

Domótica Agraria

*De bajo coste*

Grado en Ingeniería Informática



Trabajo Fin de Grado

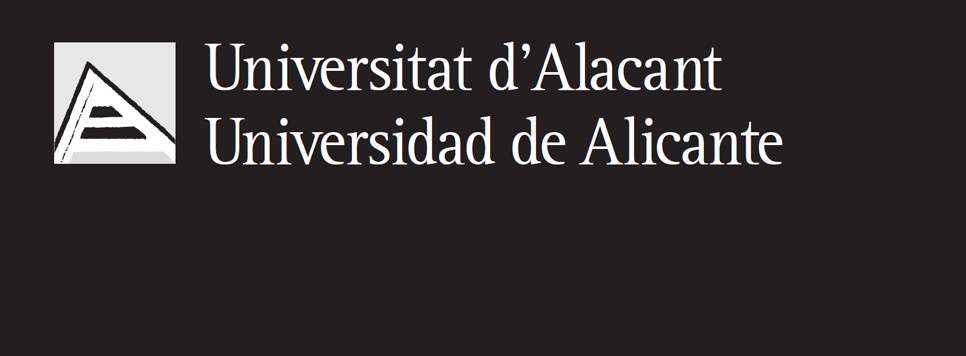
Autor:

Juan Carlos Candela Bordera

Tutor/es:

Francisco Javier Ferrández Pastor

Junio 2020



# 1. Justificación y objetivos

En un mundo cada vez más poblado y con unos recursos hídricos limitados, es necesario desarrollar tecnologías que proporcionen mayor producción de alimentos, optimizando el consumo de agua y de electricidad, con un coste asequible para el usuario.

En la actualidad existen diversas tecnologías que ayudan a optimizar el uso hídrico y energético en el campo, pero estos dispositivos no son asequibles para los agricultores de pequeña-media escala.

La idea de este proyecto es diseñar un proyecto de código libre que pueda ser construido y utilizado por usuarios de todo el mundo, el acceso a los recursos de este proyecto será público y gratuito, permitiendo su mejora en todo momento.

# 2. Agradecimientos

Agradezco en primer lugar al tutor, por haberme apoyado y guiado en el proceso de desarrollo de este trabajo. Así como a todas las personas que han acompañado en este trayecto llamado universidad, ayudándome y apoyándome para que disfrute y me dedique a mi gran pasión, el desarrollo software.

También agradecer a mi familia, la esperanza depositada en mí y todo el esfuerzo que han hecho para que haya llegado hasta dónde estoy hoy.

# 3. Índices

## 3.1 Índice de contenidos

[1. Justificación y objetivos 1](#_Toc41589182)

[2. Agradecimientos 1](#_Toc41589183)

[3. Índices 2](#_Toc41589184)

[3.1 Índice de contenidos 2](#_Toc41589185)

[3.2 Índice de Figuras 4](#_Toc41589186)

[4. Introducción 4](#_Toc41589187)

[5. Marco teórico o estado del arte 5](#_Toc41589188)

[5.1 Estado actual 5](#_Toc41589189)

[5.2 Dispositivos IoT 5](#_Toc41589190)

[Raspberry Pi 5](#_Toc41589191)

[Sistemas comerciales 6](#_Toc41589192)

[5.3 Tecnologías de comunicación 7](#_Toc41589193)

[SigFox 7](#_Toc41589194)

[NB-IoT 8](#_Toc41589195)

[GSM 9](#_Toc41589196)

[6. Objetivos 10](#_Toc41589197)

[7. Metodología 10](#_Toc41589198)

[7.1 Metodología de desarrollo de software 10](#_Toc41589199)

[7.2 Herramientas y tecnologías empleadas 11](#_Toc41589200)

[LoRa 12](#_Toc41589201)

[Arduino 13](#_Toc41589202)

[Adobe Fireworks 13](#_Toc41589203)

[Github 14](#_Toc41589204)

[Trello 15](#_Toc41589205)

[Visual Studio Code 15](#_Toc41589206)

[7.3 Planificación 16](#_Toc41589207)

[7.4 Repositorio y control de versiones 16](#_Toc41589208)

[8. Cuerpo del trabajo 16](#_Toc41589209)

[8.1 Análisis funcional 16](#_Toc41589210)

[8.2 Prototipo y mockups 16](#_Toc41589211)

[8.3 Arquitectura técnica 16](#_Toc41589212)

[8.4 Librerías utilizadas 19](#_Toc41589213)

[8.5 Implementación 19](#_Toc41589214)

[8.4.1 Configuración Gateway 19](#_Toc41589215)

[8.4.2 Registro Gateway en TTN 21](#_Toc41589216)

[8.4.3 Creación Nodo Emisor 25](#_Toc41589217)

[9. Validación 31](#_Toc41589218)

[10. Conclusiones 31](#_Toc41589219)

[10.1 Mejoras para implementar en un futuro 31](#_Toc41589220)

[11. Listado de acrónimos 31](#_Toc41589221)

[12. Biografía y referencias 31](#_Toc41589222)

## 3.2 Índice de Figuras

Todo: enlazar aquí las figuras del trabajo

## 4. Introducción

El crecimiento demográfico mundial no ha dejado de crecer en los últimos años, en la actualidad somos unos 7750 millones de individuos en el planeta Tierra y se espera que para 2030, este número llegue a los 8500 millones. Pese al crecimiento demográfico, los recursos como el agua o la comida siguen siendo limitados pero necesarios para sustentar a la población actual. Los recursos energéticos también están asociados a la producción de comida, ya que, con estos, se puede extraer el agua o se puede controlar la temperatura en la que se desarrollan las plantas.

En los últimos años ha aparecido tecnología que permite controlar el uso del agua y de la electricidad, para obtener el máximo rendimiento en los cultivos. Estás tecnologías están pensadas para grandes productores de alimentos, ya que este tipo de tecnologías no son rentables para los pequeños agricultores, porque tardan mucho en amortizar el coste de estas.

La finalidad principal del proyecto es hacer accesible esta tecnología a cualquier agricultor, para mejorar la producción de sus cultivos, optimizando recursos eléctricos e hídricos.

# 5. Marco teórico o estado del arte

## 5.1 Estado actual

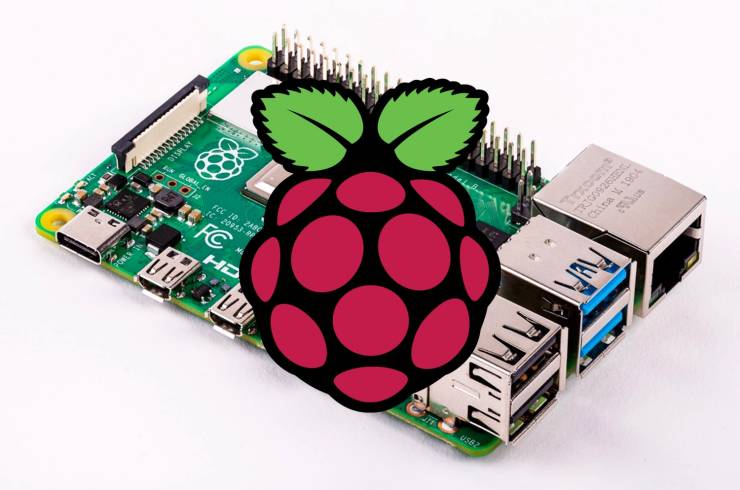
Actualmente hay una gran variedad de sistemas inalámbricos para la agricultura, pero la mayoría de ellos están enfocados a grandes agricultores para garantizar la máxima productividad en sus cultivos. Estos sistemas están construidos por dispositivos IoT, de los que existen infinidad de marcas y modelos, por lo que a continuación nombraré los más utilizados en este campo.

Además, para la comunicación de datos, hay diferentes tecnologías de comunicación, todas ellas con ciertas ventajas o desventajas que también voy a analizar.

## 5.2 Dispositivos IoT

En el mercado existen multitud de dispositivos IoT que se pueden utilizados para recoger datos de diferentes fuentes y luego poder tratarlos. En nuestro caso hemos elegido Arduino, los que se detallan a continuación también se pueden utilizar para nuestro objetivo, pero tienen una serie de desventajas que voy a mencionar

### 5.2.1 Raspberry Pi



Raspberry Pi es un ordenador de placa simple y bajo coste desarrollado en Reino Unido por la Fundación Raspberry Pi. Es lo suficientemente potente como para facilitar el aprendizaje y realizar tareas básicas, y también permite programar y compilar programas que se ejecuten en él.

Este tamaño reducido y la posibilidad de conectarle varios tipos de accesorio le dan una versatilidad que permite utilizarse para varios tipos de tareas. Su misión principal es la de enseñar informática en las aulas. Pero también sirve para utilizarse como un pequeño ordenador doméstico básico, y la comunidad de creadores también la usa como motor para varios tipos de proyecto.

En este proyecto se podría haber utilizado una Raspberry Pi para el desarrollo. Ya que muchos de los módulos que permiten la comunicación LoRa en Arduino están también esta placa. El problema radica en que el coste de este suele ser superior a las placas Arduino, ya que también es mayor el alcance de los proyectos que se pueden realizar. El objetivo actual es reducir los costes del sistema, y mientras la Raspberry Pi más económica del mercado para este proyecto, pero con más limitaciones técnicas; la Raspberry Pi Zero, tiene un coste de unos 5€, su contraparte en Arduino, se puede encontrar una placa Arduino UNO R3 con un coste de 3€, con unas especificaciones altas para proyectos embebidos.

### 5.2.2 Sistemas comerciales



Los sistemas comerciales son dispositivos embebidos que son construidos con el objetivo de realizar un fin en concreto, se pueden encontrar en muchas tiendas de Internet y hay diversos proveedores. Estos sistemas se encuentran en diferentes rangos de precios dependiendo de las funcionalidades o calidad que busque el cliente.

Por norma general estos vienen programados desde el principio por lo que el usuario no tiene que preocuparse de hacer grandes configuraciones. Esto genera que el usuario sea dependiente de la tecnología que ha comprado, ya que para poder trabajar con los datos que recogen estos sistemas, está obligado a adquirir otros dispositivos compatibles con éstos. Además, la lógica detrás de estos dispositivos está oculta al usuario por lo que éste no puede hacer modificaciones ni saber el funcionamiento detrás de ellos.

Se da el caso de que algunos sistemas comerciales para recoger datos son más económicos porque su coste está optimizado para realizar una tarea en concreto; recoger valores de humedad, temperatura, velocidad del viento. Al estar limitados para realizar ciertas tareas, es difícil poder añadir nuevas características o funcionalidades, ya que la arquitectura no lo permite. Además, al ser dependiente de la tecnología del fabricante para tratar los datos obtenidos, el coste de obtener el conjunto final de dispositivos es más elevado, ya que estás obligado a comprar todos los componentes relacionados con el producto comercial, muchas veces con sobrecostes.

## 5.3 Tecnologías de comunicación

Para el desarrollo del proyecto se ha elegido LoRa como tecnología de comunicación para nuestras sondas, pero hay otras alternativas que detallaré a continuación.

### 5.3.1 SigFox



SigFox es una compañía francesa fundada en 2009 que proporciona el servidio de red de cobertura amplia de bajo consumo LPWAN(Low Power Wide Area Network), es inalámbrica y fue creada para que funcione e interactúe con dispositivos de bajo consumo energético, con tasas de transferencias de datos de hasta 12 bytes.

La red funciona con la tecnología UNB (Ultra Narro Band) y consiste en emplear canales estrechos del espectro para alcanzar grandes distancias con un requerimiento mínimo de energía. Dispone de cobertura en 30 países y para la comunicación se emplea la frecuencia 868 MHz en Europa y 902 MHz en Estados Unidos. Estas frecuencias de radio se encuentran dentro de las bandas sin licencia, que pueden ser utilizadas para usos múltiples excepto para telecomunicaciones. [1]

La desventaja de esta tecnología de comunicación es que se trata de un protocolo propietario dependiente de la empresa SigFox, por lo que para poder utilizar esta tecnología es necesario adquirir los servicios de esta compañía. Otro problema añadido es que la red no se encuentra en todos los países, se encuentra en países con tecnologías inalámbricas desarrolladas.

Este protocolo no es el adecuado para este proyecto por que se busca abaratar al máximo el coste y que pueda ser utilizado en el mayor número de zonas posible.

### 5.3.2 NB-IoT



Narrowband-IoT es una LPWAN(Low Power Wide Area Network), es un estándar de tecnología de radio desarrollado por 3GPP. Esta tecnología hace uso de bandas de espectro licenciadas, por lo que no existen intereferencias con otras tecnologías, haciendo que la robustez de las comunicaciones sea mucho mayor y no dependa del número de dispositivos en las proximidades. Las bandas de frecuencia reservadas para NB-IoT están en el rango de los 700 y 800 Mhz, lo que facilita la penetración en edificios y sótanos, haciendo que la tecnología se pueda operar en un mayor número de escenarios que las redes WPA y LPWA[2]. Hace uso de las redes telefónicas para operar.

Los inconvenientes es que se trata de una tecnología propietaria y para poder acceder a la red, se necesita acceso privado o contratar servicios a distintas empresas de telecomunicación, motivo por el cual se ha descartado el uso de esta tecnología.

### 5.3.4 GSM



El sistema global para las comunicaciones móviles o GSM, se trata de un estándar muy utilizado desde principios de siglo y también se conoce como 2G debido a que supuso un salto de las comunicaciones analógicas a las digitales. La banda de frecuencia en la que opera el GSM difiere según el territorio. En Europa se utiliza el espectro radioeléctrico de 900 y 1800 Mhz. Esto hace que no todos los dispositivos GSM puedan funcionar en todo el mundo, a no ser que su tecnología esté preparada para conectarse a todas las bandas. [3]

Un cliente GSM puede conectarse a través de su teléfono con su computador y enviar y recibir mensajes por [correo electrónico](https://es.wikipedia.org/wiki/Correo_electr%C3%B3nico), [faxes](https://es.wikipedia.org/wiki/Fax), navegar por [Internet](https://es.wikipedia.org/wiki/Internet), acceder con seguridad a la red informática de una compañía ([red local](https://es.wikipedia.org/wiki/Red_de_%C3%A1rea_local)/[Intranet](https://es.wikipedia.org/wiki/Intranet)), así como utilizar otras funciones digitales de transmisión de datos, incluyendo el [servicio de mensajes cortos](https://es.wikipedia.org/wiki/Servicio_de_mensajes_cortos) (SMS) o mensajes de texto.

Pese a ser un sistema de comunicación ampliamente utilizado en comunicaciones IoT, tiene la desventaja de que por cada dispositivo es necesario una tarjeta SIM para comunicarse con la red. Estas tarjetas tienen asociado un coste, tanto al adquirirlas como al hacer uso de ello, y queremos evitar costos adicionales en la instalación.

# 6. Objetivos

Como he explicado anteriormente, la finalidad del proyecto es construir un sistema asequible y sencillo para que gente con pocos recursos o sin conocimiento pueda hacer uso de las ventajas que supone la tecnología en el mundo agrícola. Para conseguirlo, me he preparado una serie de objetivos:

* Creación y configuración de un Gateway
* Creación y configuración de los sensores
* Conectar dispositivos a la red TTN
* Recoger los datos que recibe la red TTN
* Creación de aplicación para interpretar los datos obtenidos por los sensores

# 7. Metodología

## 7.1 Metodología de desarrollo de software

Para el desarrollo del proyecto, se ha utilizado SCRUM, es un marco de trabajo para desarrollo ágil. SCRUM define un conjunto de prácticas y roles, que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante el proyecto. La metodología está pensada para ser utilizada por un equipo de varias personas, pero en este caso al tratarse de una persona, se ha adaptado la metodología para ello. Los puntos que se han utilizado son:

* Desarrollo incremental de los requisitos del proyecto en bloques temporales cortos y fijos.
* Prioridad a lo que tiene más valor
* Sincronización diaria y adaptaciones necesarias
* Tras cada iteración, se hace retrospectiva de los resultados reales obtenidos para tomar decisiones en la siguiente iteración
* Fijado de tiempos máximos para lograr objetivos

Al estar pensado para equipos pequeños de personas con varios roles, se han unificado para crear un único rol. Los roles existentes son los siguientes:

- **Product Owner**: se asegura de que el equipo de Scrum trabaje de forma adecuada desde la perspectiva de negocio. Se encarga de escribir las historias de usuario, priorizándolas y colocándolas en el Product Backlog.

Las historias de usuario es la representación de un requisito escrito en una o dos frases utilizando el lenguaje común del usuario. Son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos.

El Product Backlog o pila de producto es un listado de todas las tareas o historias de usuario que se pretenden hacer durante el desarrollo de un proyecto. Todas las tareas deben listarse en esta lista, para que estén visibles ante todo el equipo y se pueda tener una visión panorámica de todo lo que se espera realizar.

- **Scrum Master**: cuyo trabajo es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint o iteración. No es el líder del equipo, sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. Se asegura de que todo el proceso de Scrum se utiliza correctamente y también es el encargado de hacer las reglas que se tienen que cumplir por el resto de las personas que forman parte del equipo.

- **Equipo de desarrollo**: es el equipo encargado de entregar el producto tras cada una de las iteraciones. Es recomendable un pequeño equipo de 3 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (análisis, diseño, desarrollo, pruebas, documentación, etc.).

## 7.2 Herramientas y tecnologías empleadas

Para el desarrollo del proyecto, se han utilizado las siguientes herramientas y tecnologías:

* Protocolo de comunicación: Lora
* Hardware: Arduino
* Diseño: Adobe Fireworks
* Software: Github, Trello, Visual Studio Code,

### 7.2.1 LoRa



LoRa o LoRaWAN es una especificación para redes de baja potencia y área amplia, diseñada específicamente para dispositivos de bajo consumo de alimentación, que operan en redes de alcance local, regional, nacionales o globales.

El estándar de red de LoRa, apunta a requerimientos característicos de IoT, tales como conexiones bidireccionalmente seguras, bajo consumo de energía, largo alcance de comunicación, bajas velocidades de datos, baja frecuencia de transmisión, movilidad y servicios de localización. Permite la interconexión entre objetos inteligentes sin la necesidad de instalaciones complejas y además otorga amplia libertad de uso al usuario final, al desarrollador y a las empresas que quieran instalar su propia red, ya que es de acceso libre.

Las redes LoRa están formadas por dos tipos de dispositivos, los dispositivos finales y las puertas de enlace o gateways. Las puertas de enlace se encargan de recibir todos los mensajes de los dispositivos finales, y a partir de esta, se pueden enviar a través de Internet para su posterior tratamiento.

La comunicación entre dispositivos finales y las puertas de enlace, se hacen en diferentes canales de frecuencia y a distintas velocidades de datos. La selección de velocidad de datos depende de la distancia de alcance, la duración y el consumo de energía del mensaje. Además la comunicación entre dispositivos puede ser unidireccional o bidireccional, el estándar soporta multidifusión, permitiendo actualización de software en forma inalámbrica u otras formas de distribución de mensajes en masa.

### 7.2.2 Arduino

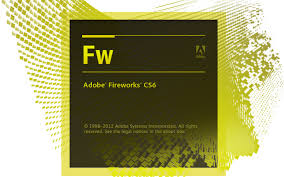


Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso. El hardware libre son los dispositivos cuyas especificaciones y diagramas son de acceso público, de manera que cualquiera puede replicarlos. De esta forma, cualquier entidad puede crear sus propias placas, pudiendo ser diferentes entre ellas pero igualmente funcionales al partir de la misma base.[4]

Esta base es sobre la que están montados los dispositivos que vamos a utilizar en el proyecto, llamados **TTGO ESP32** que voy a explicar más adelante.

Otra de las ventajas de hacer uso de placas basadas en Arduino es el gran número de información disponible que hay en Internet, ya que se pueden encontrar infinidad de proyectos que se pueden construir a partir de él. También ese conocido por su comunidad, dónde la mayoría de usuarios se ayudan entre ellos para construir sus propios dispositivos.

### 7.2.3 Adobe Fireworks



Fireworks es un software de edición de mapas de bits y de gráficos vectoriales que forma parte de Adobe Creative Suite, el cual está integrado por diferentes aplicaciones y herramientas focalizadas principalmente en la publicación web, impresa y postproducción de video.

Lo he utilizado principalmente para hacer los prototipos de la aplicación y para los gráficos de la arquitectura. A ser un editor de gráficos vectoriales, es muy rápido para realizar tareas que supongan la reutilización de imágenes en todo momento.

### 7.2.4 Github



Es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.

En el proyecto se ha utilizado como control de versiones en los distintos desarrollos de código que lo componen. También lo he utilizado como copia de seguridad, guardado cada día todo lo que iba incluyendo a la carpeta del proyecto.

Para almacenar todos los directorios, se ha creado un repositorio llamado TFG, con las diferentes carpetas y archivos que lo componen.

### 7.2.5 Trello



Trello es un software de administración de proyectos con interfaz web y con cliente para dispositivos móviles. Emplea el sistema Kanban para el registro de actividades mediante tarjetas virtuales. Se utilizan una listas que representan estados por donde van pasando las distintas actividades.

Su uso ha sido principalmente para la planificación, es una forma muy gráfica de organizar el estado de las tareas, desde las que están terminadas, en desarrollo o las que todavía están por empezar. Más adelante explicaré su función en el apartado de Planificación.

### 7.2.6 Visual Studio Code



Es un editor de código fuente multiplataforma desarrollado por Microsoft. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, y refactorización Actualmente es uno de los más utilizados por su sencillez y la gran cantidad de extensiones que ayudan al desarrollo software.

Se usado principalmente para el desarrollo de la aplicación y el servidor, que más adelante explicaré.

## 7.3 Planificación

Antes de realizar el desarrollo del proyecto, es importante planificarlo previamente. La planificación consiste en la ordenación sistemática de tareas para lograr un objetivo, exponiendo lo que se necesita hacer y cómo debe llevarse a cabo.

El trabajo se ha dividido en pequeñas tareas para poder visualizar todo el trabajo pendiente por hacer, el que está en desarrollo, el trabajo en fase de testo y el que está terminado.

Para la gestión de las tareas, se ha hecho uso de la aplicación web Trello, comentada anteriormente. Dentro de Trello se ha creado un tablero para organizar las tareas que componen el proyecto.

Dentro del tablero se han creado cuatro columnas:

## 7.4 Repositorio y control de versiones

Todo: explicar el uso de control de versiones en Github

# 8. Cuerpo del trabajo

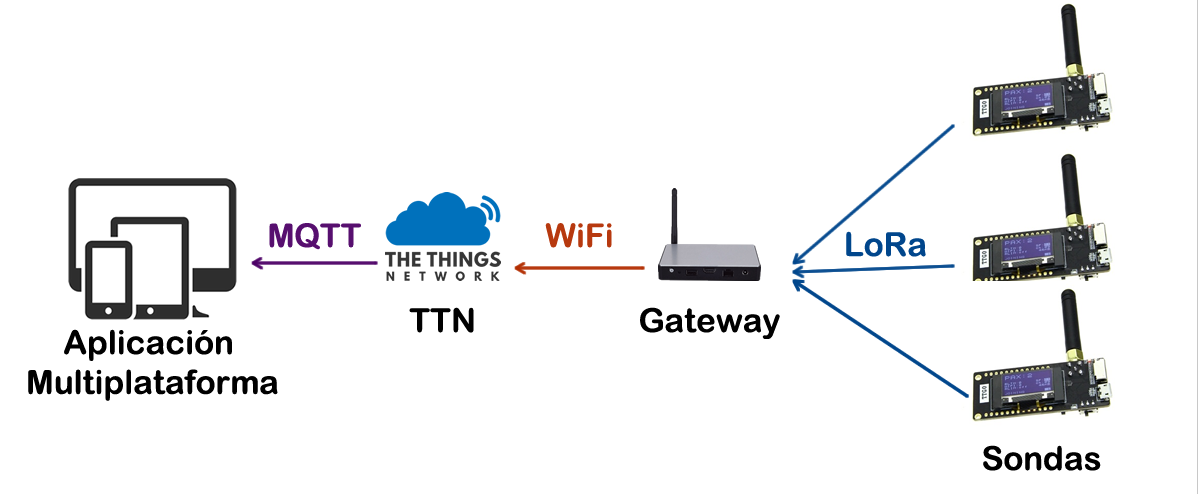
## 8.1 Análisis funcional

Todo: análisis funcional de los dispositivos y de la aplicación

## 8.2 Prototipo y mockups

Todo: prototipo de dispositivos y mockups de la aplicación

## 8.3 Arquitectura técnica



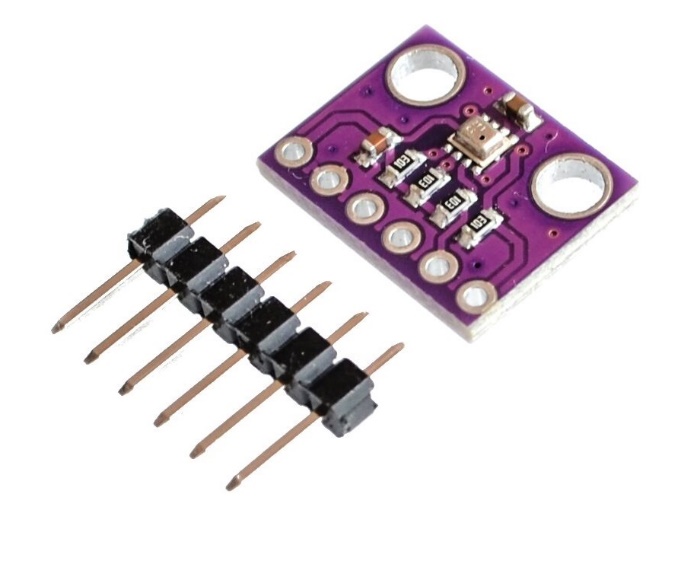
### 8.3.1 Sondas

Para las sondas se ha hecho uso del módulo TTGO ESP32, como el que se muestra en la imagen. Actualmente se trata de uno de los módulos más populares para realizar desarrollos con LoRa ya que viene integrado todo lo necesario para trabajar con este protocolo.



Es importante recalcar que los módulos comprados utilizan la frecuencia 868Mhz, que en Europa es la banda por la que se puede transmitir este tipo de comunicaciones sin tener que pedir permiso o disponer de una banda reservada para hacerlo.

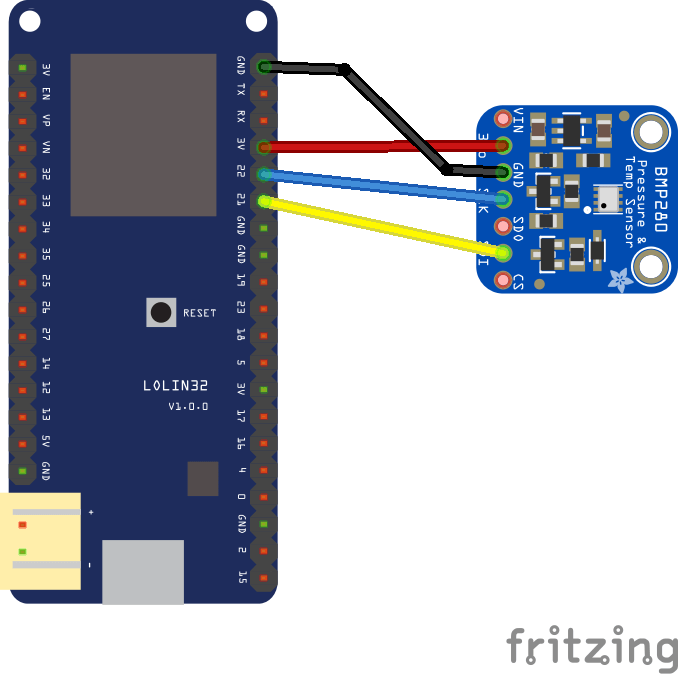
Para los sondas que obtienen los datos meteorológicos, se ha optado por el sensor BPM 280, que es capaz de obtener datos de temperatura, humedad, presión atmosférica y altura.



|  |  |  |  |
| --- | --- | --- | --- |
| Producto | Cantidad | Precio Unidad | Precio Total |
| TTGO LORA32 868Mhz OLED | 2 | 12,14€ | 24,28€\* |
| BME280 Sensor | 2 | 3,78€ | 7,56€\* |

\*Ambos productos se vendían en packs de 2, por lo que su coste es más barato que adquiriéndolos de forma individual.

Los módulos LoRa y los sensores se han conectado de la siguiente forma, lo que permite recoger datos de los sensores y enviarlos.



### 8.3.2 Gateway

### 8.3.3 TTN

### 8.3.4 Aplicación multiplataforma

## 8.4 Librerías utilizadas

Todo: nombrar las librerías utilizadas en los dispositivos Arduino y haciendo el cliente

## 8.5 Implementación

### 8.4.1 Configuración Gateway

El Gateway es el dispositivo más importante del proyecto ya que nos sirve de pasarela entre nuestras sondas y la red TTN. A partir de este elemento, se puede conectar las sondas a la red y empezar a recibir lecturas.

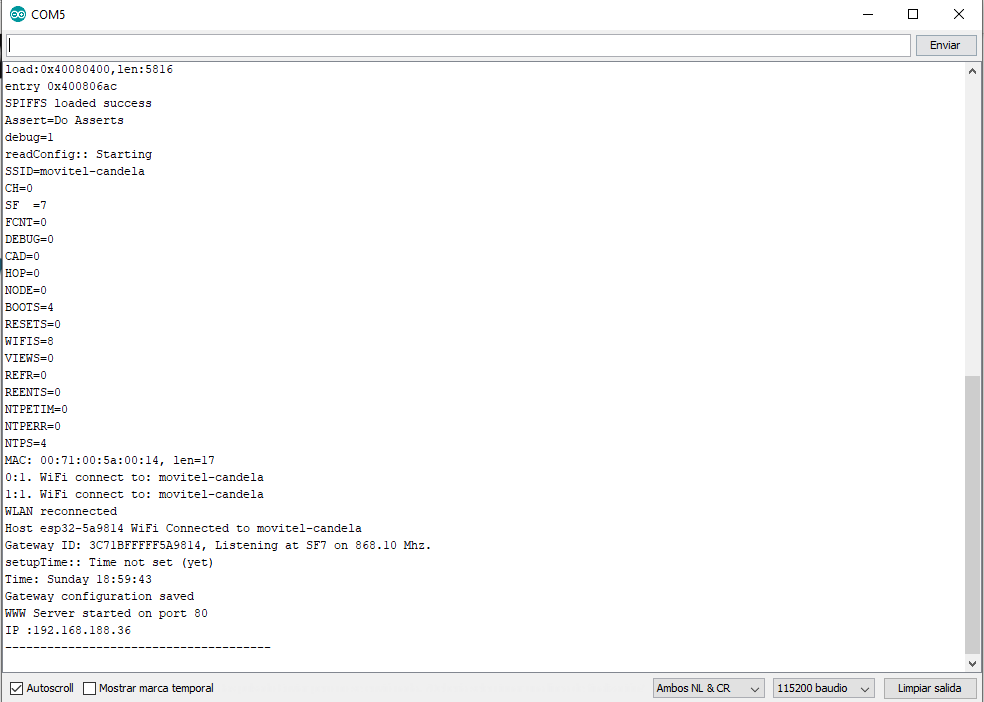
Una vez configurado el código para el Gateway, se carga en la placa, mediante el cable USB conectado al ordenador. Una vez cargado el programa, el dispositivo se reiniciará y aparece el mensaje de **STARTING**(empezando).



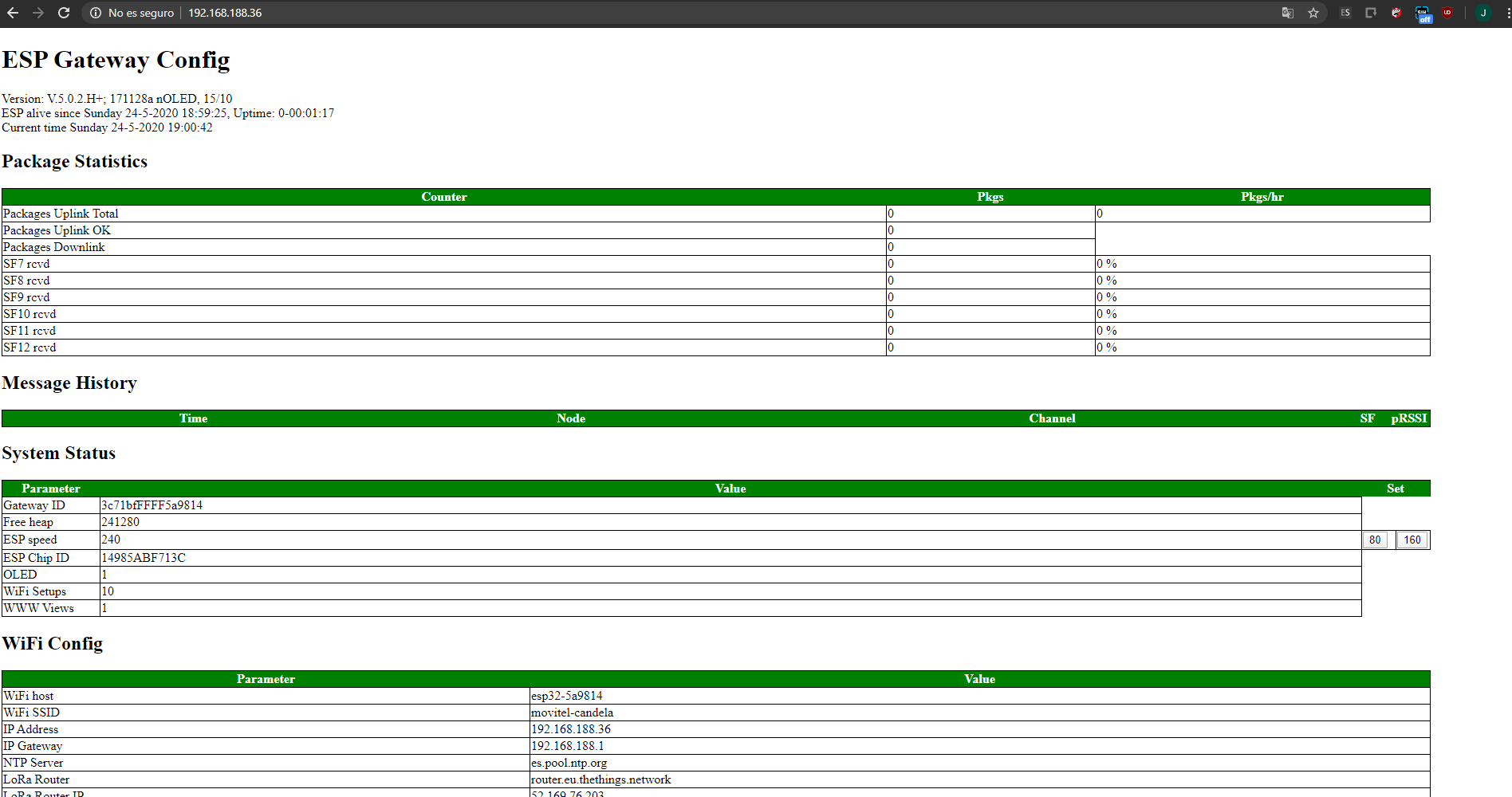
En el código cargado es importante especificar las credenciales del router Wi-Fi al que el dispositivo se tiene que conectar. Una vez conectado, cuando esté activo, aparecerá **READY**(listo) en la pantalla.

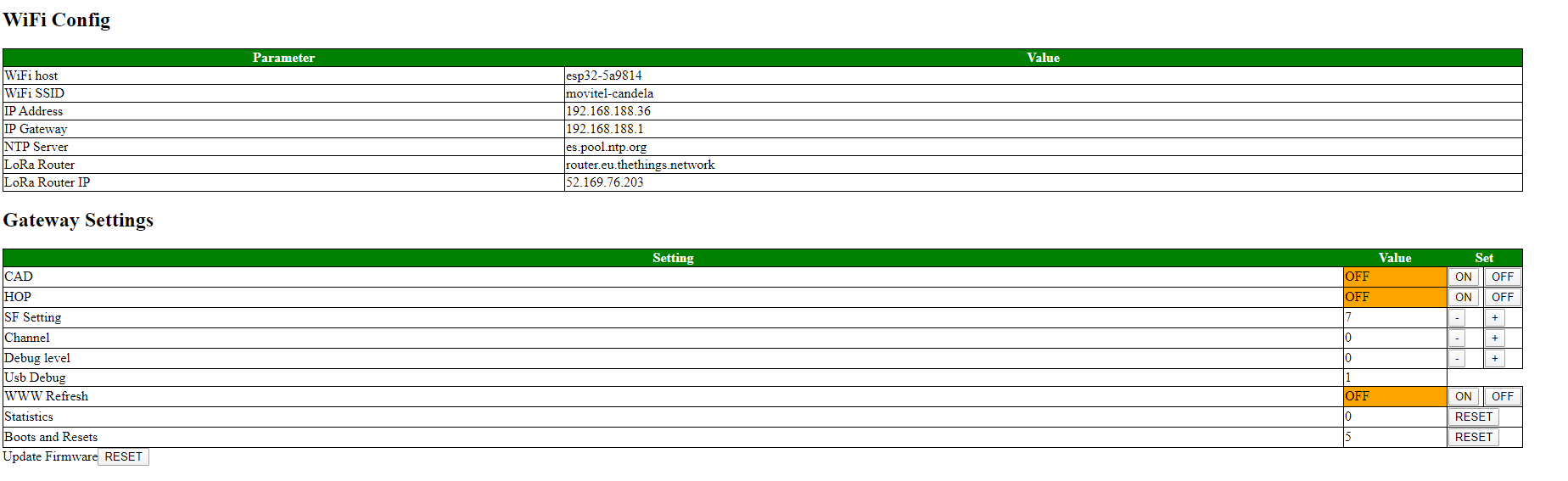


A través del monitor serial del IDE de Arduino, podemos observar información que nos muestra la placa al PC. Es útil para mostrar el estado actual, o para identificar fácilmente errores. Como podemos ver, está conectado al router WiFi de mi casa, con **SSID=movitel-candela**



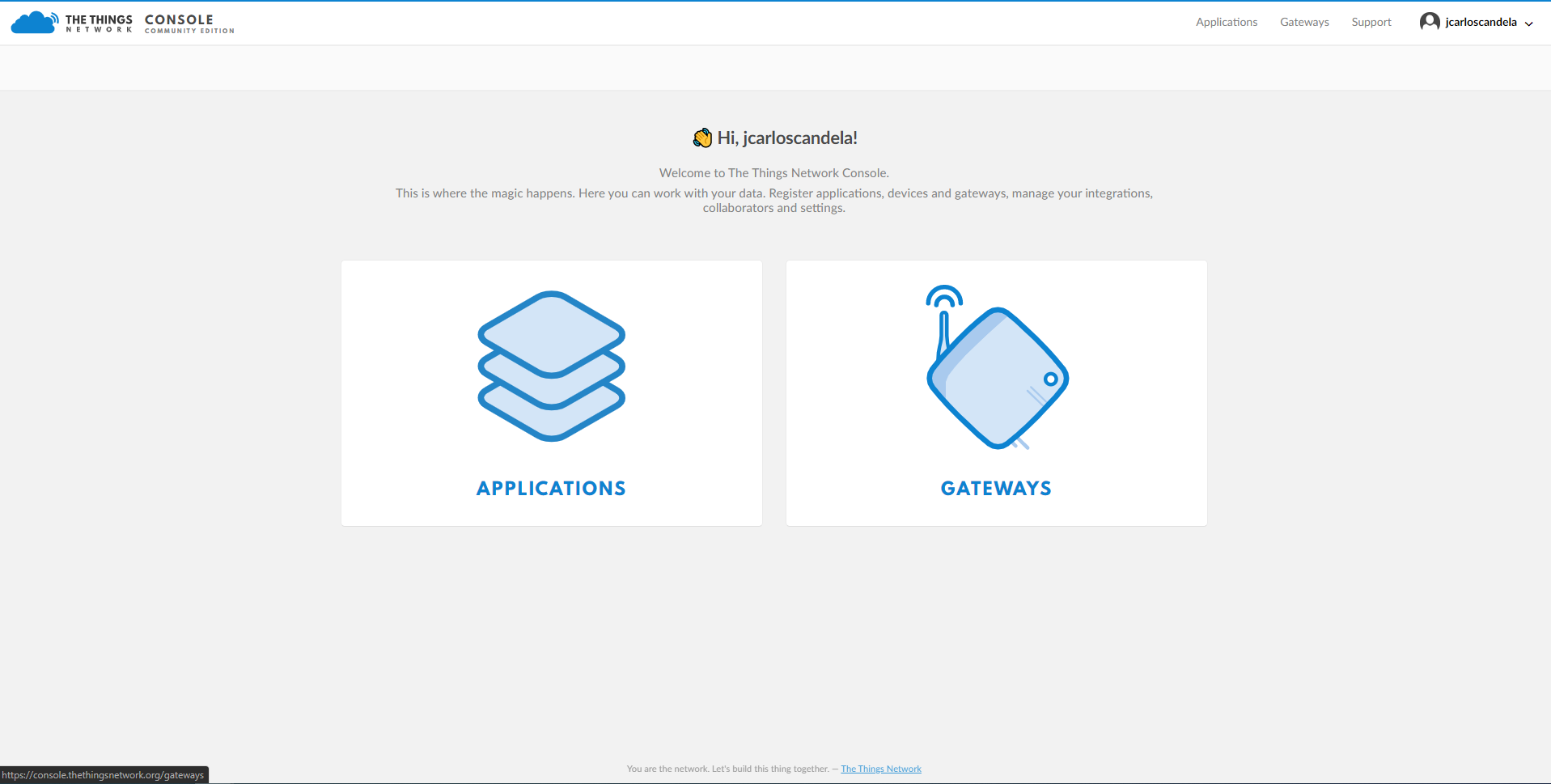
En la información que nos aparece en el monitor serial, también nos aparece un servidor web, para poder controlar el Gateway, que está situado en la siguiente **IP=192.168.188.36** y en el **Puerto=80.** Si accedemos, nos encontramos con la siguiente página de configuración, en la que podemos modificar los parámetros de forma más intuitiva.

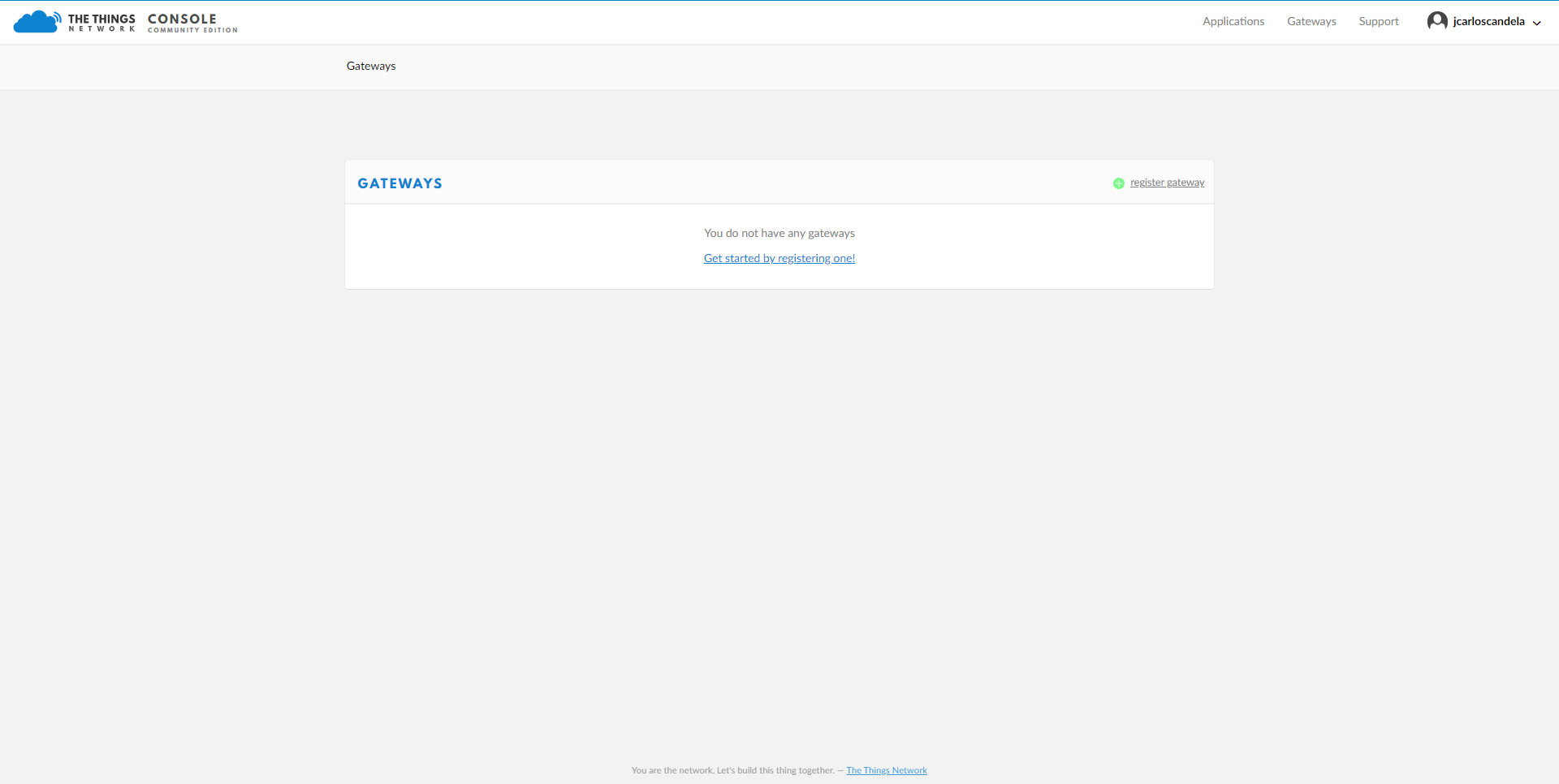


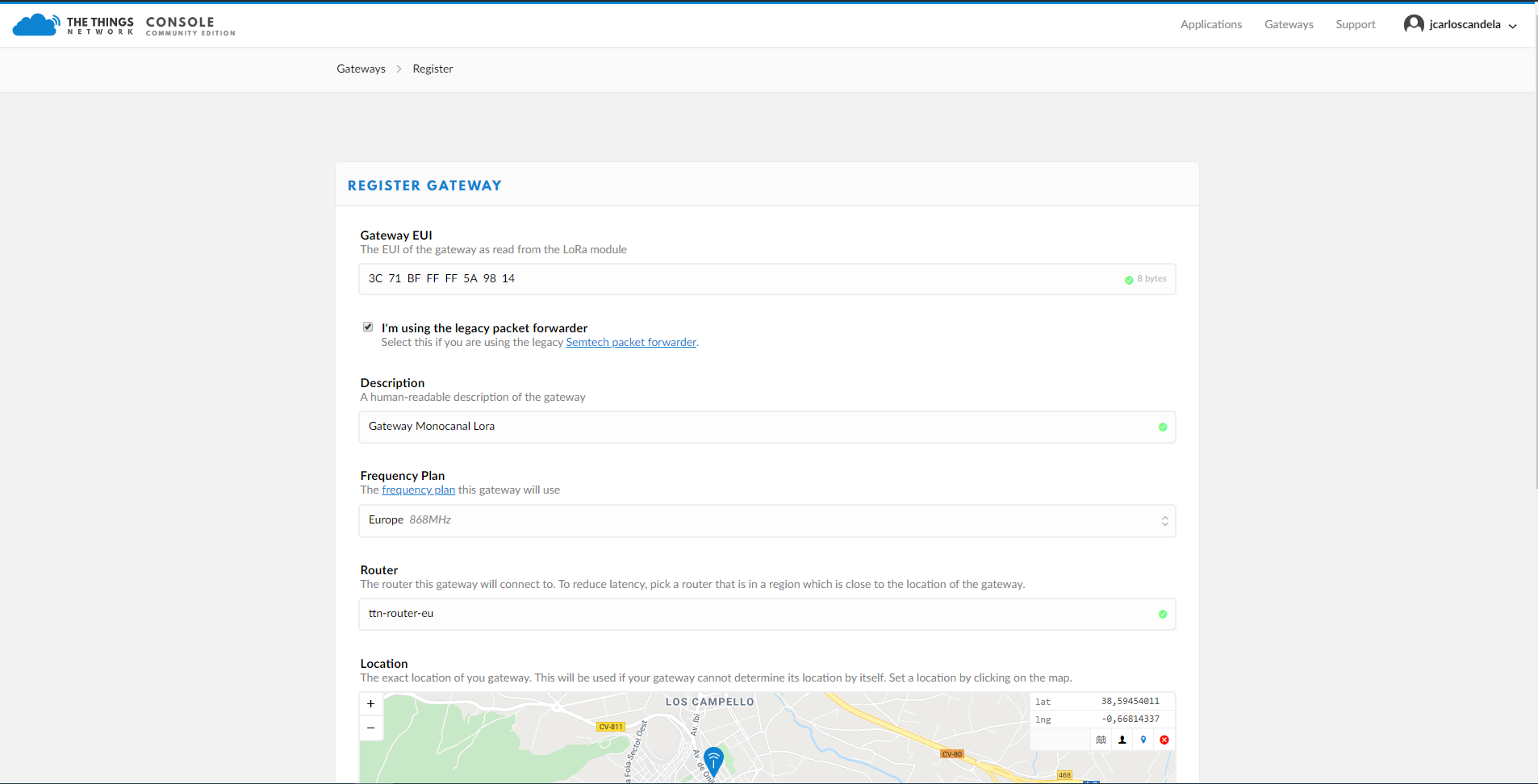


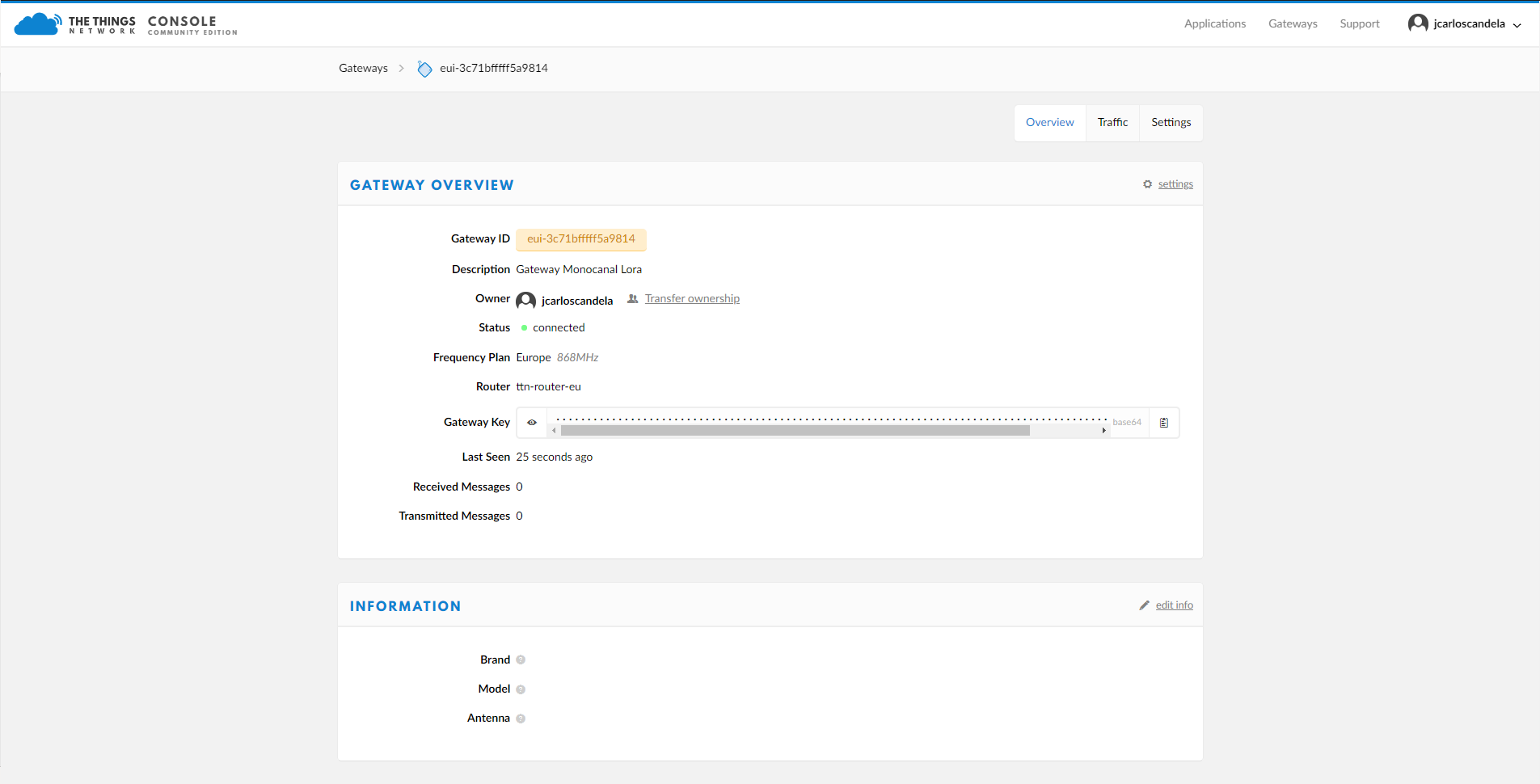
### 8.4.2 Registro Gateway en TTN

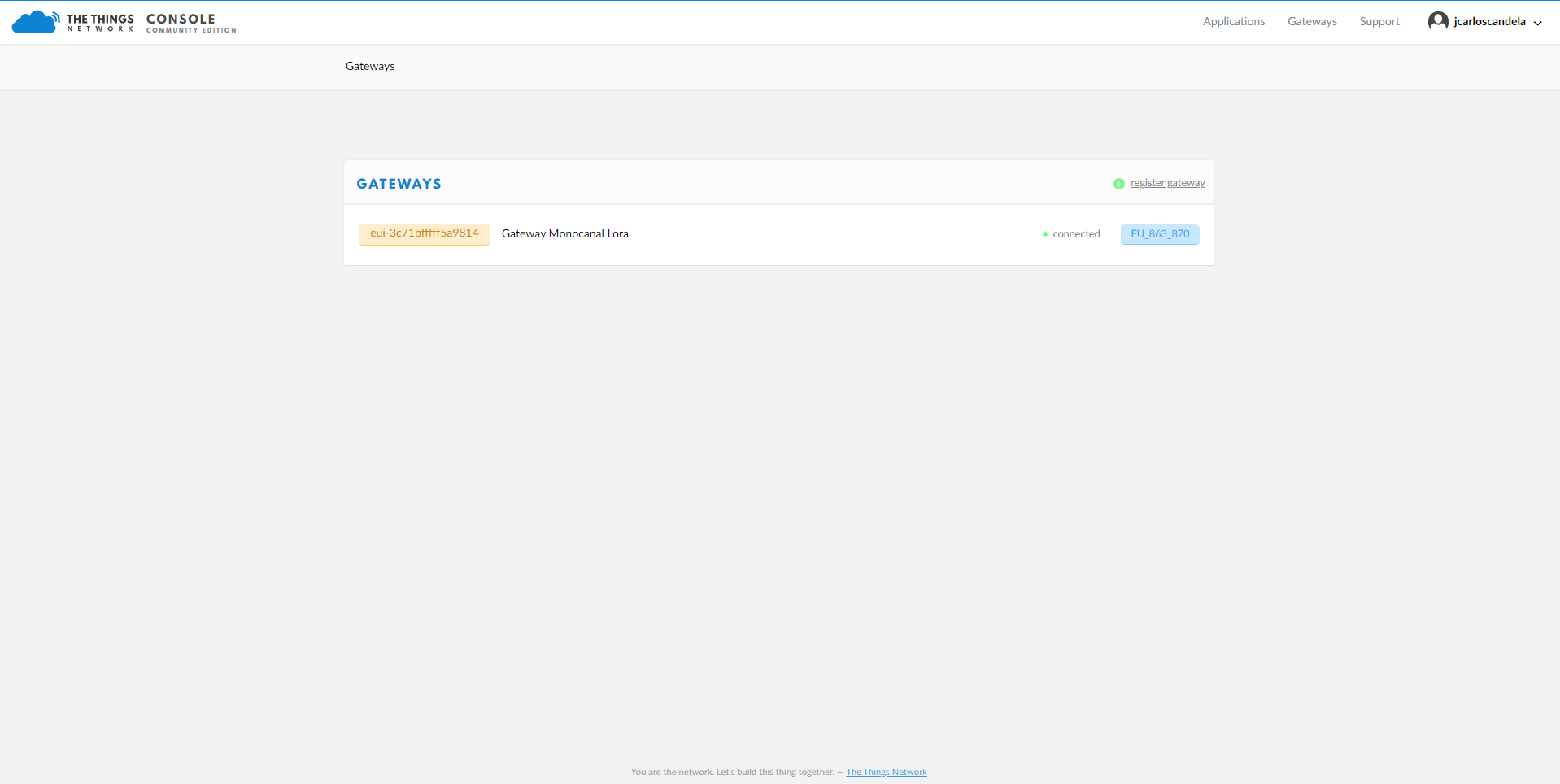




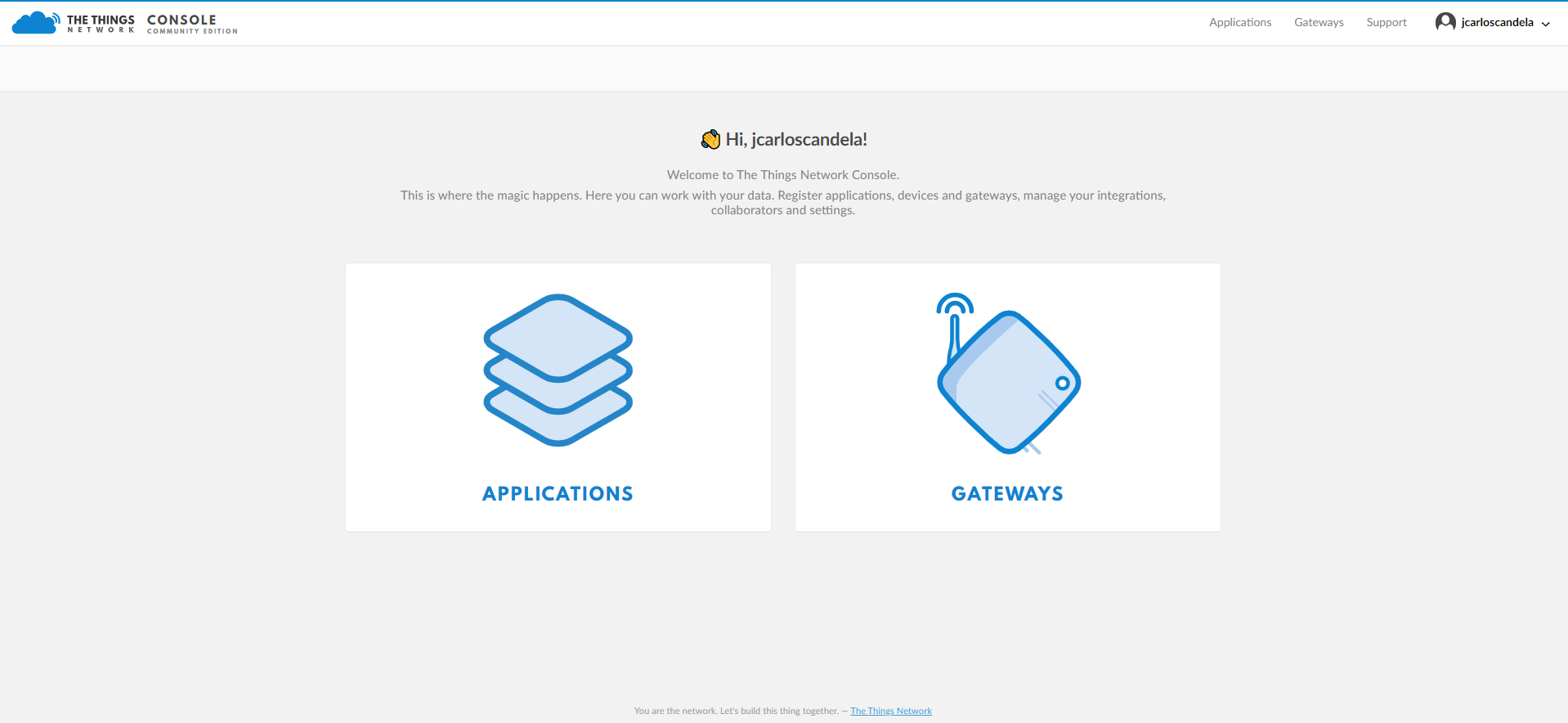


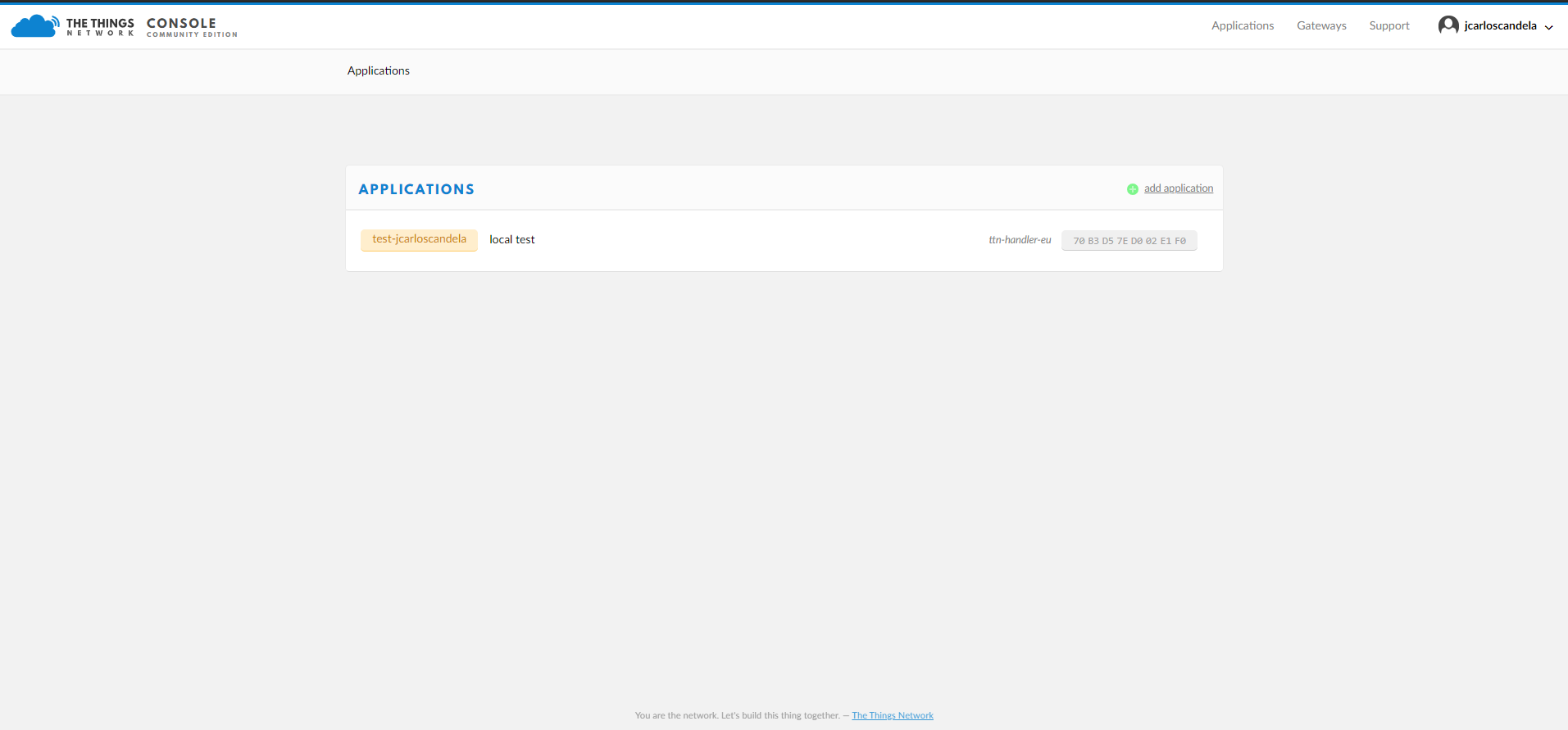


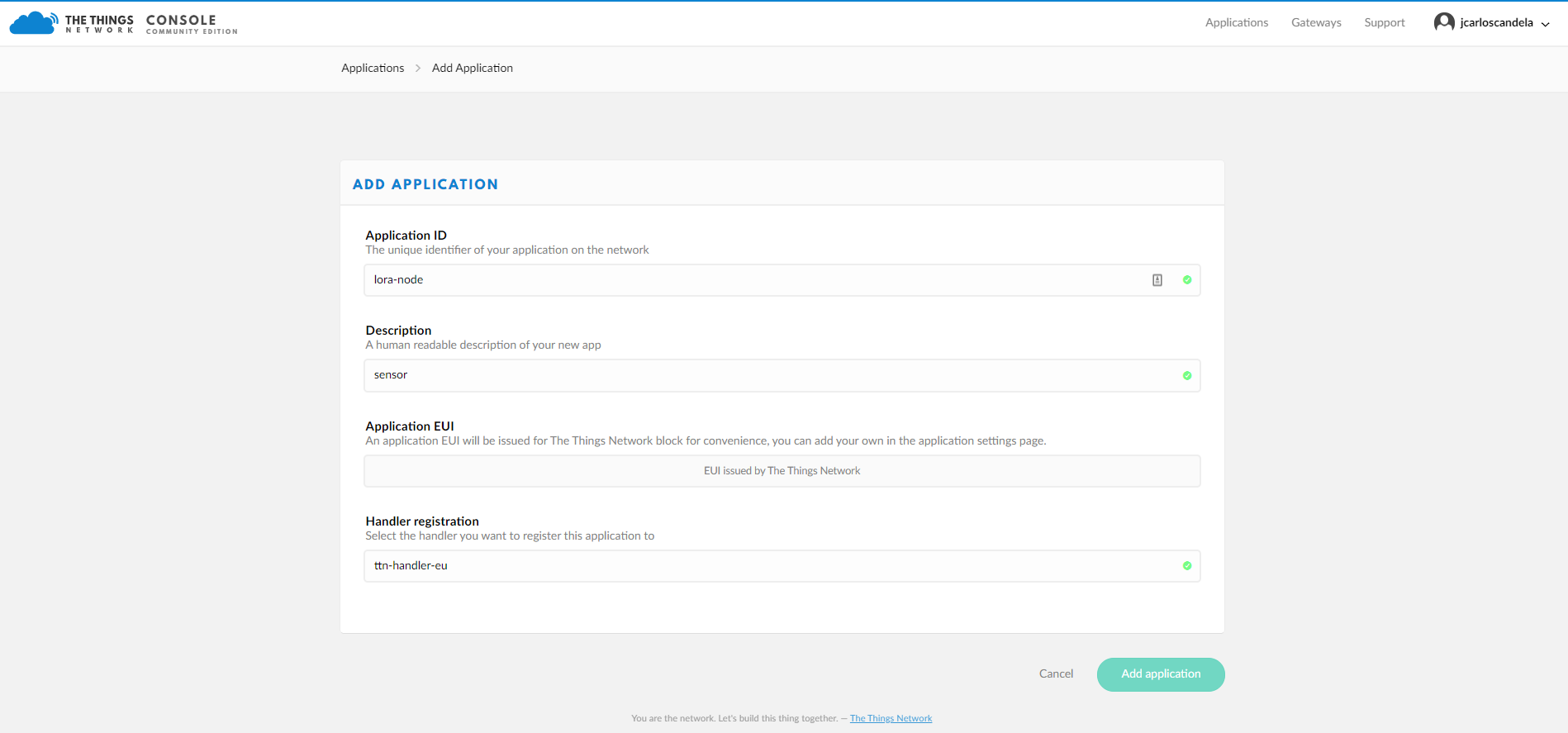


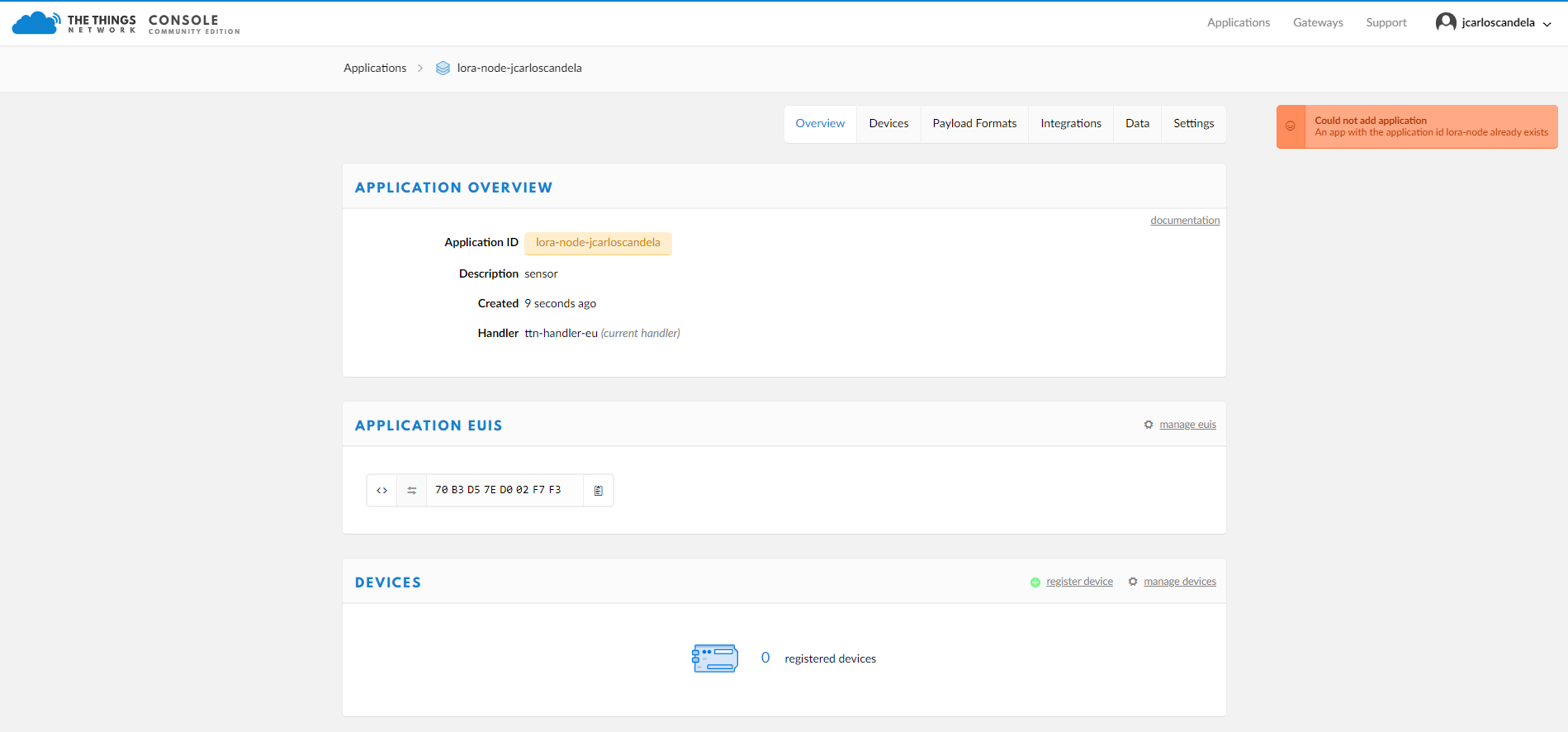


### 8.4.3 Creación Nodo Emisor

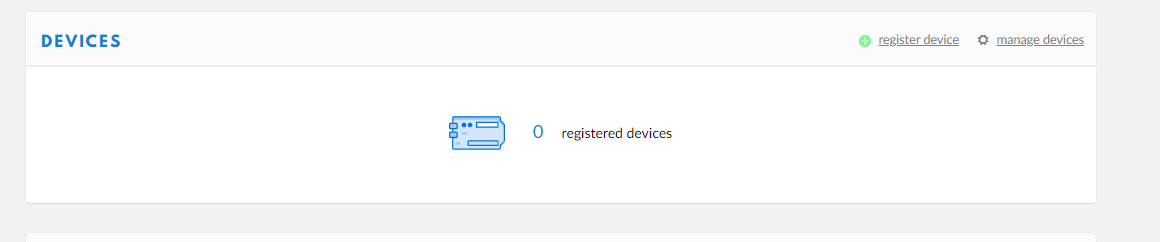


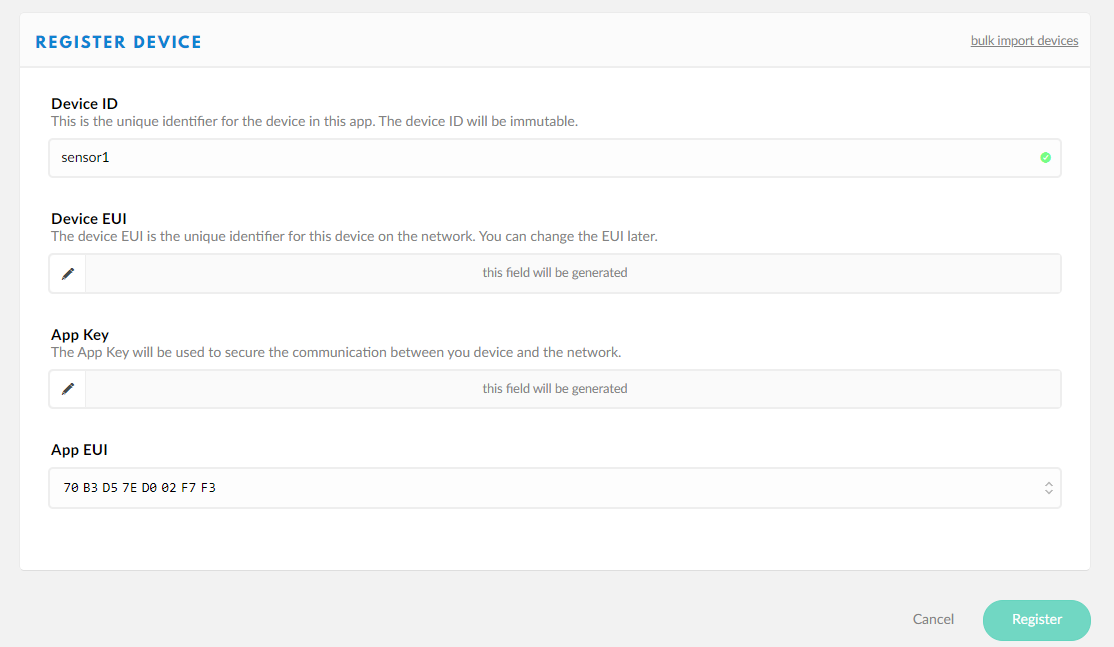


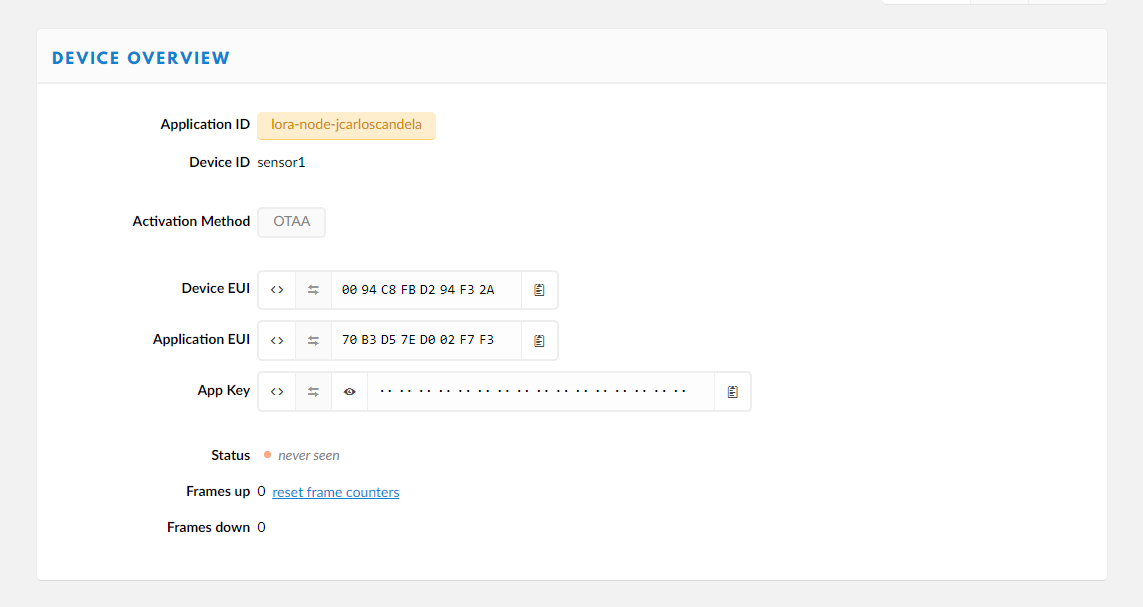




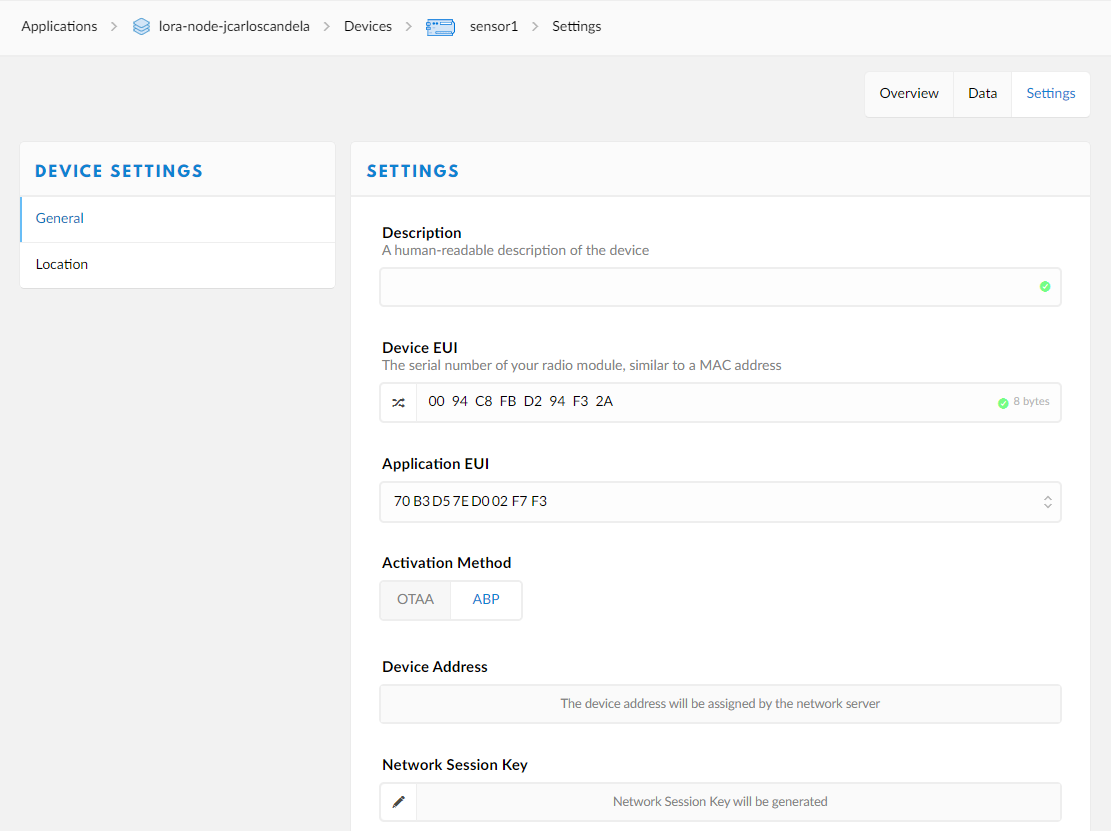
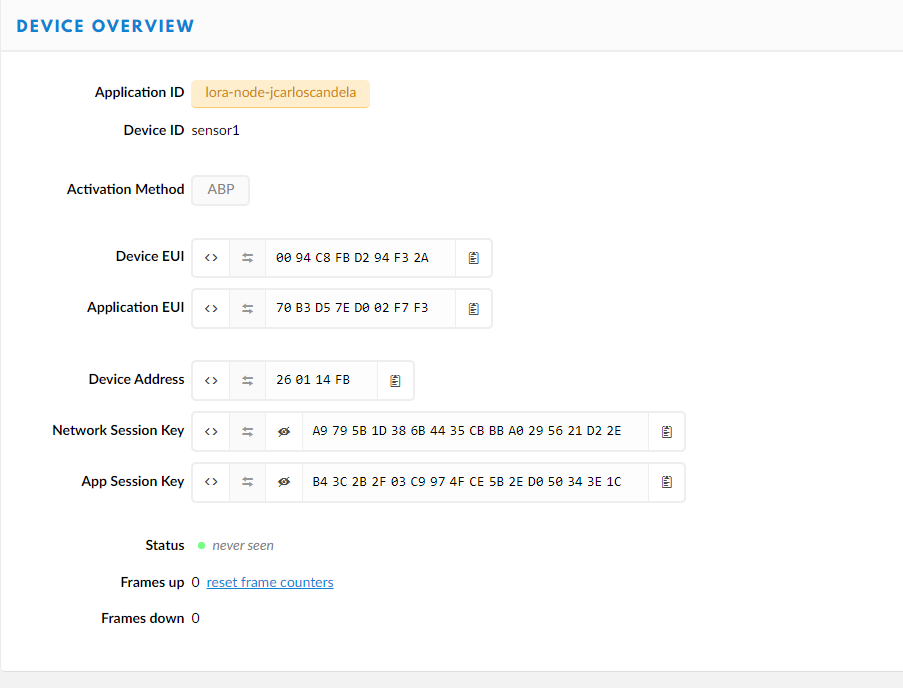
El siguiente paso es registrar un dispositivo







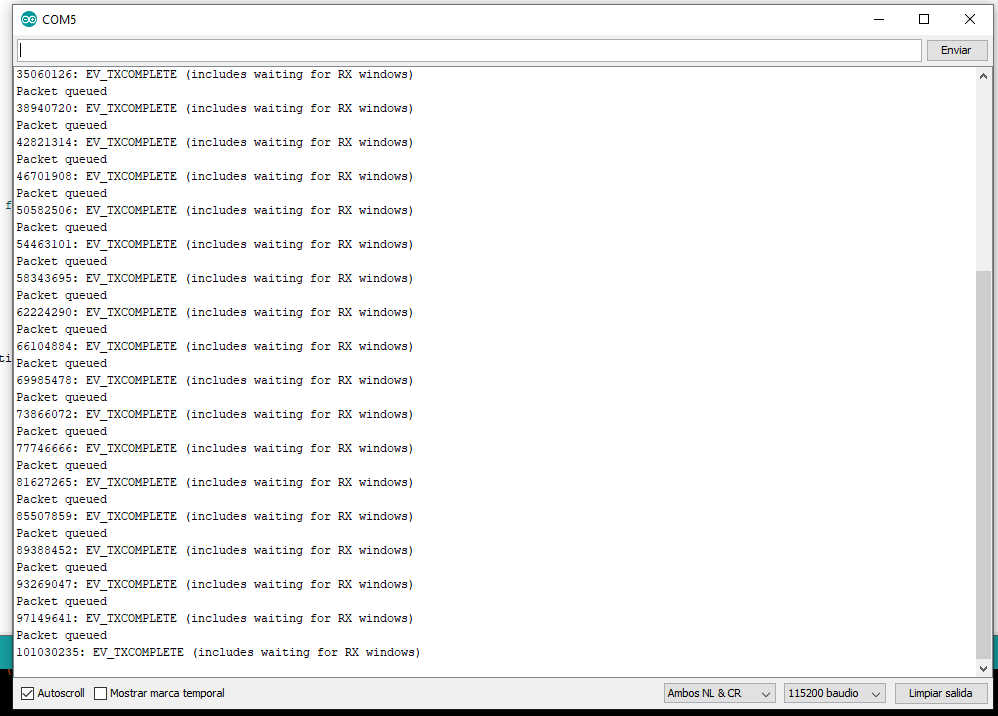
En el panel de opciones tenemos que elegir el método de activación ABP

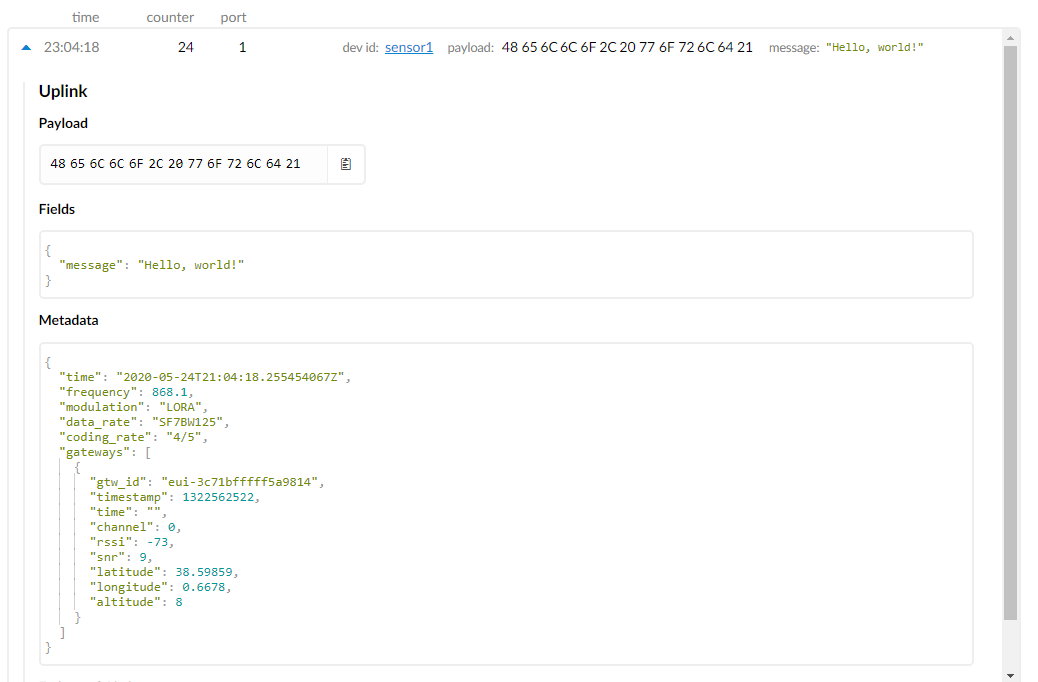
A partir de los datos generados por TTN, rellenamos los valores en las variables del script, que más tarde se cargará en el sensor:



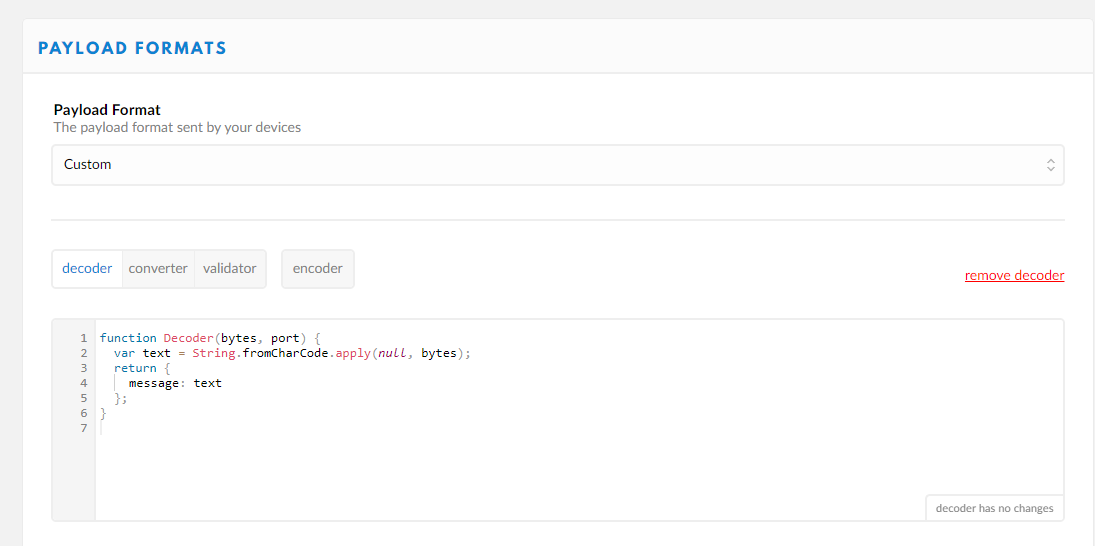
Al cargar el programa en la sonda, esta empieza a enviar datos a nuestra aplicación en TTN.



Esto es lo que recibe la aplicación



Hay que tener en cuenta que el mensaje no llega en claro, es un hexadecimal que codifica el mensaje ASCII. Para hacer la decodificación, he utilizado la siguiente función



Una vez implementada, los siguiente Payloads que reciba la aplicación, serán decodificados.

# 9. Validación

Todo: explicar lo que se ha conseguido de todo lo propuesto

# 10. Conclusiones

## 10.1 Mejoras para implementar en un futuro

## 11. Listado de acrónimos

* IoT: Internet of Things
* LoRa: Long Rage
* JSON: Javascript Object Notation
* HTTP: Hypertext Transfer Protocol
* API: Application Programming Interface
* TTN: The Things Network

# 12. Biografía y referencias

[1] <https://blog.330ohms.com/2017/05/11/que-es-sigfox-y-como-funciona/>

[2] <https://www.gradiant.org/blog/nb-iot-revolucion-mundo-conectado/>

[3]<https://es.wikipedia.org/wiki/Sistema_global_para_las_comunicaciones_m%C3%B3viles>

[4]<https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>