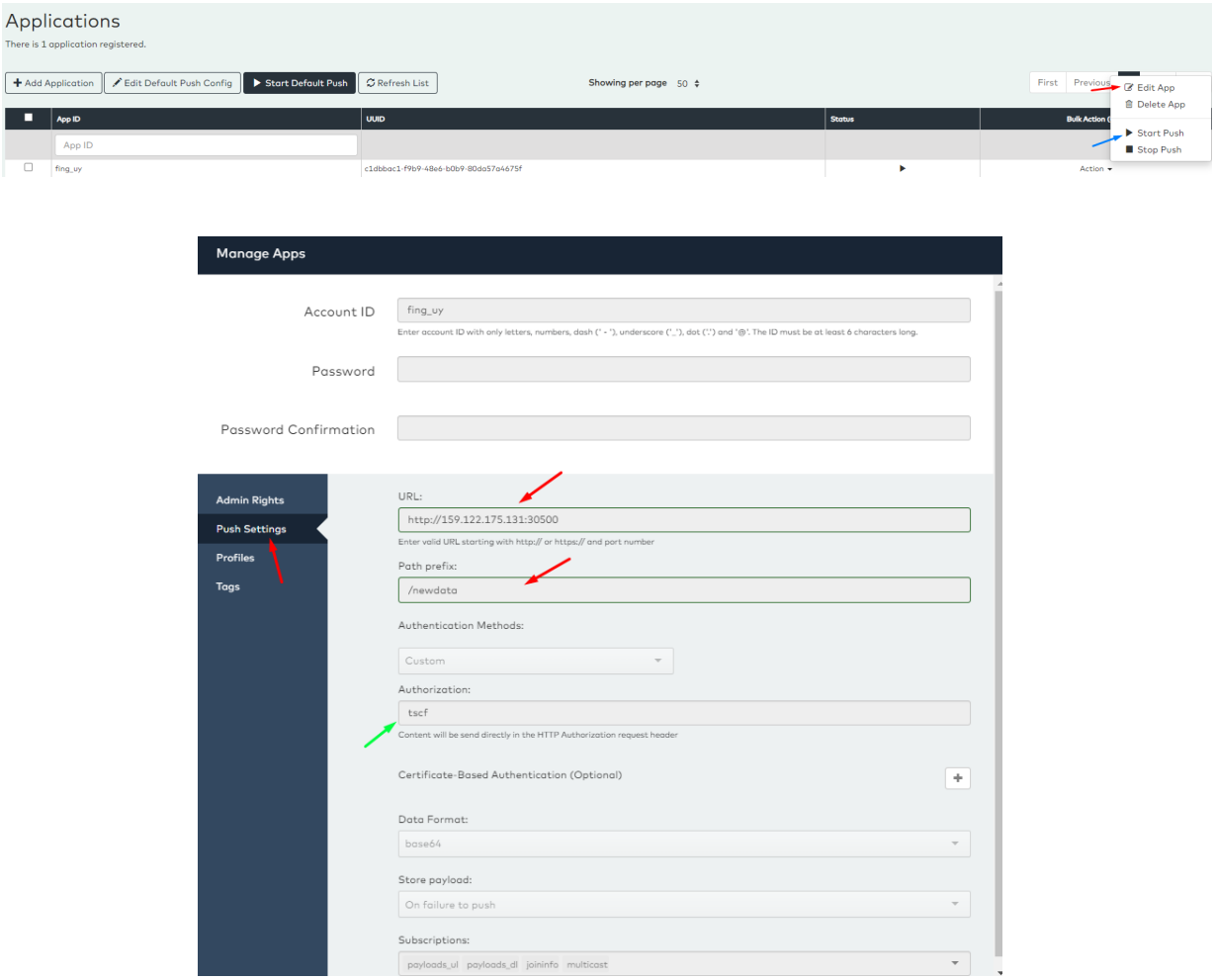
Asociar el dispositivo a la aplicación

Paso 4: Configurar el reenvío de datos hacia nuestra aplicación Por último volvemos a la sección *Manage Applications* y con el botón *Action* ingresamos a *Edit Application*, dentro del menú de configuración nos dirigimos a *Push Settings* e ingresamos la url de nuestra aplicación junto con el path donde se deben realizar los request POST de datos.

En la imagen a continuación se muestran las configuraciones de la aplicación usada en el proyecto la cual era accesible por la IP 159.122.175.131 y el puerto 30500. Se puede ver también que se configuraron los envíos al path */newdata* en nuestra aplicación. Hay que tener en cuenta al momento de programar las rutas de los endpoints de nuestra aplicación que Yeap le agrega más sufijos al path, en el caso del POST de datos se le agrega */rest/callback/payloads/ul*, por lo que teniendo esto en cuenta el path final es */newdata/rest/callback/payloads/ul*.

Opcionalmente, Yeap envía por medio de un PUT al path */newdata/rest/callback/nodeinfo* metadatos sobre las transmisiones recibidas.

Es posible también agregar un header *Authorization* con algún valor en particular para que nuestra aplicación pueda realizar algún tipo de chequeo de autenticación. En nuestro caso se agregó el string '*tscf*' en este header para realizar la validación (esta es muy básica y no brinda seguridad casi, hay otras opciones mas seguras como certificados).



Menú de configuración de reenvío de datos

Paso 5: Activar el reenvío de datos hacia nuestra aplicación Aquí basta con ir a la lista de aplicaciones y en el botón *Action* activar el envío de datos con la opción *Start Push* (En la imagen superior se muestra con la flecha azul).

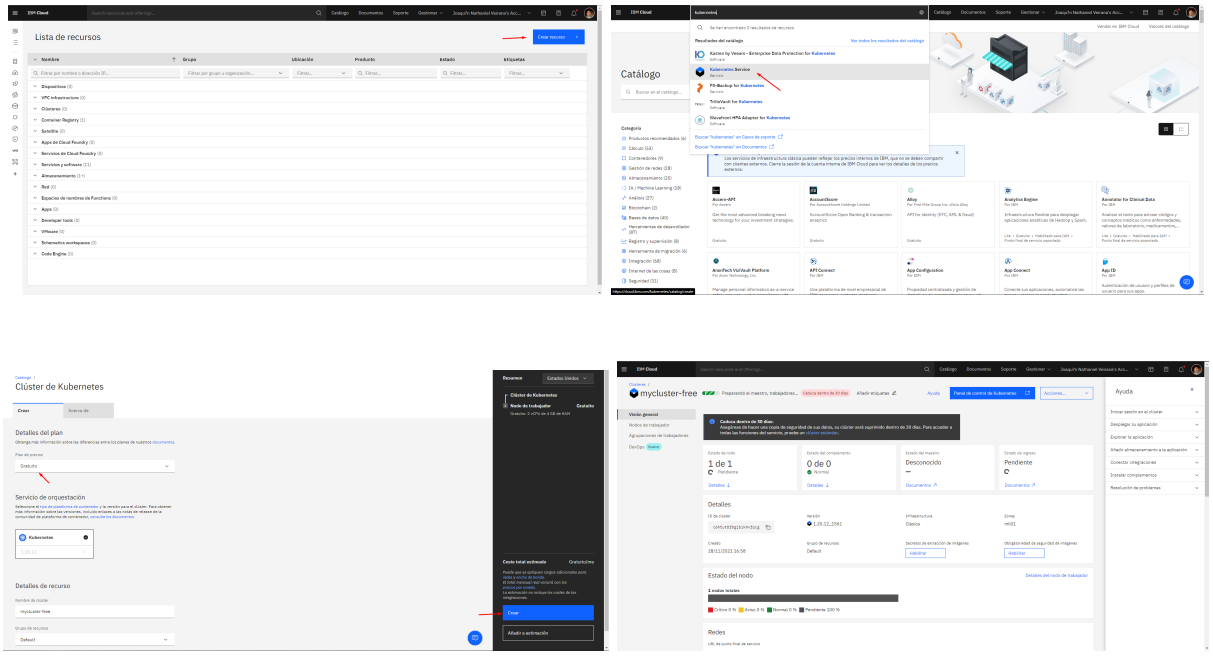
7.4.1. Observación

En Yeap es posible visualizar los logs de actividad de todas nuestras aplicaciones en la sección *Applications/Event Log*, es de allí que se obtuvo la tabla de datos mostrada en el anexo Logs de las pruebas de conectividad realizadas en Facultad.

7.5. Despliegue de la aplicación web en Kubernetes en IBM Cloud

En esta sección se describe a nivel general como desplegar nuestro cliente y servidor en un cluster de kubernetes en IBM Cloud. Esta guía está simplemente por si se quiere replicar la solución realizada en el proyecto de forma completa. Antes de comenzar con el despliegue es necesario tener instalado Docker y la CLI de IBM Cloud [19] junto con el plug-in *IBM Cloud Kubernetes Service CLI plug-in* [20]. Además es necesario tener una cuenta en IBM Cloud.

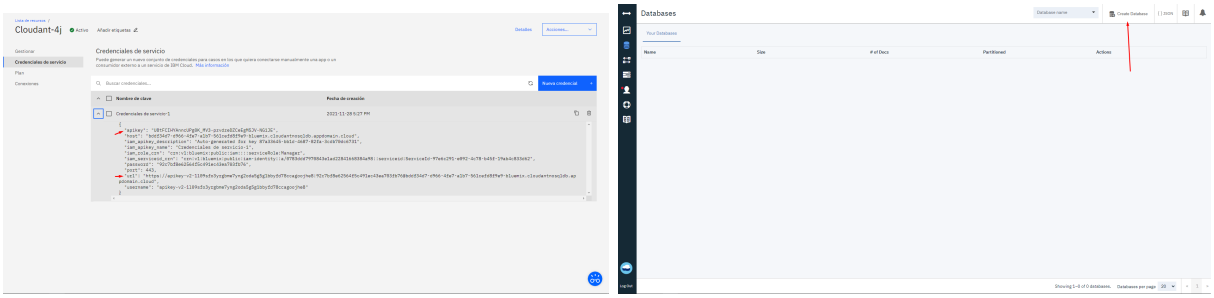
Paso 1: Creación del cluster en la nube Dentro de IBM Cloud se cliquea en *Crear recurso* y se busca por Kubernetes service, al seleccionar esta opción se selecciona la versión gratuita y se cliquea en crear. Aquí hay varias observaciones a tener en cuenta, por más que sea gratis, hay que ingresar una tarjeta de crédito para poder usar el servicio gratuito y el cluster gratuito se eliminará pasados los 30 días. Luego de unos 20 minutos se termina de crear el cluster.



Creación del cluster

Paso 2: Creación de la base de datos Cloudant Como en el paso 1, dentro de IBM Cloud se cliquea en *Crear recurso* y se busca por Cloudant, al seleccionar esta opción se selecciona la versión gratuita o lite y se cliquea en crear.

En la página del servicio de Cloudant es necesario ir a la sección *Credenciales de servicio* y crear unas nuevas credenciales, dentro de estas credenciales es necesario guardar el campo *URL* y *Apikey*. Luego se debe ingresar al Dashboard de Cloudant, para esto se usa el botón *Launch Dashboard* en la sección *Gestionar*. Allí se crean las bases de datos: *datos-sensados*, *salones* y *metadata*.



Creación de las credenciales y las bases de datos en Cloudant

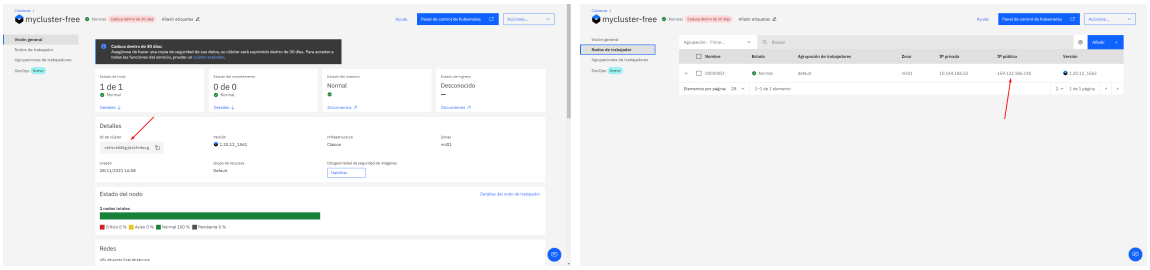
Paso 3: Creación de las imágenes de la aplicación y su subida a la nube En este paso se crean las imágenes Docker del cliente y del servidor y se suben a la nube, se documentarán los pasos para el servidor pero de manera análoga se realizan para el cliente.

Abriendo un terminal dentro de la carpeta de la aplicación *servidor* en el proyecto (carpeta que contiene el archivo Dockerfile) se usan los siguientes comandos:

```
// Login en la nube con email y pass
ibm cloud login
// Login al Container Registry (donde se guardan las imágenes)
ibm cloud cr login
// Accedemos a la region global del Container Registry
ibmcloud cr region-set global
// Creacion de la imagen y su envío al CR
docker build -t <nombre-imagen> .
docker tag <nombre-imagen> icr.io/<namespace>/<nombre-imagen>
docker push icr.io/<namespace>/<nombre-imagen>
```

Terminados estos pasos la imagen debería quedar subida a la nube y estar lista para ser usada en Kubernetes.

Paso 4: Despliegue de la imagen subida en kubernetes Una vez subidas las imágenes, dentro del mismo directorio de la aplicación (en este caso la aplicación *servidor*), se encuentran unos archivos de extensión .yaml. Estos archivos son usados para desplegar y exponer la aplicación (*servidor.yaml*), así como para declararle variables de entrono (*secrets.yaml*). Primero es necesario obtener el id del cluster creado (el cual llamaremos <id-cluster>) y su IP pública (la cual llamaremos <ip-publica-cluster>), estos se hallan en el dashboard del cluster en IBM Cloud.



Identificador e IP Pública del cluster respectivamente

Luego es necesario editar los archivos *secrets.yaml* y *servidor.yaml*. En *secrets.yaml* se hallan variables de entorno que debemos colocar, en particular:

- *window.REACT_APP_API_ENDPOINT* = "http://<ip-publica-cluster>:30500"
- *cloudantApikey*: apikey de nuestro cloudant
- *cloudantURL*: url de nuestro cloudant
- *cloudantRoomsDB*, *cloudantDataDB*, *cloudantMetadataDB*: nombres que les hayamos dado a las bases de datos creadas en cloudant

Ejemplo del archivo *secrets.yaml*

En *servidor.yaml* solamente debemos modificar la linea *image* que se encuentra anidada en la sección de containers. En el valor de image se debe ingresar la dirección del container registry a donde se subió la imagen. Si recoramos el paso anterior, este valor es *icr.io/<namespace>/<nombre-imagen>*. A continuacion se deja una imagen de ejemplo del archivo luego de ser subida la imagen bajo el comando *docker push icr.io/tscf/tscf-servidor*.

Ejemplo del archivo *servidor.yaml*

Finalmente aplicamos los archivos .yaml en nuestro cluster de kubernetes, logrando así el despliegue de la aplicación. El primer comando a continuación utiliza el id de cluster obtenido del dashboard anteriormente en este paso.

```
// Referenciar al cluster creado
ibm cloud ks cluster config --cluster <id-cluster>
// Aplicar archivos yaml (variables de entorno y despliegue)
kubectl apply -f secrets.yaml
kubectl apply -f servidor.yaml
```

Luego de aplicar el despliegue del cliente y el servidor la aplicación debería estar accesible en la dirección http://<ip-publica-cluster>:30501.

7.6. Set-up y configuración del dispositivo sensor con el código del prototipo final

Al momento de querer crear un dispositivo sensor con el código realizado para la tercera versión del proyecto se necesitan configurar algunas variables tanto para personalizar el comportamiento del dispositivo sobre cómo se envían los datos, así como para setear los endpoints y credenciales necesarias para el envío de datos por WiFi.

Dentro del proyecto, en el directorio *Codigo Principal/include* se encuentra el archivo *GlobalData.h* que contiene todas estas definiciones, a continuación se muestra el contenido del archivo junto con los valores que fueron usados en las pruebas a modo de ejemplo.

```
// Credenciales WIFI y Tiempo de espera total al realizar una conexion
#define WIFI_NETWORK "wifing"
#define WIFI_PASSWORD "wifing-pub"
#define WIFI_TIMEOUT_MS 20000

// Id que se quiere asignar al dispositivo (este id es enviado en cada medicion)
#define DEVICE_ID "01"

// Delay entre mediciones cuando se envia por WIFI (en ms)
#define DELAY_BETWEEN_SENSOR 5000
// Delay entre mediciones cuando se envia por LORA (en seg)
#define TX_INTERVAL 15

// Endpoint de la aplicacion servidor
// Necesario para envios directos por wifi, usa la <ip-publica-cluster>
#define SERVER_URL "http://159.122.175.131:30500/newdata/wifi"

// Pin al que esta conectado el led indicador de estado
#define BLINK_GPIO GPIO_NUM_23

// Maximo numero de intentos en lora antes de pasarme a WIFI
#define maxIntentosFallidosLora 10
// Cantidad de envios exitosos por WIFI antes de volver a probar LORA
#define enviosWifiParaVolverALora 3

// Credenciales de LORA OTAA
// En Little-endian
#define APP_EUI                                         \
    {                                                   \
        0x11, 0x22, 0x11, 0x00, 0xDD, 0xCC, 0xAA, 0xFF \
    }

// En little endian tambien                             \
    {                                                   \
        0xB8, 0x6D, 0x04, 0xD0, 0x7E, 0xD5, 0xB3, 0x70 \
    }

// En big endian format
#define APP_KEY
    {
        0xE4, 0xB1, 0x78, 0x4C, 0x25, 0x88, 0x8B, 0x13, 0xED, 0x6D,
        0xF5, 0xA7, 0x92, 0x1F, 0x10, 0xB4 \
    }
```

8. Conclusiones y trabajo futuro

En esta sección se presentan las principales conclusiones obtenidas del trabajo realizado y se juntan los puntos propuestos a lo largo del informe como trabajo futuro. Primeramente sobre la tecnología LoRa, se puede concluir que tiene un buen funcionamiento en general pero para sacarle el máximo desempeño tiene que ser usada en espacios al aire libre o con pocas

edificaciones y paredes. Al momento de ingresar en el interior de la Facultad la conectividad continua funcionando pero se ve notoriamente reducida.

Los dispositivos esp32 utilizados en el prototipo tienen una muy buena versatilidad en su capacidad de entrada/salida (para la conexión de sensores) y más aún en cuanto a conectividad (LoRa, WiFi y Bluetooth), el único aspecto negativo que se detectó en estas placas Sparkfun ESP32 LoRa es que si bien tiene pines de entrada/salida analógica, algunos de estos pines usan registros de memoria que son usados por el módulo WiFi cuando está en funcionamiento [24]. Esto hace que sea difícil conectar y trabajar de forma amplia con sensores o dispositivos extra que hagan uso de los pines analógicos. Por último, sobre estos dispositivos se recalca que es muy sencillo comenzar a programarlos gracias a la herramienta de desarrollo PlatformIO mostrada en este proyecto.

Como trabajo futuro se citan varios puntos a desarrollar:

- **Calibración del sensor SGP30:** se considera necesario realizar más mediciones del dispositivo en entornos realistas junto con otros sensores más precisos para poder contrastar los valores y ver como debe ser la función de calibración del SGP30.
- **Análisis de los metadatos de Yeap**: Yeap al recibir envíos de datos almacena metadatos con parámetros físicos de la transmisión, se notó también que estos parámetros físicos, como por ejemplo la frecuencia, Spread Factor, entre otros van variando a lo largo de los envíos. Se considera importante poder analizar estos metadatos durante un largo período de tiempo en un lugar de poca señal para poder buscar si existen patrones en donde hayan parámetros físicos que funcionen mejor.
- **Buscar la forma de hacer que el sensor reciba mensajes de ack:** En la solución actual el dispositivo sensor no recibe mensajes de ack del gateway. Como se explicó en la máquina de estados del prototipo 3, una vez que la placa logra conectarse a la red LoRa considera que puede enviar mensajes por este medio para siempre. En esta solución el dispositivo no tiene forma de saber si por algún motivo sus mensajes ya no están llegando, sin embargo la configuración de mensajes ack de downlink podría solucionar esto.
- **Continuar mejorando los comandos que puede procesar el dispositivo:** Otro aspecto de la solución presentado en el prototipo 3 fue la interfaz de comandos. Los comandos implementados en el proyecto son escasos y pueden mejorarse en gran medida. Se intentó en particular hacer un comando que pueda cambiar la red WiFi de forma dinámica sin la necesidad de escribirlo en el archivo de credenciales y realizando el flash de la placa pero no se tuvo éxito.
- **Mejorar el uso de energía del dispositivo:** Este es un aspecto en el que no pudo ahondar en el proyecto. Al ser un dispositivo que posiblemente esté midiendo en algún lugar sin acceso a un tomacorrientes, va a tener que depender de una batería y por lo tanto el buen uso de la misma es fundamental. Como único punto a destacar relacionado a esto es que por medio de un aparato USB con display que mide la corriente se pudo ver que el dispositivo usando WiFi tiene un consumo que oscila entre los 0.03A y 0.09A mientras que con LoRa se mantiene en 0.01A realizando picos instantáneos de 0.04A cada cierto tiempo.
- **Almacenamiento local:** en caso de perder totalmente la conectividad puede resultar de utilidad poder almacenar los datos en una memoria conectada al dispositivo, por ejemplo una tarjeta micro SD o un USB de modo de no perder los datos de ese tiempo.

9. Anexos

9.1. Logs de las pruebas de conectividad realizadas en Facultad

TIMESTAMP	EVENT	APP	DEVEUI	TEXT
19/11/2021 17:41:17	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzUuMjAwMDAvMjIuNzAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvNS4wMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":113,"freq":920800000,"id":1637354475164,"live":true,"port":1,"rssi":-105,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":1,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:41:15.164Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:40:42	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzUuMDAwMDAvMjIuNzAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMi4wMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":111,"freq":916000000,"id":1637354440040,"live":true,"port":1,"rssi":-90,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":14,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:40:40.040Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:39:49	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzUuMDAwMDAvMjIuODAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMTYuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":108,"freq":915600000,"id":1637354387272,"live":true,"port":1,"rssi":-89,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:39:47.272Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:39:14	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzUuMDAwMDAvMjIuODAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMTAuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":106,"freq":921400000,"id":1637354352624,"live":true,"port":1,"rssi":-97,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":9,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:39:12.624Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:37:46	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzQuNzAwMDAvMjIuOTAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvNy4wMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":101,"freq":916600000,"id":1637354264715,"live":true,"port":1,"rssi":-93,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":13,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:37:44.715Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:36:00	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzQuNTAwMDAvMjMuMDAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMi4wMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":95,"freq":920200000,"id":1637354158720,"live":true,"port":1,"rssi":-93,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":12,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:35:58.720Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:34:48	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzQuMjAwMDAvMjMuMDAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvNi4wMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":91,"freq":920600000,"id":1637354088437,"live":true,"port":1,"rssi":-111,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":-7,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:34:48.437Z","altitude":86,"latitude":-34.90461,"longitude":-56.18216} RespBody: Envio exitoso
19/11/2021 17:34:32	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzQuMTAwMDAvMjMuMTAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMTMuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":90,"freq":916400000,"id":1637354070848,"live":true,"port":1,"rssi":-91,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":14,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:34:30.848Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:33:58	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzMuOTAwMDAvMjMuMTAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMTMuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":88,"freq":920000000,"id":1637354036800,"live":true,"port":1,"rssi":-93,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":12,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:33:56.800Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:32:47	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzMuNDAwMDAvMjMuMzAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvNS4wMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":84,"freq":921200000,"id":1637353965385,"live":true,"port":1,"rssi":-109,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":-2,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:32:45.385Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:32:12	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzMuNTAwMDAvMjMuNDAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMi4wMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":82,"freq":915200000,"id":1637353930204,"live":true,"port":1,"rssi":-90,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":14,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:32:10.204Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:29:13	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzUuNzAwMDAvMjIuNjAwMDAvNDEzLjAwMDAvNjAwMDAuMDAvMTYuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":71,"freq":915400000,"id":1637353751478,"live":true,"port":1,"rssi":-80,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:29:11.478Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:28:36	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzMuOTAwMDAvMjIuNjAwMDAvNDEwLjAwMDAvNjAwMDAuMDAvNDguMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":68,"freq":916200000,"id":1637353714044,"live":true,"port":1,"rssi":-79,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:28:34.044Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:28:00	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzAuMTAwMDAvMjMuNTAwMDAvNDE1LjAwMDAvNjAwMDAuMDAvMTYuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":66,"freq":915800000,"id":1637353678131,"live":true,"port":1,"rssi":-103,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":2,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:27:58.131Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:24:26	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NDUuNTAwMDAvLTQ2LjYwMDAvNDAwLjAwMDAvMC4wMDAwMDAvMTIuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":53,"freq":920000000,"id":1637353464696,"live":true,"port":1,"rssi":-96,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":8,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:24:24.696Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:22:39	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NDUuMjAwMDAvLTQ2LjYwMDAvNDAwLjAwMDAvMC4wMDAwMDAvMjYuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":47,"freq":916600000,"id":1637353358769,"live":true,"port":1,"rssi":-96,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":8,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:22:38.769Z","altitude":86,"latitude":-34.90461,"longitude":-56.18216} RespBody: Envio exitoso
19/11/2021 17:22:04	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NDUuMTAwMDAvLTQ2LjYwMDAvNDAwLjAwMDAvMC4wMDAwMDAvMC4wMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":45,"freq":921400000,"id":1637353323612,"live":true,"port":1,"rssi":-102,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":0,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:22:03.612Z","altitude":86,"latitude":-34.90258333333333,"longitude":-56.1784} RespBody: Envio exitoso
19/11/2021 17:18:35	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzAuMzAwMDAvMjMuNDAwMDAvNDAwLjAwMDAvMi4wMDAwMDAvMTIuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":33,"freq":920600000,"id":1637353113167,"live":true,"port":1,"rssi":-98,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":8,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:18:33.167Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:17:25	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NjguMTAwMDAvMjQuMDAwMDAvNDAwLjAwMDAvMC4wMDAwMDAvNDguMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":29,"freq":921200000,"id":1637353043925,"live":true,"port":1,"rssi":-104,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":2,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:17:23.925Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:17:06	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NjguMDAwMDAvMjQuMDAwMDAvNDAwLjAwMDAvMC4wMDAwMDAvMC4wMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":28,"freq":920800000,"id":1637353024827,"live":true,"port":1,"rssi":-105,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":1,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:17:04.827Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:16:31	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzAuMDAwMDAvMjQuMzAwMDAvNDAwLjAwMDAvMC4wMDAwMDAvNDIuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":26,"freq":920200000,"id":1637352989585,"live":true,"port":1,"rssi":-75,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:16:29.585Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:15:39	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzEuODAwMDAvMjQuMzAwMDAvNDAwLjAwMDAvMC4wMDAwMDAvOTEuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":23,"freq":915600000,"id":1637352937324,"live":true,"port":1,"rssi":-55,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:15:37.324Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:15:20	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzIuMjAwMDAvMjQuMjAwMDAvNDAwLjAwMDAvMC4wMDAwMDAvMTEuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":22,"freq":921000000,"id":1637352919236,"live":true,"port":1,"rssi":-94,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":8,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:15:19.236Z","altitude":86,"latitude":-34.90461,"longitude":-56.18216} RespBody: Envio exitoso
19/11/2021 17:14:10	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzEuNzAwMDAvMjMuOTAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMzAuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":18,"freq":916400000,"id":1637352848911,"live":true,"port":1,"rssi":-62,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:14:08.911Z","altitude":39,"latitude":-34.919629716430364,"longitude":-56.165672660550456} RespBody: Envio exitoso
19/11/2021 17:13:01	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NjkuODAwMDAvMjMuNjAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvNS4wMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":14,"freq":916200000,"id":1637352779120,"live":true,"port":1,"rssi":-81,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:12:59.120Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:12:41	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NjkuMjAwMDAvMjMuNjAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMTU1LjAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":12,"freq":915200000,"id":1637352759484,"live":true,"port":1,"rssi":-95,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":10,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:12:39.484Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:12:21	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NjkuMDAwMDAvMjMuNTAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMTkyLjAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":10,"freq":915400000,"id":1637352739829,"live":true,"port":1,"rssi":-107,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":0,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:12:19.829Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:12:03	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NjguNTAwMDAvMjMuNjAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMzkuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":9,"freq":920400000,"id":1637352721216,"live":true,"port":1,"rssi":-101,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":8,"snr":5,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:12:01.216Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:10:16	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NjcuNDAwMDAvMjQuMjAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMTEuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF10BW125","early":false,"fcnt":2,"freq":916000000,"id":1637352614043,"live":true,"port":1,"rssi":-72,"session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc","sf_used":10,"snr":15,"time_on_air_ms":698.368,"timestamp":"2021-11-19T20:10:14.043Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:10:14	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/joininfo => 404 ReqBody: {"appeui":"ffaaccdd00112211","deveui":"70b3d57ed0046db8","join_status":"JOIN_COMPLETE","session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/joininfo</pre> </body> </html>
19/11/2021 17:09:33	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/joininfo => 404 ReqBody: {"appeui":"ffaaccdd00112211","deveui":"70b3d57ed0046db8","join_status":"JOIN_ACCEPTED","session_id":"246ead71-3a0f-4a57-9273-a1ffb097c5cc"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/joininfo</pre> </body> </html>
19/11/2021 17:07:26	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NjkuNDAwMDAvMjMuNjAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMi4wMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":114,"freq":920400000,"id":1637352444924,"live":true,"port":1,"rssi":-70,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:07:24.924Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:07:08	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NjkuNzAwMDAvMjMuNTAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMzIuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":113,"freq":920000000,"id":1637352426794,"live":true,"port":1,"rssi":-69,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:07:06.794Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:06:51	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NjkuNzAwMDAvMjMuNDAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvNi4wMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":112,"freq":921400000,"id":1637352409215,"live":true,"port":1,"rssi":-73,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:06:49.215Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:04:48	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzMuMTAwMDAvMjIuODAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMS4wMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":105,"freq":915800000,"id":1637352286168,"live":true,"port":1,"rssi":-74,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:04:46.168Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:04:31	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzQuMDAwMDAvMjIuNjAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMC4wMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":104,"freq":916200000,"id":1637352269141,"live":true,"port":1,"rssi":-73,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:04:29.141Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:04:13	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzQuMjAwMDAvMjIuNTAwMDAvNDAwLjAwMDAvMC4wMDAwMDAvMC4wMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":103,"freq":915200000,"id":1637352251572,"live":true,"port":1,"rssi":-71,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:04:11.572Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:00:42	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"LTUuODAwMDAvLTQ2LjYwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMjMuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":91,"freq":915600000,"id":1637352040030,"live":true,"port":1,"rssi":-63,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":14,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:00:40.030Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 17:00:06	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzguNDAwMDAvMjEuMDAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvNDguMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":89,"freq":921200000,"id":1637352004859,"live":true,"port":1,"rssi":-71,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T20:00:04.859Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 16:58:38	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzcuODAwMDAvMjEuMTAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMTYuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":84,"freq":920800000,"id":1637351916977,"live":true,"port":1,"rssi":-78,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:58:36.977Z","altitude":39,"latitude":-34.92015063522618,"longitude":-56.16637093358999} RespBody: Envio exitoso
19/11/2021 16:58:17	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzcuOTAwMDAvMjEuMjAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvOTEuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":81,"freq":916000000,"id":1637351896201,"live":true,"port":1,"rssi":-63,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:58:16.201Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 16:57:40	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzcuODAwMDAvMjEuMzAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMC4wMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":78,"freq":920200000,"id":1637351858710,"live":true,"port":1,"rssi":-70,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:57:38.710Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 16:56:45	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzguMzAwMDAvMjEuNDAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvNDguMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":74,"freq":916600000,"id":1637351803781,"live":true,"port":1,"rssi":-63,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:56:43.781Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 16:56:45	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.91904,"longitude":-56.16632,"loraloc_fcnt":74,"loraloc_time":"2021-11-19T19:56:43.781Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:56:44	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/nodeinfo => 404 ReqBody: {"deveui":"70b3d57ed0046db8","device_class":0,"device_status":3,"dl_fcnt":7,"expiry_time_downlink":168,"expiry_time_uplink":168,"last_reception":"2021-11-19T19:55:48.200Z","registration_status":1} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/nodeinfo</pre> </body> </html>
19/11/2021 16:55:50	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.92000036429873,"longitude":-56.16763797207043,"loraloc_fcnt":70,"loraloc_time":"2021-11-19T19:55:48.200Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:55:48	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/nodeinfo => 404 ReqBody: {"deveui":"70b3d57ed0046db8","device_class":0,"device_status":3,"dl_fcnt":6,"expiry_time_downlink":168,"expiry_time_uplink":168,"last_reception":"2021-11-19T19:55:45.404Z","registration_status":1} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/nodeinfo</pre> </body> </html>
19/11/2021 16:55:46	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/nodeinfo => 404 ReqBody: {"deveui":"70b3d57ed0046db8","device_class":0,"device_status":3,"dl_fcnt":5,"expiry_time_downlink":168,"expiry_time_uplink":168,"last_reception":"2021-11-19T19:55:45.404Z","registration_status":1} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/nodeinfo</pre> </body> </html>
19/11/2021 16:55:45	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzguMzAwMDAvMjEuNTAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMjguMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":69,"freq":916400000,"id":1637351745404,"live":true,"port":1,"rssi":-110,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":-5,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:55:45.404Z","altitude":86,"latitude":-34.90461,"longitude":-56.18216} RespBody: Envio exitoso
19/11/2021 16:55:45	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.90461,"longitude":-56.18216,"loraloc_fcnt":69,"loraloc_time":"2021-11-19T19:55:45.404Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:55:07	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzguMTAwMDAvMjEuNjAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMjAuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":66,"freq":921000000,"id":1637351707264,"live":true,"port":1,"rssi":-110,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":-2,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:55:07.264Z"} RespBody: Envio exitoso
19/11/2021 16:55:07	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/nodeinfo => 404 ReqBody: {"deveui":"70b3d57ed0046db8","device_class":0,"device_status":3,"dl_fcnt":4,"expiry_time_downlink":168,"expiry_time_uplink":168,"last_reception":"2021-11-19T19:52:29.114Z","registration_status":1} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/nodeinfo</pre> </body> </html>
19/11/2021 16:52:31	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzcuNTAwMDAvMjEuNzAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMzcuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":57,"freq":915600000,"id":1637351549114,"live":true,"port":1,"rssi":-54,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:52:29.114Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 16:52:31	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.91904,"longitude":-56.16632,"loraloc_fcnt":57,"loraloc_time":"2021-11-19T19:52:29.114Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:51:36	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzcuMTAwMDAvMjEuODAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMjIuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":54,"freq":916600000,"id":1637351496274,"live":true,"port":1,"rssi":-108,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":-6,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:51:36.274Z","altitude":86,"latitude":-34.90461,"longitude":-56.18216} RespBody: Envio exitoso
19/11/2021 16:51:36	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.90461,"longitude":-56.18216,"loraloc_fcnt":54,"loraloc_time":"2021-11-19T19:51:36.274Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:51:04	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzYuNTAwMDAvMjIuMjAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvNzIuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":52,"freq":916200000,"id":1637351462136,"live":true,"port":1,"rssi":-43,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:51:02.136Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 16:51:04	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.91904,"longitude":-56.16632,"loraloc_fcnt":52,"loraloc_time":"2021-11-19T19:51:02.136Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:50:28	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzYuNDAwMDAvMjIuMjAwMDAvNDAwLjAwMDAvNjAwMDAuMDAvMzIuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":50,"freq":920200000,"id":1637351426921,"live":true,"port":1,"rssi":-60,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:50:26.921Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 16:50:28	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.91904,"longitude":-56.16632,"loraloc_fcnt":50,"loraloc_time":"2021-11-19T19:50:26.921Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:48:24	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"LTUuODAwMDAvLTQ2LjYwMDAvNDAwLjAwMDAvMi4wMDAwMDAvNjkuMDAwMDAvMDEvMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":43,"freq":920400000,"id":1637351302938,"live":true,"port":1,"rssi":-59,"session_id":"51a830d0-c6b8-47e4-ab1a-9d686009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:48:22.938Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso

19/11/2021 16:48:24	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.91904,"longitude":-56.16632,"loraloc_fcnt":43,"loraloc_time":"2021-11-19T19:48:22.938Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:46:21	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzQxNDAwMDAxMjIaMTAwMDAxNDEzLjAwMDAvMi4wMDAwMDAvODYuMDAwMDAvMDExMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":36,"freq":921200000,"id":1637351179905,"live":true,"port":1,"rssi":-70,"session_id":"51a830d0-c6b8-47e4-ab1a-9d86009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:46:19.905Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 16:46:21	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.91904,"longitude":-56.16632,"loraloc_fcnt":36,"loraloc_time":"2021-11-19T19:46:19.905Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:45:11	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.91969011267606,"longitude":-56.16634981084505,"loraloc_fcnt":32,"loraloc_time":"2021-11-19T19:45:09.587Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:45:11	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"1TIUdDAwMDAxLTQ2LjYwMDAvNDAwLjAwMDAvMC4wMDAwMDAvMTkuMDAwMDAvMDExMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":32,"freq":916400000,"id":1637351109587,"live":true,"port":1,"rssi":-51,"session_id":"51a830d0-c6b8-47e4-ab1a-9d86009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:45:09.587Z","altitude":39,"latitude":-34.91969011267606,"longitude":-56.16634981084505} RespBody: Envio exitoso
19/11/2021 16:43:43	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.91904,"longitude":-56.16632,"loraloc_fcnt":27,"loraloc_time":"2021-11-19T19:43:41.710Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:43:43	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NjkuMTAwMDAxMjMuNzAwMDAxNDAwLjAwMDAvMC4wMDAwMDAvMTYuMDAwMDAvMDExMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":27,"freq":915200000,"id":1637351021710,"live":true,"port":1,"rssi":-65,"session_id":"51a830d0-c6b8-47e4-ab1a-9d86009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:43:41.710Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 16:42:50	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.91682244940539,"longitude":-56.17150409972877,"loraloc_fcnt":24,"loraloc_time":"2021-11-19T19:42:48.942Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:42:50	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzAxMTAwMDAxMjQuOTAwMDAxNDAwLjAwMDAvMC4wMDAwMDAvMzMuMDAwMDAvMDExMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":24,"freq":915400000,"id":1637350968942,"live":true,"port":1,"rssi":-53,"session_id":"51a830d0-c6b8-47e4-ab1a-9d86009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:42:48.942Z","altitude":39,"latitude":-34.91682244940539,"longitude":-56.17150409972877} RespBody: Envio exitoso
19/11/2021 16:41:41	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.91904,"longitude":-56.16632,"loraloc_fcnt":20,"loraloc_time":"2021-11-19T19:41:39.148Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:41:41	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzAuMjAxMDAxMjQuODAwMDAxNDAwLjAwMDAvMC4wMDAwMDAvMTYuMDAwMDAvMDExMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":20,"freq":915800000,"id":1637350899148,"live":true,"port":1,"rssi":-53,"session_id":"51a830d0-c6b8-47e4-ab1a-9d86009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:41:39.148Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 16:40:28	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzAuMjAxMDAxMjUuNTAwMDAxNDAwLjAwMDAvMTMwLjAwMDAvMTgsMDAwMDAvMDExMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":15,"freq":921400000,"id":1637350826747,"live":true,"port":1,"rssi":-82,"session_id":"51a830d0-c6b8-47e4-ab1a-9d86009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:40:26.747Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 16:40:28	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.91904,"longitude":-56.16632,"loraloc_fcnt":15,"loraloc_time":"2021-11-19T19:40:26.747Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:40:27	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/nodeinfo => 404 ReqBody: {"deveui":"70b3d57ed0046db8","device_class":0,"device_status":3,"dl_fcnt":3,"expiry_time_downlink":168,"expiry_time_uplink":168,"last_reception":"2021-11-19T19:40:06.400Z","registration_status":1} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/nodeinfo</pre> </body> </html>
19/11/2021 16:40:06	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/nodeinfo => 404 ReqBody: {"deveui":"70b3d57ed0046db8","device_class":0,"device_status":3,"dl_fcnt":2,"expiry_time_downlink":168,"expiry_time_uplink":168,"last_reception":"2021-11-19T19:40:06.400Z","registration_status":1} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/nodeinfo</pre> </body> </html>
19/11/2021 16:40:06	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.91904,"longitude":-56.16632,"loraloc_fcnt":13,"loraloc_time":"2021-11-19T19:40:06.400Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:40:06	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzEuNzAwMDAxMjUuMzAwMDAxNDAwLjAwMDAvMTMwLjAwMDAvMC4wMDAwMDAvMTMwLjAwMDAvMDExMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":13,"freq":920600000,"id":1637350806400,"live":true,"port":1,"rssi":-97,"session_id":"51a830d0-c6b8-47e4-ab1a-9d86009084a","sf_used":8,"snr":-2,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:40:06.400Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 16:39:31	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.91904,"longitude":-56.16632,"loraloc_fcnt":10,"loraloc_time":"2021-11-19T19:39:29.466Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:39:31	InfoPush	fing_uy	70b3d57ed0046db8	POST http://159.122.175.131:30500/newdata/rest/callback/payloads/ul => 200 ReqBody: {"confirmed":false,"cr_used":"4/5","dataFrame":"NzcuNDAwMDAxMjQuNjAwMDAxNDAwLjAwMDAvMS4wMDAwMDAvNjQuMDAwMDAvMDExMA==","data_format":"base64","decrypted":true,"devaddr":3802758,"deveui":"70b3d57ed0046db8","device_redundancy":1,"dr_used":"SF8BW125","early":false,"fcnt":10,"freq":916000000,"id":1637350769466,"live":true,"port":1,"rssi":-86,"session_id":"51a830d0-c6b8-47e4-ab1a-9d86009084a","sf_used":8,"snr":15,"time_on_air_ms":205.312,"timestamp":"2021-11-19T19:39:29.466Z","altitude":39,"latitude":-34.91904,"longitude":-56.16632} RespBody: Envio exitoso
19/11/2021 16:39:30	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/nodeinfo => 404 ReqBody: {"deveui":"70b3d57ed0046db8","device_class":0,"device_status":3,"dl_fcnt":1,"expiry_time_downlink":168,"expiry_time_uplink":168,"last_reception":"2021-11-19T19:38:33.918Z","registration_status":1} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/nodeinfo</pre> </body> </html>
19/11/2021 16:38:34	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/nodeinfo => 404 ReqBody: {"deveui":"70b3d57ed0046db8","device_class":0,"device_status":3,"dl_fcnt":0,"expiry_time_downlink":168,"expiry_time_uplink":168,"last_reception":"2021-11-19T19:38:33.918Z","registration_status":1} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/nodeinfo</pre> </body> </html>
19/11/2021 16:38:34	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.90461,"longitude":-56.18216,"loraloc_fcnt":6,"loraloc_time":"2021-11-19T19:38:33.918Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:38:33	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/joininfo => 404 ReqBody: {"appeui":"ffaaccdd00112211","deveui":"70b3d57ed0046db8","join_status":"JOIN_COMPLETE","session_id":"51a830d0-c6b8-47e4-ab1a-9d86009084a"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/joininfo</pre> </body> </html>
19/11/2021 16:36:40	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/joininfo => 404 ReqBody: {"appeui":"ffaaccdd00112211","deveui":"70b3d57ed0046db8","join_status":"JOIN_ACCEPTED","session_id":"51a830d0-c6b8-47e4-ab1a-9d86009084a"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/joininfo</pre> </body> </html>
19/11/2021 16:34:59	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.91751941935484,"longitude":-56.16798916129032,"loraloc_fcnt":19,"loraloc_time":"2021-11-19T19:34:58.172Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:33:22	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.91904,"longitude":-56.16632,"loraloc_fcnt":16,"loraloc_time":"2021-11-19T19:33:20.874Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:31:42	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/nodeinfo => 404 ReqBody: {"deveui":"70b3d57ed0046db8","device_class":0,"device_status":3,"dl_fcnt":4,"expiry_time_downlink":168,"expiry_time_uplink":168,"last_reception":"2021-11-19T19:31:42.441Z","registration_status":1} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/nodeinfo</pre> </body> </html>
19/11/2021 16:31:42	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.9050925,"longitude":-56.1880375,"loraloc_fcnt":13,"loraloc_time":"2021-11-19T19:31:42.441Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:30:36	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.91904,"longitude":-56.16632,"loraloc_fcnt":10,"loraloc_time":"2021-11-19T19:30:35.679Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:30:36	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/nodeinfo => 404 ReqBody: {"deveui":"70b3d57ed0046db8","device_class":0,"device_status":3,"dl_fcnt":3,"expiry_time_downlink":168,"expiry_time_uplink":168,"last_reception":"2021-11-19T19:28:23.089Z","registration_status":1} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/nodeinfo</pre> </body> </html>
19/11/2021 16:28:25	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/location => 404 ReqBody: {"deveui":"70b3d57ed0046db8","latitude":-34.91904,"longitude":-56.16632,"loraloc_fcnt":5,"loraloc_time":"2021-11-19T19:28:23.089Z"} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/location</pre> </body> </html>
19/11/2021 16:28:23	ErrorPush	fing_uy	70b3d57ed0046db8	PUT http://159.122.175.131:30500/newdata/rest/callback/nodeinfo => 404 ReqBody: {"deveui":"70b3d57ed0046db8","device_class":0,"device_status":3,"dl_fcnt":2,"expiry_time_downlink":168,"expiry_time_uplink":168,"last_reception":"2021-11-19T19:27:48.292Z","registration_status":1} RespBody: <!DOCTYPE html> <html lang="en"> <head> <meta charset="utf-8"> <title>Error</title> </head> <body> <pre>Cannot PUT /newdata/rest/callback/nodeinfo</pre> </body> </html>

Referencias

[1] Sparkfun ESP32 LoRa 1-Channel Gateway

[2] SparkFun Humidity and Temperature Sensor Breakout - Si7021

[3] SparkFun Sound Detector

[4] SparkFun Air Quality Sensor - SGP30

[5] ESP32-WROOM-32 Datasheet

[6] Arduino-LMIC library ("MCCI LoRaWAN LMIC Library")

[7] ¿Qué es el Internet de las cosas (IoT)?

[8] ¿Qué es la tecnología LoRa y por qué es importante para IoT?

[9] Web de FreeRTOS

[10] IBM Cloudant página principal

[11] Documentación de la API de Cloudant en NodeJS

[12] The Things Network página principal

[13] Plataforma Orbiwise página principal

[14] PlatformIO página principal

[15] Video tutorial de setup de conexión de dispositivo a TTN usando ABP

[16] Guía oficial The Things Kickstarter Gateway

[17] Información Útil para debug, TTN Gateway - FAQ

[18] Drivers USB para Windows

[19] Documentación IBM Cloud Cli usada para el despliegue en Kubernetes

[20] Documentación de instalación de plug-ins para IBM Cloud Cli

[21] Lista de reproducción del canal Digi-Key sobre FreeRTOS

[22] Documentación de FreeRTOS: API Reference

[23] Documentación de FreeRTOS: Tasks

[24] Github: Sobre pines analógicos ADC2 y el uso de WiFi