



**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE CUENCA  
CARRERA DE INGENIERÍA DE SISTEMAS**

**DESPLIEGUE Y CONFIGURACIÓN DE UNA RED LoRaWAN USANDO UNA  
PLATAFORMA THE THINGS NETWORK (TTN) PARA DISPOSITIVOS DE  
INTERNET DE LAS COSAS (IoT)**

Trabajo de titulación previo a la obtención del  
Título de Ingeniero de Sistemas

AUTORES:

**RICARDO MARCELO CHUQUI ORTEGA  
DIEGO MAURICIO CASTRO CASTRO**

TUTOR:

**ERWIN JAIRO SACOTO CABRERA, PhD**

Cuenca, Ecuador  
2022

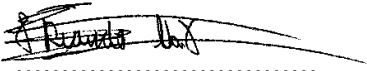
## CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Ricardo Marcelo Chuqui Ortega con documento de identificación N° 0106754823 y Diego Mauricio Castro Castro con documento de identificación N° 0106078876; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 31 de agosto de 2022

Atentamente,



.....  
Ricardo Marcelo Chuqui Ortega  
0106754823



.....  
Diego Mauricio Castro Castro  
0106078876

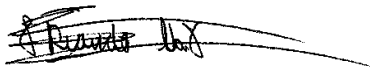
**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN  
A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Ricardo Marcelo Chuqui Ortega con documento de identificación N° 0106754823 y Diego Mauricio Castro Castro con documento de identificación N° 0106078876, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: **“Despliegue y configuración de una red LoRaWAN usando una plataforma The Things Network (TTN) para dispositivos de Internet de las Cosas (IoT)”**, el cual ha sido desarrollado para optar por el título de: Ingeniero de Sistemas, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 31 de agosto de 2022

Atentamente,



.....  
Ricardo Marcelo Chuqui Ortega  
0106754823



.....  
Diego Mauricio Castro Castro  
0106078876

## CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Erwin Jairo Sacoto Cabrera con documento de identificación N° 0301185229, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **DESPLIEGUE Y CONFIGURACIÓN DE UNA RED LORAWAN USANDO UNA PLATAFORMA THE THINGS NETWORK (TTN) PARA DISPOSITIVOS DE INTERNET DE LAS COSAS (IoT)**, realizado por Ricardo Marcelo Chuqui Ortega con documento de identificación N° 0106754823 y por Diego Mauricio Castro Castro con documento de identificación N° 0106078876, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca 31 de agosto de 2022

Atentamente,

ERWIN JAIRO SACOTO  
SACOTO  
CABRERA

Firmado digitalmente  
por ERWIN JAIRO  
SACOTO CABRERA  
Fecha: 2022.12.12  
16:20:56 -05'00'

.....

Ing. Erwin Jairo Sacoto Cabrera, PhD  
0301185229



## **Dedicatoria**

### **Ricardo Chuqui**

Primeramente, quiero dar gracias a Dios por haberme permitido alcanzar uno de mis sueños, a mi padre Conelio Chuqui que desde el cielo fue mi fortaleza, a mi querida mamá Rosa Ortega por su apoyo incondicional, a mis hermanos Juan y Alonso por estar siempre presentes y por todo el apoyo que me brindaron para alcanzar mis metas.

A las personas que me brindaron acogida en mis años de estudios, la familia Condo Sinchi quienes fueron un gran apoyo durante toda mi trayectoria universitaria.

A mi mismo por haber tenido dedicación, constancia y perseverancia.

Dedico esta tesis a todos ellos, muchas gracias por estar siempre presentes y haberme ayudado a alcanzar mis metas.

### **Diego Castro**

Este trabajo va dedicado a Dios por creer en mí y permitirme culminar este proceso educativo, a mi maravillosa familia Daniela y Aitana quienes serán un pilar fundamental de aliento en los buenos y malos momentos.

A todos lo que han contribuido en el alcance de este logro; a mis padres por su confianza, a mis profesores por su valioso tiempo de enseñanzas en las aulas.

Estoy seguro de que siempre contaré con su valioso e incondicional apoyo y sepan que todo es posible con esfuerzo y dedicación.

# Resumen

Las tecnologías de Red de Área Amplia de Baja Potencia (LPWAN) desempeñan un papel fundamental en las aplicaciones de Internet de las Cosas (Internet of Things - IoT), debido a su capacidad para cumplir con los requisitos clave de IoT (por ejemplo, largo alcance, bajo costo, datos pequeños). La principal problemática se presenta con el poco conocimiento de la población y la necesidad de conocer los elementos de seguridad necesarios para la implementación de estas tecnologías buscando siempre beneficiar a la población con un gran alcance de cobertura y con un bajo consumo de energía ideal para IoT. Para el cumplimiento de las metas, presentamos una descripción técnica de la tecnología LoRaWAN para considerar sus principales características, oportunidades y problemas abiertos. También comparamos las herramientas de simulación más importantes para investigar y analizar el rendimiento de la red que se han desarrollado recientemente. Luego, presentamos una evaluación comparativa de los simuladores de LoRa para resaltar sus características. Además, clasificamos los esfuerzos recientes para mejorar el rendimiento en términos de consumo de energía, tasa de extracción de datos puros, escalabilidad de red, cobertura de red, Calidad de Servicio (Quality of Service - QoS) y seguridad. Finalmente, aunque nos enfocamos más en los problemas y soluciones de LoRa/LoRaWAN, presentamos orientación e instrucciones para futuras investigaciones sobre tecnologías usando como servidor de red The Things Network (TTN) y como servidor de aplicaciones Azure IoT Hub. Gracias a esta investigación se pudo obtener los resultados del comportamiento de la información generada por los nodos finales, equipos de tecnología enviando resultados hacia las bases de datos configuradas, pudiendo obtener reportes, los mismos que nos ayudaran para la toma de decisiones y centralizando la información necesaria teniendo como resultado una herramienta útil que puede ser usada en proyectos como estaciones meteorológicas, drones, aplicaciones móviles o GPS.

**Palabras claves:** (LoRaWAN, IoT, Servidor, Nodos, Tecnología, Red, Wifi, Gateway, LoRa, TTN, servicio, Hub, sensor, QoS).

# Abstract

Low Power Wide Area Network (LPWAN) technologies play a critical role in Internet of Things (IoT) applications, due to their ability to meet key IoT requirements (for example, long scope, low cost, small data). The main problem arises with the little knowledge of the population and the need to know the security elements necessary for the implementation of these technologies, always seeking to benefit the population with a wide range of coverage and with low energy consumption, ideal for IoT. . For the fulfillment of the goals, we present a technical description of the LoRaWAN technology to consider its main characteristics, opportunities and open problems. We also compare the most important simulation tools for researching and analyzing network performance that have been recently developed. Then, we present a benchmarking of LoRa simulators to highlight their features. In addition, we rank recent efforts to improve performance in terms of power consumption, raw data extraction rate, network scalability, network coverage, Quality of Service (QoS), and security. Finally, although we focus more on LoRa/LoRaWAN problems and solutions, we present guidance and instructions for future research on technologies using The Things Network (TTN) as network server and Azure IoT Hub as application server. Thanks to this investigation, it was possible to obtain the results of the behavior of the information generated by the final nodes, technology teams sending results to the configured databases, being able to obtain reports, which will help us to make decisions and centralize the information. necessary information resulting in a useful tool that can be used in projects such as weather stations, drones, mobile applications or GPS.

**Keys words:** (LoRaWAN, IoT, Server, Nodes, Technology, Network, WiFi, Gateway, LoRa, TTN, service, Hub, sensor, QoS).

# Contenido

<b>Resumen</b>	<b>vi</b>
<b>Abstract</b>	<b>vii</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Problema</b>	<b>4</b>
2.1. Antecedentes . . . . .	5
2.2. Importancia y Alcances . . . . .	6
2.3. Delimitación . . . . .	7
2.3.1. Espacial . . . . .	7
2.3.2. Temporal . . . . .	7
2.3.3. Académica . . . . .	8
<b>3. OBJETIVOS GENERALES Y ESPECÍFICOS</b>	<b>9</b>
3.1. Objetivo General . . . . .	9
3.2. Objetivos Específicos . . . . .	9
<b>4. REVISIÓN DE LA LITERATURA O FUNDAMENTOS TEÓRICOS</b>	<b>10</b>
4.1. Internet de las Cosas (Internet of Things - IoT) . . . . .	10
4.1.1. Elementos de IoT . . . . .	10
4.1.2. Características de IoT . . . . .	11
4.2. Red de Área Extendida de Baja Potencia (Low Power Wide Area Network - LPWAN) . . . . .	11
4.3. Red de Largo Alcance (Long Range - LoRa) . . . . .	12
4.3.1. Red de Área Extensa de Largo Alcance (Long Range Wide Area Network - LoRaWAN) . . . . .	13
4.3.2. Sigfox . . . . .	18
4.3.3. IoT de Banda Estrecha (Narrow Band IoT - NB IoT) . . . . .	19
4.3.4. Comparación entre LoRaWAN, Sigfox y NB-IoT . . . . .	19
4.4. The Things Network (TTN) . . . . .	20
4.4.1. Cayenne LPP (Low Power Payload) . . . . .	21
4.5. Microsoft Azure IoT Hub. . . . .	21
4.5.1. Azure Stream Analytics . . . . .	22

4.5.2.	Cosmos DB . . . . .	23
4.5.3.	Blob Storage . . . . .	24
4.5.4.	Power BI . . . . .	25
4.6.	Gateway LoRaWAN . . . . .	26
4.6.1.	Características de MikroTik RouterOS . . . . .	27
4.6.2.	Kit wAP LoRa8 . . . . .	29
4.7.	Hardware Heltec Wi Fi Lora ESP32 . . . . .	30
4.7.1.	Bus de datos I2C . . . . .	31
4.7.2.	IDE Arduino para la programación HelTec ESP32 . . . . .	32
4.8.	Sensor de Presión Digital BMP280 . . . . .	32
<b>5.</b>	<b>MARCO METODOLÓGICO</b>	<b>34</b>
5.1.	Metodología de diseño de una Red LoRaWAN . . . . .	34
5.2.	Diseño y Arquitectura de la red LoRaWAN con TTN y Azure IoT Hub. . . . .	35
5.3.	Creación y configuración del GW, TTN, HelTec ESP32-Sensor BMP280, Azure IoT Hub. . . . .	37
5.3.1.	Configuración del equipo wAP LR8 kit. . . . .	37
5.3.2.	Configuración del GW en TTN. . . . .	41
5.4.	Creación y configuración de la aplicación y el dispositivo final en TTN . . . . .	52
5.4.1.	Configuración de la aplicación en TTN. . . . .	52
5.4.2.	Creación y Configuración del dispositivo Final HelTec ESP32 . . . . .	55
5.4.3.	Programación del dispositivo HelTec ESP32 Lora 32 V2 en el IDE Arduino . . . . .	59
5.4.4.	Desarrollo del entorno con Arduino en la placa HelTec ESP32 . . . . .	62
5.5.	Integración de Azure IoT Hub y TTN. . . . .	76
5.5.1.	Crear cuenta en Azure IoT Hub . . . . .	76
5.5.2.	Integración de Azure IoT Hub con la plataforma de TTN . . . . .	79
5.5.3.	Implementación/Instalación de Servicios para Recolectar y Interpretar los datos del dispositivo HelTec ESP32-Sensor de Presión BMP280. . . . .	85
5.5.4.	Azure Stream Analytics . . . . .	85
5.5.5.	Azure Cosmos DB . . . . .	86
5.5.6.	Azure Storage Account . . . . .	90
5.5.7.	Azure Power BI . . . . .	95
5.5.8.	Integración de Cosmos DB, Storage Account y Power BI en Azure Stream Analytics. . . . .	98
<b>6.</b>	<b>RESULTADOS</b>	<b>107</b>
6.1.	Pruebas y resultados con Azure IoT Hub . . . . .	107
6.2.	Cayenne LPP . . . . .	118
<b>7.</b>	<b>CRONOGRAMA</b>	<b>123</b>

---

<b>8. PRESUPUESTO</b>	<b>124</b>
<b>9. CONCLUSIONES</b>	<b>125</b>
<b>10. RECOMENDACIONES</b>	<b>127</b>
<b>11. TRABAJO FUTURO</b>	<b>128</b>
<b>Bibliografía</b>	<b>129</b>
<b>A. Anexo: Código Arduino</b>	<b>134</b>
A.1. Código implementado para la conexión del dispositivo HelTec ESP32 V2 + Sensor BMP280 en TTN . . . . .	134
A.1.1. Código para ver el ID del dispositivo HelTec ESP32 . . . . .	136
A.1.2. Código de la biblioteca de Cayenne para TTN . . . . .	136
A.2. Código para ver la dirección de bus del dispositivo . . . . .	137
<b>B. Anexo: Generar Reportes</b>	<b>139</b>
B.1. Pasos para usar PowerBI Robots . . . . .	139

# Lista de Figuras

4-1. Protocolo de Comunicación LoRaWAN [1] . . . . .	13
4-2. Arquitectura de la LoRaWAN [2] . . . . .	14
4-3. Clases de dispositivos LoRaWAN [3] . . . . .	15
4-4. Seguridad LoRaWAN [2] . . . . .	18
4-5. Arquitectura de Sigfox [4] . . . . .	19
4-6. Cobertura de GW a nivel mundial [5] . . . . .	20
4-7. Página web de Cayenne [6] . . . . .	21
4-8. Arquitectura de Stream Analytics [7]. . . . .	23
4-9. Arquitectura de Cosmos DB [8]. . . . .	24
4-10. Proceso de detención de mensajes Blob Storage [9]. . . . .	25
4-11. Visualización de Power BI [10]. . . . .	26
4-12. Red GW LoRaWAN [11] . . . . .	27
4-13. Consumidores internacionales de MikroTik [12] . . . . .	28
4-14. Circuitos integrados de hardware [12] . . . . .	29
4-15. Diagrama de Pines Heltec ESP32 LoRa 32(v2) [13]. . . . .	31
4-16. Sensor de presión BMP280 [14] . . . . .	33
5-1. Arquitectura de la red LoRaWAN. . . . .	35
5-2. Login del GW MikroTik wAP LR8 kit. . . . .	37
5-3. Crear Servidor LoRa. . . . .	38
5-4. Agregar nuevo servidor TTN. . . . .	38
5-5. Configurar Servidor TTN V3. . . . .	39
5-6. Levantar Servidor TTN. . . . .	40
5-7. Estado del equipo LoRa GW. . . . .	41
5-8. Página de inicio de TTN . . . . .	42
5-9. Tipo de suscripción para TTN . . . . .	43
5-10. Cluster y región de TTN. . . . .	44
5-11. Registrar una cuenta nueva. . . . .	45
5-12. Formulario para la crear una cuenta de TTN. . . . .	46
5-13. Login de ingreso a TTN. . . . .	47
5-14. Ingresar para crear GW. . . . .	48
5-15. Creación de un nuevo GW. . . . .	48
5-16. Formulario para el registro del GW. . . . .	49

5-17.	Formulario para la registro del GW. . . . .	50
5-18.	Verificación del estado del GW. . . . .	50
5-19.	Datos de conexión entre el GW y TTN. . . . .	51
5-20.	Página de inicio para crear una aplicación. . . . .	52
5-21.	Agregar una nueva aplicación. . . . .	53
5-22.	Formulario para la creación de la aplicación. . . . .	54
5-23.	Aplicación creada correctamente. . . . .	55
5-24.	Registro del dispositivo usando el Repositorio de TTN. . . . .	56
5-25.	Crear el DevEUI . . . . .	57
5-26.	Configuración y creación del dispositivo. . . . .	58
5-27.	Nodo final creada correctamente. . . . .	59
5-28.	Inicio de la configuración del IDE. . . . .	60
5-29.	Configuración biblioteca de gestor de tarjeta ESP32. . . . .	60
5-30.	Gestor de Tarjetas en el IDE de Arduino. . . . .	61
5-31.	Instalación correcta de la tarjeta. . . . .	61
5-32.	Configuración correcta del IDE Arduino. . . . .	62
5-33.	Implementación de librerías. . . . .	62
5-34.	Biblioteca de librerías para el IDE de Arduino. . . . .	63
5-35.	Descarga de librería CayenneLPP. . . . .	63
5-36.	Descarga de librería Adafruit BMP280. . . . .	63
5-37.	Descarga de librería ESP32_LoRaWAN. . . . .	64
5-38.	Descarga de librería Adafruit_Sensor. . . . .	65
5-39.	Descarga de librería I2CScanner. . . . .	65
5-40.	Código para obtener el ChipID . . . . .	66
5-41.	Pagina web para ingresar el ChipID. . . . .	66
5-42.	Licencia valida para nuestro dispositivo. . . . .	66
5-43.	Licencia valida obtenida agregada al código. . . . .	67
5-44.	Parámetros reemplazados en el código. . . . .	67
5-45.	Parámetros de nuestro dispositivo creado en TTN. . . . .	67
5-46.	Código de velocidad de transmisión y dirección de conexión del Bus de datos. . . . .	67
5-47.	Código ejecutado con la librería I2C. . . . .	68
5-48.	Código para la conexión al TTN. . . . .	69
5-49.	Código para la recolección de datos con el sensor BMP 280. . . . .	69
5-50.	Revisión de parámetros antes de cargar el código . . . . .	70
5-51.	Carga y ejecución del código en HelTec ESP32. . . . .	71
5-52.	Aplicación conectada . . . . .	72
5-53.	Dispositivo HelTec ESP32 conectado . . . . .	72
5-54.	Conexión establecida correctamente con ESP32 y BMP280 con TTN. . . . .	73
5-55.	Formateo de datos Uplink . . . . .	74
5-56.	Formateo de datos Downlink . . . . .	75



5-57.Datos en tiempo real en TTN . . . . .	75
5-58.Página de inicio Azure IoT Hub . . . . .	76
5-59.Suscripción de pago Azure IoT Hub. . . . .	76
5-60.Acuerdos Suscripción Azure IoT Hub. . . . .	77
5-61.Formulario Información de Pago. . . . .	77
5-62.Tipo de Soporte Azure IoT Hub. . . . .	78
5-63.Página de inicio a los recursos de Azure IoT Hub. . . . .	79
5-64.página de inicio de Azure IoT Hub. . . . .	79
5-65.Integración de Azure IoT Hub. . . . .	80
5-66.Plantilla de Azure IoT Hub para la integración con TTN. . . . .	81
5-67.Creación del recurso Azure IoT Hub. . . . .	82
5-68.Implementación Correcta del recurso de Azure IoT Hub. . . . .	83
5-69.Configuración de Salida de Azure IoT Hub . . . . .	83
5-70.Finalización integración Azure IoT Hub y TTN. . . . .	84
5-71.Dispositivo detectado en Azure IoT Hub. . . . .	84
5-72.Instalar servicio Stream Analytics. . . . .	85
5-73.Crear el servicio. . . . .	85
5-74.Crear servicio Stream Analytics . . . . .	86
5-75.Instalar el servicio Cosmos DB. . . . .	87
5-76.Crear el servicio . . . . .	87
5-77.Seleccionar el API para la instalación . . . . .	88
5-78.Formulario para crear servicio de Cosmos DB. . . . .	89
5-79.Crear el servicio Cosmos DB . . . . .	90
5-80.Buscar y crear el servicio . . . . .	91
5-81.Formulario para crear servicio Storage Account. . . . .	92
5-82.Crear el servicio . . . . .	93
5-83.Crear un contenedor de almacenamiento. . . . .	94
5-84.Completar el formulario para el contenedor . . . . .	94
5-85.página de inicio de Power BI . . . . .	95
5-86.Tipo de suscripción . . . . .	95
5-87.Instalar el servicio Power BI. . . . .	96
5-88.Crear el servicio. . . . .	96
5-89.Completar los campos del formulario . . . . .	97
5-90.Crear el servicio. . . . .	98
5-91.Explorador de datos Cosmos DB. . . . .	99
5-92.Crear container en Cosmos DB. . . . .	99
5-93.Formulario para crear un container en Cosmos DB. . . . .	100
5-94.Configurar entrada de Stream Analytics. . . . .	100
5-95.Crear una entrada de IoT. . . . .	101
5-96.Crear Input centro de IoT. . . . .	102

<b>5-97.</b>	Servicios de Salidas en Stream Analytics. . . . .	102
<b>5-98.</b>	Salidas del Stream Analytics Blob Storage. . . . .	103
<b>5-99.</b>	Salidas del Stream Analytics s DB. . . . .	104
<b>5-100.</b>	Salidas del Stream Analytics Power BI. . . . .	104
<b>5-101.</b>	Consulta SQL de Salidas Cosmos DB. . . . .	105
<b>5-102.</b>	Consulta SQL de Salidas Blob Storage. . . . .	105
<b>5-103.</b>	Consulta SQL de Salidas Power BI. . . . .	106
<b>6-1.</b>	Diagrama de conexión HelTec ESP32 y el Sensor BMP280. . . . .	107
<b>6-2.</b>	GW y HelTec ESP32 - Sensor BMP280. . . . .	108
<b>6-3.</b>	Dispositivo conectado al TTN. . . . .	109
<b>6-4.</b>	Consulta SQL para filtrar los datos. . . . .	109
<b>6-5.</b>	Ejecutar servicio Stream Analytics. . . . .	110
<b>6-6.</b>	Ejecución del servicio Stream Analytics. . . . .	110
<b>6-7.</b>	Tabla de datos en archivo csv. . . . .	111
<b>6-8.</b>	Base de datos de Cosmos DB. . . . .	111
<b>6-9.</b>	Conectar los servicios de Cosmos DB y Power BI. . . . .	112
<b>6-10.</b>	Datos de conexión del servicio de CosmosDB. . . . .	113
<b>6-11.</b>	Conexión de CosmosDB y PowerBI. . . . .	113
<b>6-12.</b>	Transformar datos en PowerBI. . . . .	114
<b>6-13.</b>	Filtrado de datos específicos en Power BI. . . . .	114
<b>6-14.</b>	Tabla de datos en PowerBI desde CosmosDB. . . . .	115
<b>6-15.</b>	Gráfica Temperatura por Tiempo. . . . .	115
<b>6-16.</b>	Gráfica de Pastel Temperatura por Tiempo. . . . .	116
<b>6-17.</b>	Gráfica de la Temperatura por Altitud. . . . .	116
<b>6-18.</b>	Gráfica Presión por Altitud. . . . .	117
<b>6-19.</b>	Ubicación geográfica ESP32. . . . .	117
<b>6-20.</b>	Creación de un Webhook para Cayenne. . . . .	118
<b>6-21.</b>	Seleccionamos el template de Cayenne. . . . .	118
<b>6-22.</b>	Formulario para la creación de Cayenne. . . . .	119
<b>6-23.</b>	Creacion correcta de Cayenne. . . . .	119
<b>6-24.</b>	Visualización por gráficos de las mediciones en Cayenne. . . . .	120
<b>6-25.</b>	Gráfico de la temperatura en Cayenne. . . . .	120
<b>6-26.</b>	Gráfico de la presión en Cayenne. . . . .	121
<b>6-27.</b>	Localización del dispositivo IoT en Cayenne. . . . .	121
<b>6-28.</b>	Tabla de datos enviados por el sensor en tiempo real en Cayenne. . . . .	122
<b>A-1.</b>	Código en Arduino para cargar al dispositivo. . . . .	135
<b>A-2.</b>	Código para obtener el ChipID del dispositivo. . . . .	136
<b>A-3.</b>	Código de la biblioteca Cayenne. . . . .	137
<b>A-4.</b>	Código para obtener el BUS del dispositivo. . . . .	138

<b>B-1.</b> Logging PowerBI Robots. . . . .	139
<b>B-2.</b> Pagina de inicio PowerBI Robots. . . . .	140
<b>B-3.</b> Pagina para ver My Id. . . . .	140
<b>B-4.</b> Ventana de PowerBI Robots Agent. . . . .	141
<b>B-5.</b> Conexión con PowerBI. . . . .	141
<b>B-6.</b> Iniciar servicio PowerBI Robots. . . . .	142
<b>B-7.</b> Formulario para crear Reportes. . . . .	142
<b>B-8.</b> Correo de Recepcion. . . . .	143
<b>B-9.</b> Visualización datos de PowerBI. . . . .	143
<b>B-10</b> Crear Reporte en PowerBI Robots. . . . .	144
<b>B-11</b> Ejecución de TestPlaylistExecution. . . . .	144
<b>B-12</b> Carpetas de Reportes. . . . .	145

# Lista de Tablas

4-1. Frecuencias para las regiones de LoRa [15] . . . . .	12
4-2. Características principales de LoRaWAN [4] . . . . .	17
4-3. Características principales de LoRaWAN, Sigfox y NB-IoT [16] . . . . .	20
4-4. Características de Kit wAP LoRa8 [17] . . . . .	30
6-1. Pines de Conexión. . . . .	107
7-1. Cronograma de actividades . . . . .	123
8-1. Presupuesto para el proyecto. . . . .	124

# 1. Introducción

El Internet de las Cosas (Internet of Things - IoT), es una definición que ha venido evolucionando desde una sencilla interconexión digital de dispositivos de uso diario a través de internet, a una interrelación inteligente de dispositivos, personas y procesos, obteniendo así sistemas que sean sostenibles por otras tecnologías como la inteligencia artificial, la analítica de datos y que tengan la aptitud de interactuar por cuenta propia con otros sistemas y de poder tomar decisiones sin la intervención de personas. IoT provocará una disrupción considerable en el empleo de la sociedad, ocasionado fundamentalmente por la automatización de procesos, por lo que se exigirá personal más capacitado, aparecerán más empleos en el área que involucran investigación y desarrollo, mencionando como ejemplos la fabricación de dispositivos, programación para el desarrollo de software, implementar soluciones, etc. Para este tipo de trabajo se requerirán un alto nivel de formación y destreza, se crearán nuevos roles, nuevas necesidades; los empleos serán mucho más especializados, lo que significa que serán bien remunerados, pero también generará desempleo en el sector de menor conocimiento técnico excluyendo a quienes no estén preparados profesionalmente, tal como se describe en [18].

Según Cisco en la referencia [19], se pronostica que 29.300 mil millones de dispositivos estarán conectados con el paradigma de Internet de las cosas (IoT) para el 2023. Por otro lado, la Internet industrial de las cosas (IIoT) y la conectividad de máquina a máquina (M2M) son directamente conceptos interconectados y son actores clave en el siguiente paso de la evolución de la Industria 4.0 y la fabricación inteligente. Estos conceptos (IoT, IIoT, M2M e IR 4.0) han cambiado el método de interacción entre las personas, los dispositivos y las máquinas que los rodean. Allanan el camino para construir infraestructuras conectadas de forma ubicua para soportar servidores de red, aplicaciones y servicios pioneros. Dichos paradigmas con características prometedoras son atractivos tanto para los clientes como para la industria. Recientemente, hemos sido testigos de dispositivos IoT masivos conectados a internet en varias soluciones y aplicaciones diseñadas a propósito. Los dispositivos IoT y M2M están conectados a través de tecnologías de red de área amplia (TTN) de bajo consumo de forma inalámbrica. Pueden reaccionar, detectar su entorno y encenderse en cualquier momento y en cualquier lugar para actualizar los datos en tiempo real en la nube. TTN están destinadas a resolver varios problemas con las tecnologías de redes de comunicación y de corto alcance existentes para abordar las aplicaciones de IoT. Los dispositivos masivos de IoT están conectados en diversas aplicaciones, como edificios inteligentes, medidores inteligentes, agricultura inteligente, redes capilares, atención médica remota, automóviles conectados, calles

inteligentes, seguridad y control del tráfico, fabricación remota, redes inteligentes, logística, seguimiento y gestión de flotas.

El autor en [20] menciona que Las Redes de Área Extendida de Baja Potencia (LPWAN) representan una evolución de las redes destinadas a IoT, ofrecen conectividad a varios actuadores y sensores. Con la diferencia de las redes de banda ancha móviles tradicionales, este tipo de redes no se enfocan en ofrecer altas velocidades de datos y baja latencia, sino más bien escalabilidad, bajo costo, cobertura extendida, eficiencia energética para los dispositivos finales. Es importante destacar las tecnologías de redes disponibles hoy en día:

- Redes LAN (Local Área Network). Se usa para redes locales que abarcan sitios específicos y concretos de poco alcance, como por ejemplos una casa, oficina o un edificio corporativo, entre otros. Son de tipo Ethernet cableados y con acceso a WIFI.
- Redes WAN (Wide Área Network). Su empleo se da en redes de gran escala, pudiendo ser nacionales o internacionales, de tipo Ethernet en cada oficina, con routers e infraestructura para la conexión a las grandes redes internacionales.
- Redes PAN (Personal Área Network). Nacen con la necesidad de integrar una pequeña red personal que integre dispositivos como auriculares, teléfonos móviles, entre otros, típicamente se conectan con Bluetooth.
- Redes LPWAN (Low Power Wide Área Network). Pesando para la comunicación a todas las cosas que quedan fuera de las definiciones anteriores, como sensores y en general cualquier aplicación de categoría IoT.

Como se describe en [15], Bajo Alcance (Low Range - LoRa) fue desarrollado en Europa y no debe ser confundida con LoRaWAN, este pertenece a una asociación abierta sin ningún tipo de lucro, llamada LoRa Alliance cuyo principal objetivo es desarrollar una tecnología para redes de tipo IoT. A continuación describimos los objetivos funcionales de LoRa:

- Bajo costo.
- Comunicación de largo alcance (10 a 20 km en condiciones óptimas).
- Conexiones punto a punto
- Bajo consumo de batería (puede durar hasta años).
- Tolerancia a las interferencias
- Alta sensibilidad.

LoRa es solo un estándar de comunicación y transporte de mensajes entre nodos, pero es muy importante entender que no gestiona una red de transporte de alto nivel como el IP (Internet Protocol).

Como se cita en [21] , LoRaWAN es un estándar público de acceso de redes que se basa en la tecnología LoRa con características muy aceptables como seguridad, encriptación y transporte mediante Gateway (GW) a través de redes en estrella, públicas o privadas. Algunas operadas están en análisis para ofrecer conexiones LoRaWAN públicas con tarifas bajas y se considera que las redes 5G ofrecerán la conexión LoRaWAN bajo ciertos criterios, pero aún están por determinarse.

Para poder administrar los nodos es necesario implementar un servidor de red que nos permita administrar y gestionar la red LoRaWAN y los dispositivos IoT para esto se empleó la plataforma The Things Network (TTN) como se menciona en [5], ya que se destaca por tener varias herramientas capaces de manejar IoT. Además, se hace uso del servidor de aplicaciones de Azure IoT Hub el que nos brinda una comunicación altamente confiable y segura para dispositivos de IoT lo que menciona el autor en [22], siendo un recurso alojado en la nube al que podemos enlazar cualquier dispositivo IoT para obtener datos a través de los servicios que dispone, nos permitirá almacenar la información que nos envía los generó para finalmente poder filtrar los datos importantes y poder presentar los resultados para un posterior análisis y toma de decisiones.

## 2. Problema

La innovación de nuevas tecnologías en nuestro país es un paso muy importante para que las personas se involucren en la misma, una de ellas que se viene desarrollando a gran escala en la actualidad es la de los dispositivos inteligentes basados IoT, dentro de nuestro país no es muy conocida por lo que se puede estar desaprovechando por falta de conocimiento y siendo un gran beneficio en países más desarrollados, es por ello que con la elaboración de este proyecto técnico vamos a abrir paso a que se conozca más de esta tecnología analizando sus ventajas y desventajas.

Con TTN los datos capturados de los dispositivos finales no tienen una gestión adecuada, por lo que los datos quedan dispersos y no son almacenados en una base de datos, el cual facilitaría la administración de los mismos en casos de presentar fallas, conocer en que dispositivo se generó y también es necesario conocer la manera de almacenar los datos mediante un servidor de aplicaciones como el que vamos a emplear de Azure IoT Hub.

En la actualidad existen varias plataformas en la nube para la gestión de redes con dispositivos de IoT basadas en la tecnología LoRaWAN, una de ellas es la plataforma TTN; sin embargo, de ello, es necesario desarrollar un estudio sobre el estado del arte de LoRaWAN así como, analizar la plataforma de TTN para realizar el despliegue y configuración de una red LoRaWAN con la integración de dispositivos de IoT.

Al realizar la integración de dispositivos IoT, iniciando el proyecto se empleó una placa elaborada en una empresa nacional de Ecuador conocida como Tarpuq, ubicado en la provincia del Azuay en la ciudad de Cuenca, sector Chaullabamba, siendo estos dispositivos medidores de agua inteligente, pero los mismos presentaron problemas al realizar la configuración e integración de estos medidores al GW, ya que la plataforma de TTN no fue compatible con este tipo de dispositivos IoT.

Tratando el problema de compatibilidad de la plataforma de TTN, según la información recolectada se menciona que esta plataforma es compatible con varios dispositivos elaborados en países como EEUU, China y Japón, por lo que, la comunidad de usuarios están realizando pruebas de compatibilidad con otros tipos de dispositivos de fabricación distinta.

Los dispositivos que emplean la tecnología IoT para la red LoRaWAN no son muy utilizados



en la ciudad de Cuenca por lo que la adquisición de los mismos para trabajar en proyectos no son nada económicos y no se cuenta con personas capacitadas y con experiencia en la configuración y manipulación de los mismos.

Para poder recolectar los datos que envía el dispositivo de prueba IoT, es necesario contar con un servidor de aplicaciones capaz de poder manejar grandes cantidades de datos, el servidor de aplicaciones implementado en este caso es Azure IoT Hub, no obstante se debe tomar en cuenta que para trabajar con los servicios de Microsoft tiene un costo de paga, esto termina siendo un problema, ya que si se trabaja en un proyecto de bajo presupuesto no se le puede sacar mayor provecho a todos los beneficios que ofrecen los servicios de Azure IoT Hub.

El elevado costo de los servicios que ofrece Azure IoT Hub para este proyecto son muy altos, se usó una suscripción de Microsoft que ofrece la Universidad Politécnica Salesiana para los estudiantes pero con un crédito muy bajo y con el inconveniente de es que es muy limitado en algunos servicios que necesitamos, por lo que se usó otra suscripción de pago por tarjeta de crédito para poder acceder a más servicios pero con un presupuesto muy bajo, sin embargo, la recopilación y la visualización de datos en tiempo real se pueden mejorar con servicios con mejores características pero con tarifas de pago por uso más altas.

## 2.1. Antecedentes

Los autores en [23], mencionan que LoRaWAN es una tecnología que nos permite la comunicación con dispositivos inteligentes IoT, su impacto está en sectores como la agricultura, ciudades inteligentes, control de entornos remotos, monitoreo de infraestructura y redes inteligentes, por lo que es necesario trabajar en proyectos basados en esta tecnología y así ayudar al crecimiento tecnológico dentro de nuestra ciudad y en la sociedad.

Los autores en [24], TTN es una plataforma de pruebas abierta para poder registrar dispositivos IoT (nodos- sensores) y permite conectar el GW. Es una plataforma completa que cuenta con todos los servicios de back-end necesarios para la conexión y gestión de una estación base de LoRaWAN. Esta plataforma de red se encarga de recopilar, formatear y redirigir los datos de los nodos-sensores a nuestro servicio de aplicaciones en la nube, garantizando la integridad y seguridad de los datos.

Como se hace mención en [25], la tecnología de IoT se puede implementar, por ejemplo, en ciudades inteligentes, en casas y salud, entre otros. Por este motivo se hace uso de la plataforma de TTN como nuestro servidor de red que permite poder realizar la integración con plataformas externas. Gracias a estos beneficios se nos facilita las conexiones de nuestros dispositivos a un servidor de aplicación y de esta manera llevar redes LoRaWAN a diferentes plataformas de IoT en la nube como Azure IoT Hub, AWS e IBM Watson siendo estos los

más populares.

Según la referencia en [26], mediante un GW se busca dar paso al tráfico proveniente de los nodos clientes hacia la internet, donde todos estos datos podrán ser visualizados de forma histórica en los servidor de red TTN. La cual nos permite en la investigación entender o diferencia el trafico de datos a través de IoT y profundizar la salida de datos a través de histórico o gráficos.

Como hace mención en [27], la estructura básica de este tipo de redes cuenta con nodos, GW, servidor de red y aplicaciones. Los nodos recolectan la información y la envían al GW por medio de ondas de radio. Esta a su vez la transfiere al servidor vía Internet y desde el servidor llega, también gracias a Internet, a una aplicación donde el usuario puede consultarla.

Sobre la base de lo mencionado, es importante analizar las características que tiene TTN, que dispositivos son compatibles con esta plataforma, su funcionamiento, la codificación o modo de integración de cada dispositivo en TTN. Se considera que para este proyecto técnico se adquirió un dispositivo IoT LoRa HelTec ESP32, el cual cumple con el requisito de compatibilidad con la plataforma de TTN, esta alternativa nos ayudó a realizar las pruebas necesarias y probar el funcionamiento de la red LoRaWAN, se menciona que se hace el uso de este dispositivo como una opción a la no compatibilidad de los medidores Ultrasónicos de la empresa de Tarpuq.

En el desarrollo de este proyecto técnico, es necesario adquirir un equipo LoRaWAN (GW) con la finalidad de impulsar el estudio y desarrollo de la tecnología de dispositivos IoT. Dentro de la elaboración de este proyecto técnico, se pretende configurar y desarrollar el despliegue de un GW en la plataforma TTN, además de configurar un servidor de aplicaciones, en este caso se hace uso de Azure IoT Hub, el cual nos brinda una variedad de servicios para poder gestionar, visualizar y obtener información con el fin de que esta implementación se pueda usar y mejorar en investigaciones o proyectos futuros.

## **2.2. Importancia y Alcances**

Tiene como principal importancia el de ser un antecedente bibliográfico y técnico para futuras investigaciones ya que presenta una estructura técnica, conceptos fundamentales recopilados para ayudar a la comprensión efectiva de la configuración de redes TTN aplicado a la tecnología LoRaWAN.

Este proyecto podrá ser utilizado en las diferentes áreas profesionales para una excelente

gestión de redes de sensores inalámbricos, teniendo en cuenta su estructura, conexión y aplicación sobre a cuál se desarrollará y al ser una tecnología innovadora dar a conocer a más personas el uso que se le puede dar a las redes LoRaWAN con los dispositivos IoT.

La importancia que proporciona el manejo de esta tecnología LoRaWAN es que presenta muchas oportunidades para trabajar con dispositivos de IoT en sectores como son en la industria, biología, agricultura, ciudades inteligentes, hospitales, etc., todas estas áreas pueden beneficiarse de este proyecto técnico desarrollado.

Al elaborar este proyecto técnico basado en el uso de la plataforma de TTN, se pretende dar a conocer su funcionamiento, compatibilidad de la plataforma con dispositivos IoT, sus diferentes características, de esta manera se impulsa el desarrollo tecnológico en dispositivos de comunicación para IoT.

El presente proyecto cumplirá una serie de parámetros los cuales se harán pruebas locales que permitirán visualizar el cumplimiento de las metas, no se realizara la implementación en un área extensa. se realizaran en un ambiente controlado logrando tener parámetros necesarios para la configuración de los distintos dispositivos tecnológicos. Por otra parte se busca configurar, implementar y integrar un servidor de Aplicaciones Azure IoT Hub que permita almacenar, visualizar y interpretar el comportamiento de los nodos así como recolectar información para generar pruebas y resultados, toda esta información recaudada pueda ser usada en investigaciones o proyectos.

## **2.3. Delimitación**

Lo que se planifica es puntualizar, diseñar, implementar y desarrollar una red LoRaWAN en la plataforma TTN que permita la creación y conexión de un GW con la integración de nodos finales para el uso que se vea conveniente en la Universidad Politécnica Salesiana.

### **2.3.1. Espacial**

El proyecto propuesto se desarrollará en diversos puntos en la Universidad Politécnica Salesiana de Cuenca para la medición de parámetros de un sensor de presión.

### **2.3.2. Temporal**

La captura y medición de datos que serán tomados en cuenta para el desarrollo del proyecto propuesto se contemplara los datos recolectado del sensor de presión.

### **2.3.3. Académica**

El proyecto técnico planteado cumplirá con lo exigido por la Universidad Politécnica Salesiana la cual el entorno al grado social, investigativo, documental, metodológica y la estructura de presentación para proyectos técnico.

# 3. OBJETIVOS GENERALES Y ESPECÍFICOS

## 3.1. Objetivo General

Desplegar y configurar una red LoRaWAN usando la plataforma The Things Network (TTN) para dispositivos de Internet de las Cosas (IoT).

## 3.2. Objetivos Específicos

- OE1. Estudiar los fundamentos del estado del arte de las redes LoRaWAN con el servidor TTN.
- OE2. Analizar las características de TTN para el despliegue de redes LoRaWAN.
- OE3. Diseñar y desarrollar el despliegue de la puerta de enlace (GW) y las aplicaciones en el servidor de TTN.
- OE4. Implementar y visualizar los datos de sensores TTN con la integración de Azure IoT Hub.
- OE5. Diseñar y ejecutar un plan de pruebas que permita la validación del funcionamiento de los nodos finales de IoT con la red LoRaWAN.

# 4. REVISIÓN DE LA LITERATURA O FUNDAMENTOS TEÓRICOS

## 4.1. Internet de las Cosas (Internet of Things - IoT)

Tal como se detalla en [18, 28], el IoT está definido como una red de objetos físicos interconectados a través de internet de manera remota, estos objetos interactúan vía sistemas embebidos, redes de comunicación, medios de computación y aplicaciones de la nube. Esto permite que los objetos puedan comunicarse entre sí y acceder a la información mediante internet y capturar, almacenar y recuperar datos e interactuar con los usuarios, también permite la conexión con otros sistemas y aplicaciones permitiendo crear ambientes cada vez más interconectados e inteligentes.

Como se describe en [18, 28, 29, 30, 31, 32, 33] varios especialistas en el tema de IoT proyectan que para el año 2030 se tendría algo así como 500.000 millones o 500.000 billones en el estándar americano de sensores interconectados, esto correspondería que cada ser humano le correspondería alrededor de 70 dispositivos. Esto implicaría grandes cambios entre dispositivos acortando distancias, creando procesos y con infraestructuras más inteligentes.

### 4.1.1. Elementos de IoT

Existen tres elementos o componentes los cuales interactúan entre sí, esto según [34, 28, 29, 35]:

- El hardware como sensores, actuadores (dispositivos que se encargan del control del sistema) y otros dispositivos para la comunicación alojados en los objetos.
- La plataforma de Middleware este permite el intercambio de información entre las aplicaciones, así como también facilita las herramientas necesarias para el análisis de los datos.
- Herramientas que permiten la visualización e interpretación de la información y que son diseñadas para ser gestionadas por diferentes aplicaciones y dispositivos.

### 4.1.2. Características de IoT

Los autores en [36, 28, 37, 38, 39] describen las principales características que implementa IoT, mismas que se citan a continuación:

- **Detección y Actuación:** Se basa en que los objetos poseen inteligencia embebida y disponen de sensores para la detección y medición de variables físicas y actuadores para ejecutar sobre otros elementos de su entorno.
- **Comunicación:** Se establecen vínculos de comunicación entre los objetos se emplea protocolos inter-operables con otros objetos o aplicaciones estos están disponibles según los requerimientos de los usuarios.
- **Procesamiento:** Toda la data obtenida por los diferentes objetos es procesada en centros de datos tanto locales o en nubes computacionales, son almacenados en bases de datos y se analizan para obtener datos útiles para los usuarios.
- **Aplicación:** Toda la información se debe presentar a los usuarios finales de forma gráfica mediante diferentes aplicaciones de software que son desarrollados para terminales computacionales fijos o móviles como se describe en [37, 36, 40, 41, 42].

## 4.2. Red de Área Extendida de Baja Potencia (Low Power Wide Area Network - LPWAN)

Tal como se detalla en [20], LPWAN parten de la evolución de las redes inalámbricas de sensores que se aplican en IoT, generando de esta manera la conectividad de los sensores a través de internet. Este tipo de redes no enfocan en asegurar una alta tasa de transferencia de datos o de brindar una baja latencia, sin embargo, se encarga de brindar escalabilidad, mayor rango de comunicación, menos costo y asegura la eficiencia del consumo de energía.

Una de las ventajas tal como se describe en [43], es la de garantizar una muy buena comunicación de decenas de kilómetros de distancia y una duración de la batería de hasta 10 años, la tecnología de LPWAN es perfecta para lograr los más bajos costos de mantenimiento. Funciona en las bandas de frecuencia sin licencia ISM (Industrial, Scientific and Medical; son bandas reservadas internacionalmente para uso no comercial) según en la región que se opera. Algunos de estos protocolos son los de comunicación como: SigFox, LoRa, NB-IoT, etc.

### 4.3. Red de Largo Alcance (Long Range - LoRa)

Un gran salto a la tecnología sigue y seguirá siendo el IoT, actualmente no todas las ubicaciones donde existen procesos que se debe mejorar se encuentran bajo la señal de cobertura de internet (sea por cable o de manera inalámbrica), por lo que debería haber un dispositivo concentrador conocido como GW que pueda tener acceso a internet y pueda comunicarse con estos dispositivos de manera remota, logrando así integrar dispositivos desde cualquier lugar a sistemas que estén conectados a internet y para esta comunicación existe una tecnología que ha logrado protagonismo conocido como LoRa. Esta tecnología permite la comunicación inalámbrica con bajo consumo y a grandes distancias (kilómetros) siendo una excelente opción para proyectos de IoT.

Tal como se describe en [15], LoRa es una tecnología de radiofrecuencia la cual permite comunicaciones a larga distancia hasta en kilómetros, con un bajo consumo de energía, en cuanto a las frecuencias de funcionamiento utiliza sub-giga Hertz (debajo de 1 GHz) con bandas dedicadas según a la región, como se detalla en la siguiente tabla.

Región	Banda
Estados Unidos y Américas	De 902 MHz hasta 928 MHz
Europa	De 863 MHz hasta 870 MHz
China	De 779 MHz hasta 787 MHz

**Tabla 4-1.:** Frecuencias para las regiones de LoRa [15]

Como se menciona en [44, 45], la cobertura de alcance de un GW en una red LoRa es aproximadamente de hasta 20km en un área rural y en un área urbana es de 5km. La tecnología LoRa permite el tráfico de datos de tres maneras como se detalla en [43]:

#### Modulación LoRa.

En la siguiente cita [46, 45], se establece que la comunicación de LoRaWAN es a través de las puertas de enlace en las cuales se usa la tecnología LoRa, está basada en un esquema de modulación la cual es una adaptación de la modulación de CSS (Chirp Spread Spectrum), la cual fue diseñada en la década de 1940 para radares, se usó en comunicaciones militares y son seguras debido a sus requisitos de potencia de transmisión es relativamente bajo y con una alta robustez a los efectos de la degradación del canal, permitiéndonos la conectividad de largo alcance usando una técnica de espectro ensanchando.

Según los autores en [47] LoRa es una modulación de RF (Radio Frecuencia) la cual corresponde a la capa física en el modelo de referencia OSI, a diferencia de LoRaWAN que es un



estándar de la capa MAC que se encarga de la coordinación en el medio.

### De-modulación LoRa.

En este paso lo que se va a realizar es desmodular LoRa eliminando el chirrido de la señal para averiguar qué símbolos envió el remitente. La facilidad es que tanto la modulación como la de-modulación son muy simples que se puede llevar a cabo en el nodo final como en el GW.

#### 4.3.1. Red de Área Extensa de Largo Alcance (Long Range Wide Area Network - LoRaWAN)

LoRaWAN es un protocolo de comunicación inalámbrica patentado por LoRa Alliance para satisfacer a los diferentes desafíos que enfrentan las comunicaciones de largo alcance en las aplicaciones de IoT, LoRaWAN está empleado bajo la tecnología LoRa lo cual permite operaciones a largas distancias, dar formatos a las tramas, mensajes de aprovisionamiento y gestión, varios mecanismos para la seguridad, gestión de dispositivos y más aspectos. La tecnología LoRaWAN está formada por una topología en estrella de un salto alrededor de las puertas de enlace, estos actúan como los reenvíos de paquetes entre los dispositivos finales o nodos tal como se detalla en [21, 45].

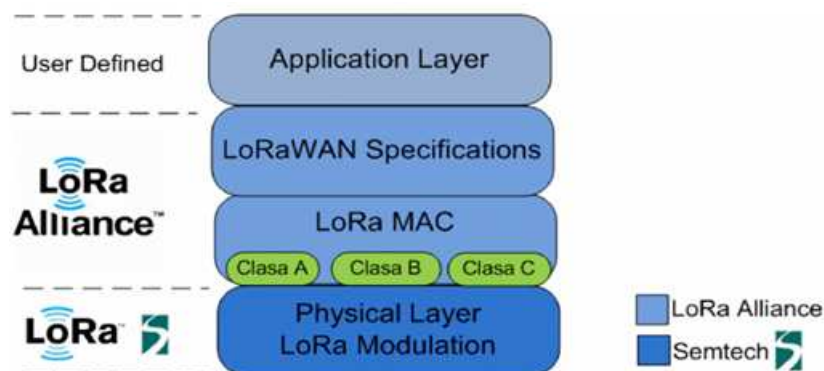


Figura 4-1.: Protocolo de Comunicación LoRaWAN [1]

### Arquitectura de una red LoRaWAN.

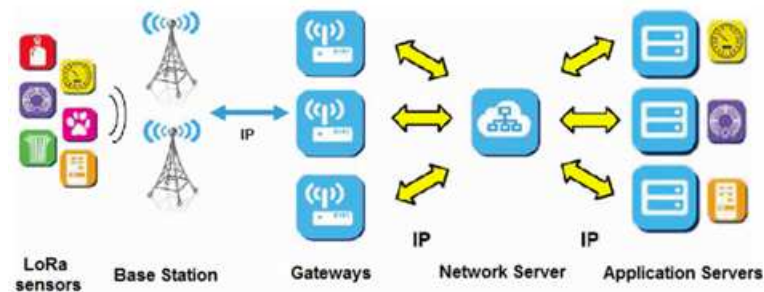


Figura 4-2.: Arquitectura de la LoRaWAN [2]

Como se hace mención en [47], LoRaWAN es un protocolo de capa MAC el cual se encarga de resolver los problemas de gestión del medio y la congestión de la red.

Una red LoRaWAN está compuesta por varios componentes como son los dispositivos finales, GW, servidor de red y aplicaciones que se describe en [4].

- **Dispositivos Finales(nodos):** Establecen la comunicación por el GW utilizando la tecnología LoRaWAN.
- **Gateway:** Se encargan de enviar las tramas LoRaWAN desde los nodos hasta el servidor de red mediante una interfaz de retorno con un gran rendimiento, por lo general Ethernet, 3G/4G, satelital o wifi.
- **Servidor de red:** Se encarga de la decodificación de los paquetes que son enviados por todos los dispositivos, realiza los controles de seguridad y la tasa de datos adaptables, de esta manera genera paquetes que deben de enviarse de regreso a los dispositivos.
- **Aplicación:** Se encarga de recibir todos los datos desde el servidor de red, la decodificación de los paquetes de seguridad y usar esta información para decidir qué acción llevará acabo la aplicación.

### Clases de dispositivos Finales

- **Clase A (bidireccional):** Todos los dispositivos cuentan con una ventana de que se encarga de la transmisión de enlace ascendente programada, que le siguen dos ventanas de enlace descendentes breves, cuentan con bajo consumo de energía y una alta latencia al enviar/recibir los mensajes.
- **Clase B:** Dispositivos con ventanas de enlace descendente que son programadas adicionalmente, con un consumo de energía medio, de baja latencia en él envío/recepción de los mensajes de unicast y multicast.

- **Clase C:** Dispositivos que utilizan ventanas para la recepción de manera continua, parecido a la de los algoritmos de ventanas deslizantes, de gran consumo de energía, de latencia mucho más baja para enviar/recibir los mensajes de unidifusión y multidifusión tal como se hace mención en [4].

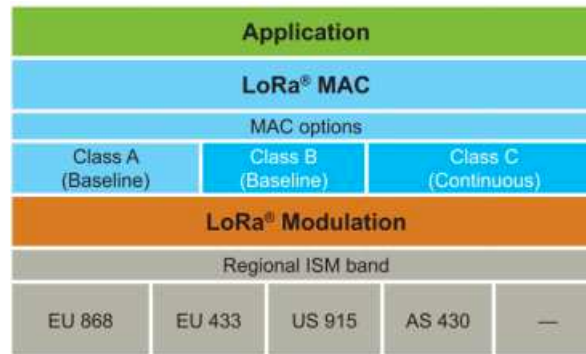


Figura 4-3.: Clases de dispositivos LoRaWAN [3]

### Modo de acceso a la red LoRaWAN

Las dos formas que existen de que el nodo se registre en una red LoRaWAN según lo que se menciona en [5] es:

**Activation by Personalization (ABP):** Se considera el modo más sencillo de conexión a la red y los parámetros que se usan para esta conexión son:

- **Network Session Key:** Clave de sesión empleada entre el servidor de la red y el nodo.
- **Application Session Key:** Clave de sesión empleada para la encriptación y des - encriptación del mensaje.
- **DevAddress:** Es la dirección lógica que se emplea durante toda la comunicación realizada.

Todos estos parámetros son usados por el nodo que se encarga de enviar los datos al GW el mismo que se encarga de validar que los datos correspondan a la sesión y que concuerden para que se procesen los datos caso contrario se los deben rechazar.

**Over-The-Air-Activation (OTAA):** Este modo de acceso más seguro de conexión a la red y en el cual se emplean los siguientes parámetros para su configuración:

- **DevEUI:** Es un identificador que hace que a cada dispositivo sea único.

- **AppEUI:** Es un identificador de aplicación único que consta de 64 bits, es utilizado para la clasificación de los dispositivos por cada aplicación a la que pertenece.
- **AppKey:** Es la clave AES de 128 bits que es compartida entre la red y el nodo la cual se emplea para determinar las claves de sesión seguras.

Para que se inicie el proceso de acceso a la red, el nodo envía un *join request* a la red con los datos de la configuración y abre una ventana de recepción que se encarga de recibir la respuesta de parte del GW. En el momento de que el GW recibe este join request, se le envía al servidor, el cual se encarga de verificar que el nodo que requiere acceder a la red y la llave de encriptación sean correctos. Si la información es la correcta, se le otorga una sesión temporal y por medio del GW le envía al nodo el que ya podrá enviar datos a la red, así se detalla en [5].

## Nodos

Como se menciona en [48] los nodos son dispositivos finales en una red LoRaWAN, se los conoce como un sensor o un actuador, estos dispositivos se encargan de transportar los datos de la tecnología LoRa por lo general leen dos variables, como pueden ser la temperatura, la presencia de agua o la humedad, los nodos son los que se encargan de generar los paquetes de datos con la información que necesitamos transportar hacia las aplicaciones finales en donde se le dará uso.

## Las Características de LoRaWAN.

- **Largo Alcance:** Tiene una cobertura de 5 y 15 Km lo que permite que un GW pueda proporcionar una gran cobertura a varios Kilómetros cuadrados e inclusive a una ciudad entera todo esto depende de los obstáculos que se atraviesen.
- **Tasa de transferencia:** Se puede obtener una tasa de transferencia desde 250 bps hasta 50 kbps.
- **Tiempo de vida de las baterías:** El tiempo de vida de las baterías alcanzan un promedio de entre 10 y 20 años, el funcionamiento de los nodos es asíncrona lo que permite transmitir información solo cuando sucede algún cambio en las condiciones físicas o ambientales en el lugar donde se encuentra los nodos ubicados.
- **Gran Capacidad de dispositivos en la red:** Se puede recibir múltiples mensajes simultáneamente desde los diferentes nodos todo esto gracias a que se logra disponer de una tasa de transmisión de datos adaptativa y transceptor a multicanal de la puerta de enlace.

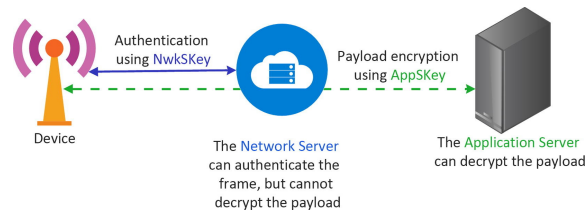
- **Escalabilidad:** Se puede implementar muchos GWs en la red dado que los requisitos y la estructura de la misma es sencilla y de igual manera se puede aprovechar mucho mejor la conexión tanto de los dispositivos antiguos con los nuevos.
- **Seguridad:** Se usa dos capas de seguridad la primera se encarga de la capa de red permitiendo comprobar la autenticidad del nodo dentro de la red, la segunda capa se encarga de implementar el protocolo de encriptación AES. para proteger los datos del usuario esto se menciona en [49].

Características	LoRaWAN
Topology	Star on Star
Modulation	CSS
Data Rate	290 bps - 50 kbps
Packet Size	154 dB
Battery lifetime	8 a 10 years
Power Efficiency	Very High
Security/Authentication	Yes (32 bits)
Range	2-5 km urban, 15 km suburban, 45 km rural
Interference Immunity	Very High
Scalability	Yes
Mobility/Localization	Yes

**Tabla 4-2.:** Características principales de LoRaWAN [4]

### Seguridad

Como se menciona en [2], LoRaWAN hace un gran énfasis en la seguridad de su red haciendo uso de algoritmos estándar los cuales son sometidos a varias pruebas lo cual permite una protección de extremo a extremo así mismo como la criptografía simétrica. Como se puede observar en la fig. 4-4, se implementan dos niveles de seguridad; uno que se encarga de la capa de red (autenticación del dispositivo) y el otro que se encarga de la capa de aplicación (el usuario solo puede leer los datos). Una de las principales características del sistema de seguridad LoRaWAN es la autenticación mutua, la protección de la integridad y la confidencialidad de los datos.



**Figura 4-4.:** Seguridad LoRaWAN [2]

### **Bandas de Frecuencia LoRaWAN**

Según los autores en [50], LoRaWAN trabaja bajo un espectro radioeléctrico sin licencia, con la gran capacidad de alcance de las frecuencias también presenta problemas en las restricciones que se presentan para cada país. Para poder solucionar esto y ser más eficiente LoRaWAN propone bandas para estas regiones, estas bandas son muy similares para poder soportar protocolos independientes para cada región, pero con una consecuencia en los sistemas backend. Las bandas de frecuencia más bajas son de 2,4 o 5,8 GHz esto permite lograr una mejor cobertura para los módulos y los dispositivos inalámbricos LoRa en especial cuando los nodos se encuentran ubicados dentro de edificios.

### **Duración de la batería**

Como se detalla en [4] uno de los desafíos al cual se enfrenta la tecnología de IoT es al consumo de energía, ya que la mayoría de las cosas funcionan con baterías no recargables y los costos son muy elevados. Por lo que hay una gran necesidad de reemplazar estos recursos energéticos muy limitados a lo cual las aplicaciones ecológicas se convierten en un gran problema, ya que al alimentar millones de dispositivos IoT, estos dispositivos emplean baterías no recargables como fuente de alimentación de energía principal. Para poder maximizar el tiempo útil de las baterías de los dispositivos finales el servidor LoRaWAN se encarga del control de la salida de la Radio Frecuencia y a su vez de la tasa de esquema adaptativo para cada dispositivo final.

### **4.3.2. Sigfox**

Fue fundada por los franceses Ludovic Le Moan and Christophe Fourtet en el año 2010 con la visión de conectar todos los objetos de nuestro mundo físico con el universo digital, construyeron una red global dedicada a IoT que está basada en datos pequeños, de bajo consumo y de largo alcance que ofrece un servicio de extremo a extremo siendo uno de los pioneros de la próxima revolución del Internet.

Como se hace mención en [51], Sigfox es otra LPWAN que tiene popularidad y es una tecnología patentada que opera en 868 MHz en Europa y en 902 MHz en EE. UU, dado que esta

tecnología usa la tecnología llamada Banda Ultra Estrecha (Ultra NarrowBand - UNB) y usa la Modulación por Desplazamiento de Fase Binaria Diferencial (Differential Phase Shift Keying - DBPSK) que le da una característica fuerte ante factores como el ruido y la sensibilidad en amplias coberturas, con un alcance de conexión de dispositivos desde 10 hasta 50 Km con un bajo consumo de energía, configuración de hardware de bajo costo y fácil implementación.

El modelo de negocio es a través de licencias para los usuarios que quieran usar esta tecnología, esta licencia varía del tipo de contrato que se realice permitiendo la conexión de un número determinado de dispositivos y acceso libre al backend en la web.

Utiliza una topología tipo estrella, como se ve en la fig. 4-5, en donde los distintos nodos finales se conectan a la estación base, pero la operadora es la que se encarga de construir la red para conexión de los dispositivos. Con la gran diferencia de que LoRaWAN tienes sus propios GW a disposición de los usuarios para que puedan generar su propia red sea una privada o publica dependiendo del caso de uso.



Figura 4-5.: Arquitectura de Sigfox [4]

### 4.3.3. IoT de Banda Estrecha (Narrow Band IoT - NB IoT)

Como describe en [16], NB-IoT es una tecnología LPWAN que puede convivir con la Evolución a Largo Plazo (Long Term Evolution - LTE) o el Sistema Global para Comunicaciones móviles (Global System for Mobile communications - GSM) estandarizado para la comunicación inalámbrica para IoT permitiéndonos conectar dispositivos que necesitan pequeñas cantidades de datos con poco ancho de banda y una batería de larga duración.

### 4.3.4. Comparación entre LoRaWAN, Sigfox y NB-IoT

En la siguiente tabla se destacan las tecnologías en aspectos técnicos.

	Sigfox	LoRaWAN	NB-IoT
Modulation	BPSK	CSS	QPSK
Frequency	Unlicensed ISM bands (868 MHz in Europe, 915 MHz in North America, and 433 MHz in Asia)	Unlicensed ISM bands (868 MHz in Europe, 915 MHz in North America, and 433 MHz in Asia)	Licensed LTE frequency bands
Bandwidth	100 Hz	250 kHz and 125 kHz	200 kHz
Maximum data rate	100 bps	50 kbps	200 kbps
Bidirectional	Limited / Half-duplex	Yes / Half-duplex	Yes / Half-duplex
Range	110 km (urban), 40 km (rural)	5 km (urban), 20 km (rural)	1 km (urban), 10 km (rural)
Interference immunity	Very high	Very high	Low
Authentication and encryption	Not supported	Yes (AES 128b)	Yes (LTE encryption)
Handover	End-devices do not join a single base station	End-devices do not join a single base station	End-devices join a single base station
Allow private network	No	Yes	No
Standardization	Sigfox company is collaborating with ETSI on the standardization of Sigfox-based network	LoRa-Alliance	3GPP

Tabla 4-3.: Características principales de LoRaWAN, Sigfox y NB-IoT [16]

## 4.4. The Things Network (TTN)

En [5], se indica que TTN es una red del IoT mundial gratuita, descentralizada, abierta y cualquier persona puede contribuir sus conocimientos sobre IoT con el objetivo de mejorar cada vez la plataforma. Fue creado en verano del 2015 en Ámsterdam por los fundadores Johan Stokking y Wienke Giezeman, entre otros y cuya finalidad es facilitar la creación y diseño de redes. Esta plataforma da soporte a GWs que están distribuidos por todo el mundo como se presenta en la fig. 4-6 y además posee la infraestructura de backend para sus consumidores, en resumen, implementa funcionalidades tanto de servidor de red y servidor de aplicaciones.

El desarrollo de esta plataforma es open-source, desde el repositorio de GitHub se puede acceder a los códigos con los que fue construido el backend, ya que son públicos. Aproximadamente TTN tiene registrados 9000 GWs disponibles en todo el mundo y están registrados alrededor de 80000 usuarios. Por lo que está en continuo crecimiento de la agrupación TTN y permite a muchas personas acceder de una manera muy fácil y gratis a un ambiente IoT donde podrán implementar sus ideas y realizar proyectos de menor o mayor complejidad.



Figura 4-6.: Cobertura de GW a nivel mundial [5]



### 4.4.1. Cayenne LPP (Low Power Payload)

Como se cita en [52], Cayenne es un generador de proyectos de IoT drag and drop (termino usado para interacción del usuario con la aplicación o programa) del mundo que permite a los desarrolladores, diseñadores e ingenieros crear rápidamente prototipos y compartir sus proyectos de dispositivos conectados, su uso es para redes LPWAN como LoRaWAN.

Está diseñado para ayudar a crear prototipos IoT y luego llevarlos a producción. Usa widgets personalizables para la visualización de datos, configurar reglas, programa, eventos entre otras cosas. Desde TTN se puede integrar esta aplicación desde el menú de Webhooks.

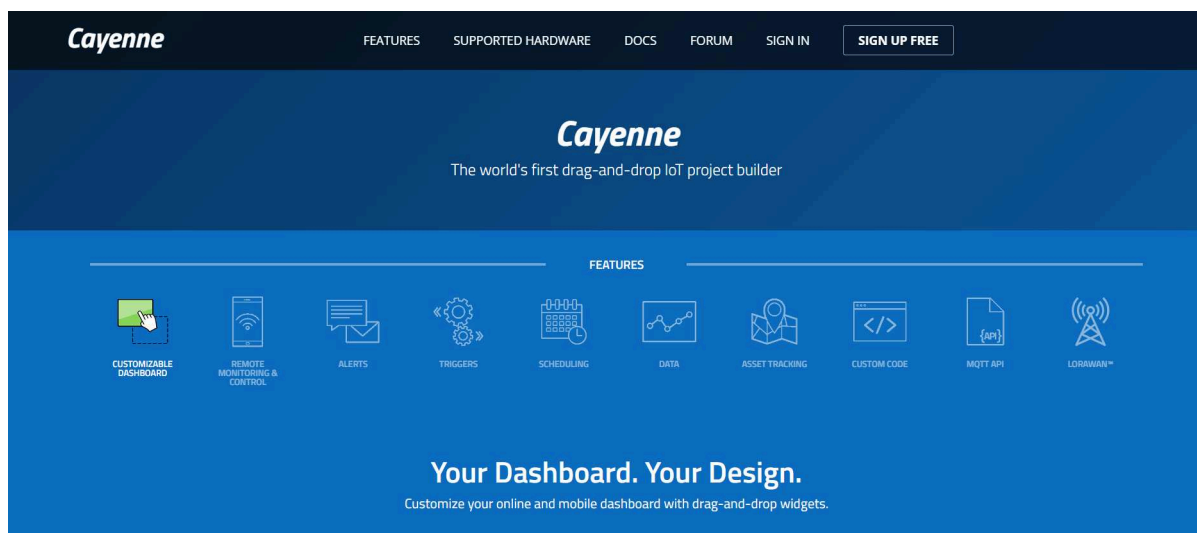


Figura 4-7.: Página web de Cayenne [6]

## 4.5. Microsoft Azure IoT Hub.

Como se detalla en [22], en febrero de 2016, Microsoft anunció la disponibilidad general de Azure IoT Hub, un servicio en la nube diseñado para las comunicaciones de dispositivo IoT a nube y de nube a dispositivo IoT. A través de IoT Hub, es posible realizar un seguimiento de la creación de dispositivos, las conexiones de dispositivos y las fallas de dispositivos. Se pueden admitir protocolos de comunicaciones adicionales mediante la implementación de puertas de enlace de protocolo Azure IoT personalizadas en la nube.

Actualmente, existen miles de dispositivos de diferentes clases generando una enorme cantidad de datos. Una porción de estos datos proviene de una solución de IoT y su conectividad para estos dispositivos son simplemente a través de una puerta de enlace en la nube y esta actúa como intermediario que recopila los datos de entrada y lo pone a orden para el siguiente paso que es el procesamiento con la intervención de otros servicios y procesos que

forman parte de la solución de IoT. Azure IoT Hub es un servicio administrado hospedado en la nube que actúa como un centro de mensajes central para la comunicación entre una aplicación de IoT y sus dispositivos conectados. Puede conectar millones de dispositivos con sus soluciones de back-end de manera confiable, segura y casi cualquier dispositivo se puede conectar a un centro de IoT.

#### 4.5.1. Azure Stream Analytics

Stream Analytics se sitúa en la categoría de PaaS (Platform-as-a-Service) donde los usuarios no necesitan administrar la infraestructura, pero si son responsables de la administración de la aplicación que se implementa. Es un motor de procesamiento para eventos en tiempo real diseñado para procesar grandes volúmenes de datos y su procesamiento es con baja latencia estimada en milisegundos y también posee una característica analítica para la transmisión de datos desde aplicaciones, dispositivos, sensores, entre otros. Este servicio es altamente escalable y administrado que brinda la opción de usar una puerta de enlace de campo, nube o IoT Hub para dispositivos heredados o de bajo consumo, como se cita en [7].

##### Capacidades y Beneficios

- **Confiable.** Posee funciones de recuperación integradas para la prevención o pérdida de datos, puede archivar eventos y volver a ejecutar el procesamiento para garantizar resultados sin pérdida de la información.
- **Escalable.** Esta construido y diseñado para manejar miles y millones de eventos por segundo (1GB/segundo) que son enviados por los dispositivos a la nube.
- **Facilidad de uso.** Para la creación de entradas y salidas es extremadamente sencillo, también nos ayuda a simplificar las transformaciones de datos mediante una variante de lenguaje SQL (Stream Analytics Query - Consulta de Análisis de Flujo) que es un tipo lenguaje para consultas a las bases de datos.
- **Conectividad.** Diseñado para la conectividad desde muchas fuentes y destinos, por ejemplo como IoT Hub, Event Hubs como el ingreso de información y para la salida o resultados destacamos a los servicios de Azure SQL Database, Cosmos DB, Power BI, entre otros.
- **Rentabilidad.** Proporciona una solución de análisis en tiempo real basado en un modelo de pago por uso, el costo se basa a la cantidad de unidades y de datos procesados.

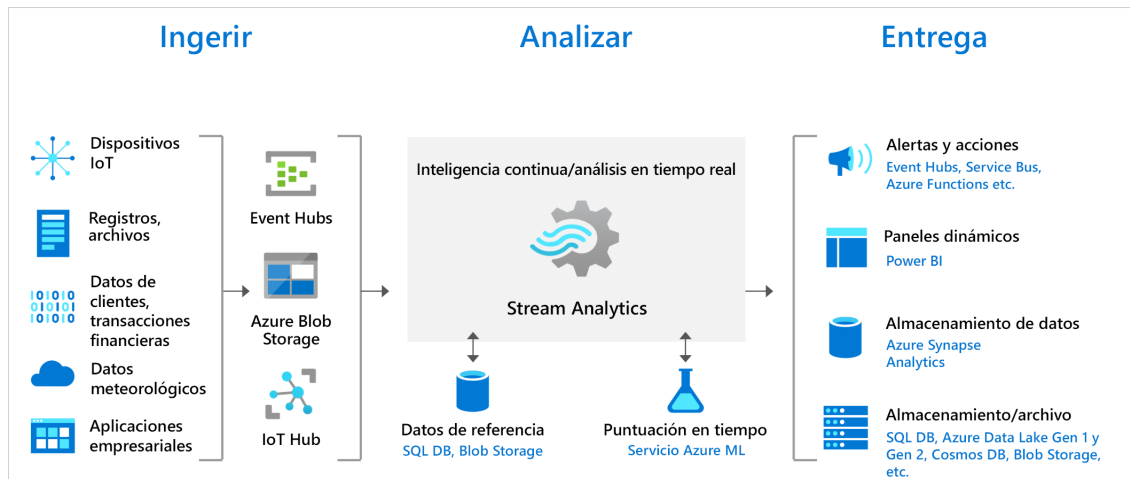
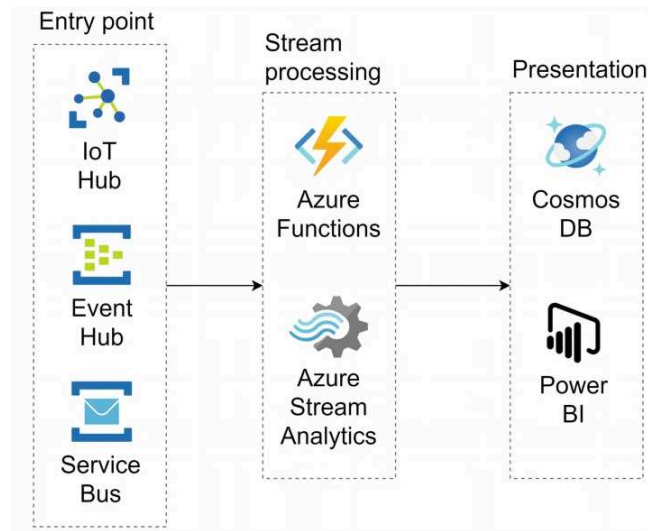


Figura 4-8.: Arquitectura de Stream Analytics [7].

#### 4.5.2. Cosmos DB

Tal como se menciona en [53] Azure Cosmos DB es un proyecto de Microsoft que comenzó en el año 2010, este proyecto tenía como objetivo solucionar algunos problemas con los que se enfrentaban los desarrolladores de Microsoft con aplicaciones de gran escala en internet. Para el año 2015 este proyecto se puso a disposición de desarrolladores externos en Microsoft Azure y dio paso a la creación de un nuevo producto con el nombre de DocumentDB. Pero finalmente en la conferencia de Microsoft Build 2017, se realizó el lanzamiento oficial de Azure Cosmos DB que viene incluido con las características existentes de DocumentDB, como una distribución global y a escala horizontal con baja latencia y un alto rendimiento.

Una de la ventaja de Azure Cosmos DB es de que admite de manera nativa varios modelos de datos como clave-valor, documentos, gráficos, columnas y otros recursos que están en desarrollo tal como se describe en [53]. Esto le permite trabajar con múltiples datos de la forma que el usuario lo crea conveniente. Otra característica es que es compatible con las API para acceder a datos, incluidos los DocumentDB, SQL, MongoDB, Apache Cassandra, Graph y Table.



**Figura 4-9.:** Arquitectura de Cosmos DB [8].

Para cargar los datos en Cosmos DB tal como se visualiza en fig. 4-8, desde un dispositivo IoT se lo puede realizar mediante programación o de algunas otras maneras como se describe en [54]:

- Se puede cargar los datos desde el motor en memoria de Databricks (parte en donde los datos se guardan automáticamente en Azure en IoT Hub para luego ser cargados en Databricks).
- Con la creación de procedimientos almacenados y el uso de aplicaciones lógicas en un Event Grid implementado en la plataforma de IoT Hub que almacenan datos en Cosmos DB.
- Implementando funciones de Azure en el enrutamiento de mensajes de IoT Hub que manada a guardar los datos en Cosmos DB.

### 4.5.3. Blob Storage

Como mencionan los autores en [55], Azure Blob Storage es muy similar al de un sistema de archivos tradicionales. Este servicio de almacenamiento de Blogs, organizado en una jerarquía de carpetas, se la puede usar para el almacenamiento de una gran cantidad de datos no estructurales en diferentes formatos uno de ellos es CSV. Una cuenta de Blob Storage puede tener varios contenedores de blogs y un contenedor puede almacenar múltiples blogs. Los blogs están divididos en función de la combinación (nombre del contenedor – nombre del bloc) de esta manera cada bloc se puede almacenar en un servidor diferente y así obtener un mayor rendimiento, el rendimiento de un bloc es de equivalente de 60 MB por segundo.

Algunos factores importantes de Blob Storage son [9]:

- El servicio Blois está constituido por archivos simples estos pueden ser; doc, xls, archivos de registro, etc. Los cuales se almacenan en carpetas que son conocidas como contenedores y la configuración se la denomina almacenamiento de Bloc, al cual se puede acceder desde un sistema distribuido a través de la URL, interfaz REST o de las bibliotecas de almacenamiento del kit SDK (Software Development Kit - Kit de Desarrollo de Software) de Azure .
- En el servicio de archivos se hace uso del protocolo de bloque de mensajes del servidor SMB (Server Message Block - Bloque de Mensajes del Servidor) que es un estándar para poder acceder a los archivos de red que están disponibles.
- El servicio de cola es utilizado para poder almacenar y recuperar mensajes. Permitiendo a los mensajes un tamaño de cola de hasta 64 KB, una cola puede guardar millones de mensajes. Este servicio por lo general es utilizado para el almacenamiento asíncrono de listas de mensajes.

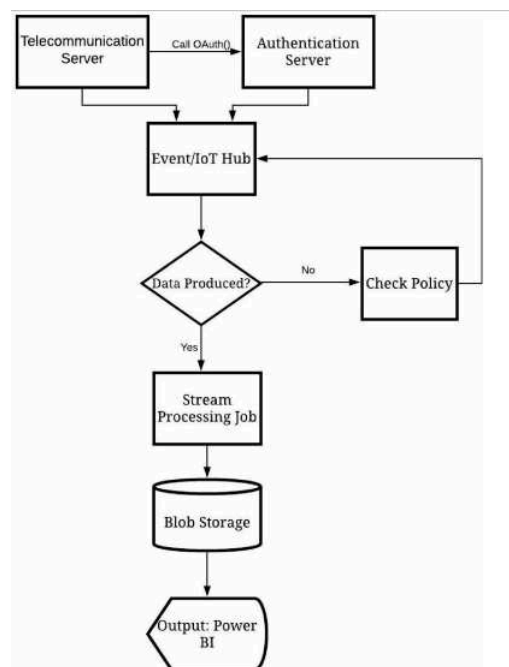


Figura 4-10.: Proceso de detención de mensajes Blob Storage [9].

#### 4.5.4. Power BI

Según los autores en [10], mencionan que el servicio de Power BI es una de las soluciones más novedosa de Microsoft y en el mundo del Business Intelligence en Cloud, a su vez es una

potente herramienta de análisis de los datos necesarios en cada momento de carga y obtener las mejores conclusiones y resultados para poder tomar decisiones rápidas y eficaces.

### Power BI Desktop

Es una solución de Business Intelligence la cual es muy importante y que permite crear informes y análisis sin tener un conocimiento básico de su uso. Incluye algunas funcionalidades que permiten conectar, dar forma y compartir perspectivas, ya que contiene un contenedor flexible que permite arrastrar y soltar contenidos y así poder mostrar resultados. Con Power BI Desktop se puede transformar datos, crear informes y visualizaciones eficaces y dicha información puede ser publicada fácilmente en Power BI tal como se describe los autores en [10].



Figura 4-11.: Visualización de Power BI [10].

## 4.6. Gateway LoRaWAN

Recordemos que una red LoRaWAN esta compuesto por tres elementos importantes, el GW, los nodos o dispositivos finales y el servidor de red (TTN). La función principal de un GW es crear la red inalámbrica LoRaWAN para dar cobertura a los dispositivos finales, poder comunicarse con los dispositivos para recolectar la información que estos envían y por último conectarse con el servidor de red a través de la comunicación TCP/IP.

En síntesis, el GW tiene la capacidad de integrar los equipos distribuidos (actuadores, instrumentación, sensores, etc) con el servidor de red que administrar la red y envía la información a otras aplicaciones de otros niveles tal como se menciona en [11].

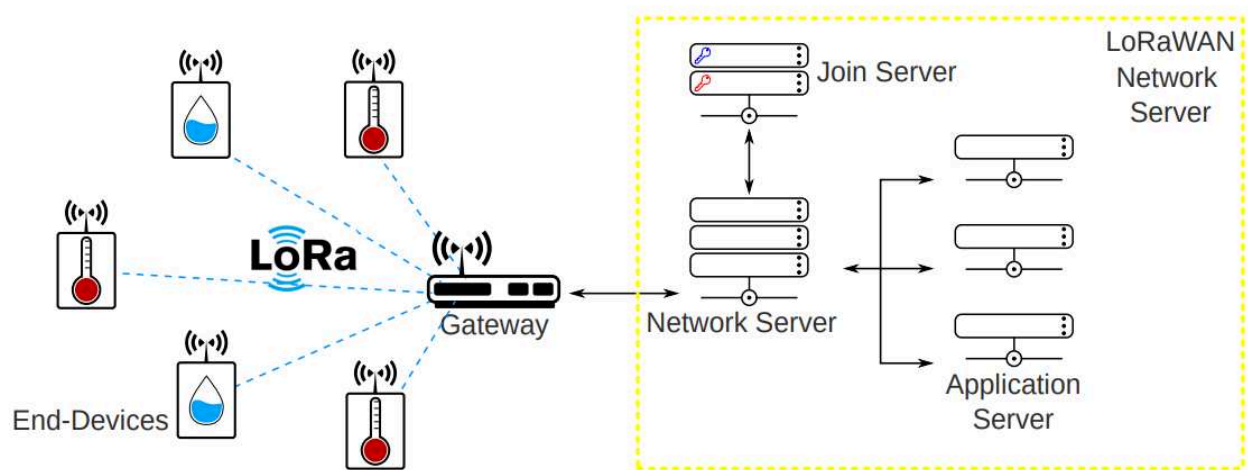


Figura 4-12.: Red GW LoRaWAN [11]

#### 4.6.1. Características de MikroTik RouterOS

Tal como menciona en [12], RouterOS es la combinación entre del sistema operativo y software de MikroTik lo que permite controlar y convertir el hardware normal de una PC o RouterBOARD en un router dedicado, se describen algunas de las características mas importantes:

- Compatible con 802.11a/b/g/n/ac
- QoS (Quality of Service - Calidad de Servicio) Avanzado
- Protocolo Nv2 TDMA personalizado
- Puente STP con filtrado
- Cortafuegos con estado, túneles
- HotSpot para acceso Plug-and-Play
- WDS y Virtual AP
- Interfaz gráfica de usuario remota de WinBox y Administración Web
- Enrutamiento RIP, OSPF, BGP, MPLS
- Vinculación de interfaces
- Alta disponibilidad con VRRP
- Telnet/mac-telnet/ssh/administrador de consola

- Compatibilidad con 3G/LTE
- Configuración y monitoreo en tiempo real
- Soporte para OpenFlow



Figura 4-13.: Consumidores internacionales de MikroTik [12]

La construcción del hardware son de pequeños circuitos integrados que ejecutan el software RouterOS, para poder ensamblar sus propios sistemas y accesorios para la empresa.



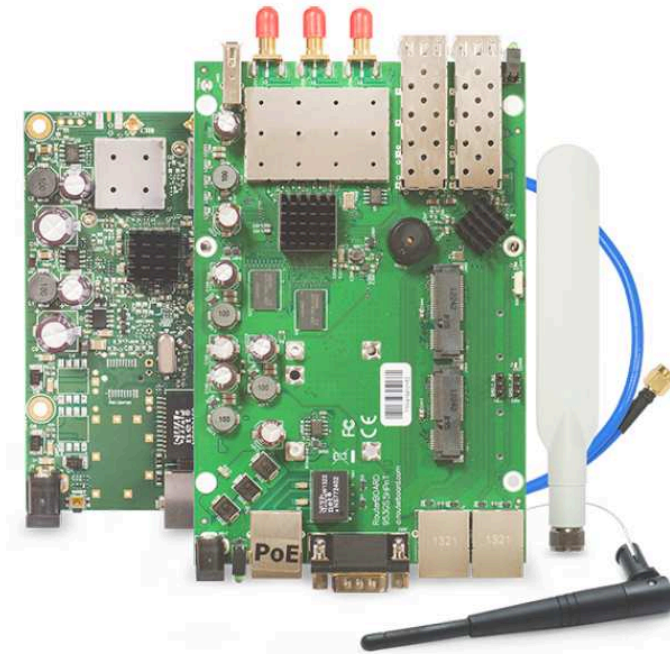


Figura 4-14.: Circuitos integrados de hardware [12]

### 4.6.2. Kit wAP LoRa8

Para realizar la configuración de GW se adquirió un equipo Kit wAP LoRa8 la cual es una solución que nos permite hacer uso de la tecnología LoRaWAN, este dispositivo contiene un envidador de paquetes UDP el que se encarga de la recolección del tráfico de la red. Este dispositivo de MikroTik ofrece un rendimiento excelente y la cual está apta para cualquier condición climática, también viene incorporado con una antena interna opcional de 2 dBi o también se puede conectar una antena externa LoRa.

#### Características de Kit wAP LoRa8

Es una solución lista para usar con la tecnología LoRa que contiene un reenvío de paquetes UDP preinstalado para cualquier servidor privado o público y un punto de acceso inalámbrico muy resistente a la intemperie para exteriores.

A continuación se detallan las características principales de este equipo:

<b>CPU/Frecuencia Nominal</b>	QCA9531/650 MHz
<b>RAM</b>	64MB
<b>Sistema Operativo</b>	RouterOS
<b>Ethernet</b>	10/100 Ethernet ports
<b>Wifi</b>	2,4 GHz 802.11b/g/n
<b>Concentrador LoRa</b>	R11e-LoRa8 MiniPCIE
<b>Potencia LoRa TX</b>	Hasta 20dBm
<b>Sensibilidad Lora</b>	-137dB @ SF12
<b>Dispositivos LoRa compatibles</b>	Clase A y C
<b>Consumo máximo de energía</b>	7 W
<b>Velocidad de datos inalámbrica máxima de 2,4 GHz</b>	300 Mbit/s
<b>Generación inalámbrica de 2,4 GHz</b>	WiFi 4

Tabla 4-4.: Características de Kit wAP LoRa8 [17]

## 4.7. Hardware Heltec Wi Fi Lora ESP32

Como se hace mención en [56], el módulo Heltec ESP32 es una placa desarrollada por Heltec que se basa en la tecnología LoRa la cual utiliza un SOC (Security Operations Center - Centro de Operaciones de Seguridad) que permite la mejora el trabajo de cálculo, procesamiento y utiliza el módulo transceptor LoRa para enviar y recibir mensajes a través de una interfaz aérea.

Características de este módulo son:

- MCU 240MHz Tensilica LX6 dual core.
- Memoria 520KB RAM.
- Conectividad Wi-Fi y BLE,.
- Bandas LoRa EU 863–870, AU-US 902–928.
- Pantalla OLED 128\*64 de 0,96 pulgadas.
- Flash SPI de 8 MB (64 M bits).

Como se menciona en [57] los módulos de ESP32 están dentro de los microcontroladores de chip de bajo costo, tamaño pequeño y baja potencia con módulos de comunicación múltiple integrados en una placa de circuitos. Tiene mucha diferencia a los microcontroladores que se usaban antes, ESP32 en realidad es un microcontrolador ideal para poder trabajar con dispositivos IoT. Debido al rendimiento electrónico, el precio y el tamaño pequeño, también

está considerado como una excelente opción para integrar y trabajar en proyectos de IoT.

Con el uso del dispositivo hardware de Heltec se puede conectar a diferentes sensores y los módulos de memoria no volátil en el Arduino Uno, dejando las tareas de comunicación en el módulo Heltec Wifi Lora 32 tal como se muestra en [13]. Para poder capturar datos debe ser programado en el IDE de Arduino o de Visual Studio Code. El uso de este dispositivo nos permite utilizar pines de conexión para facilitar el procesamiento de información tal como se ilustra en la fig. 4-15.

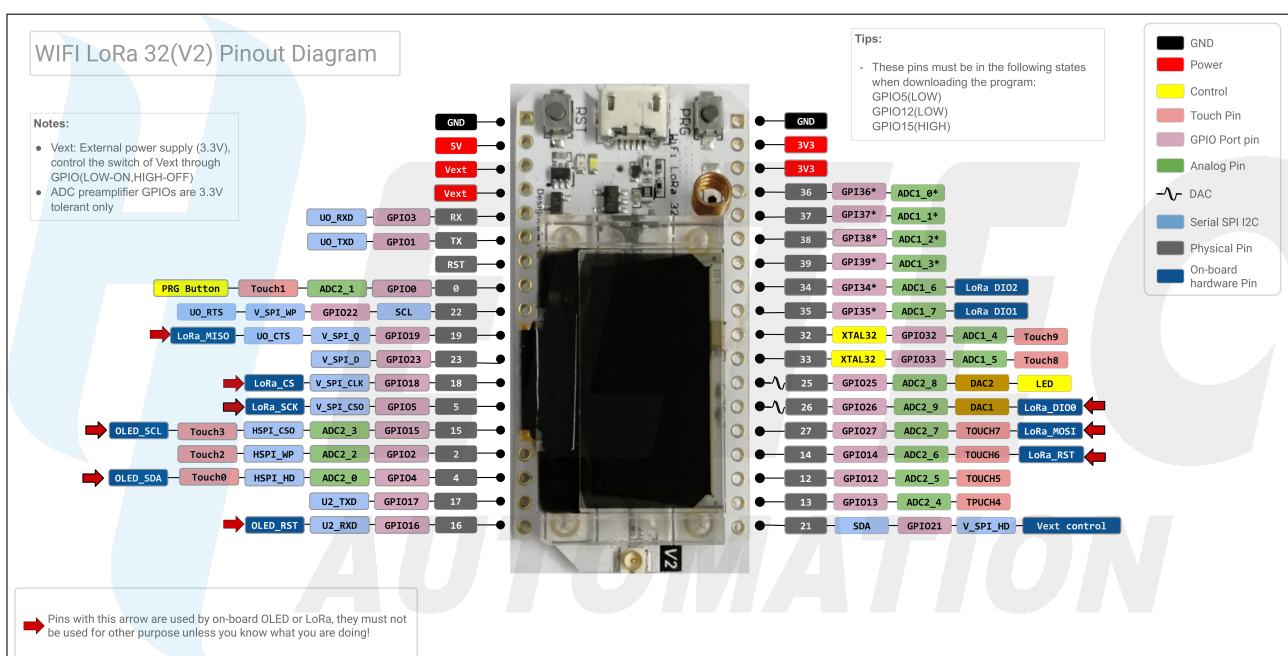


Figura 4-15.: Diagrama de Pines Heltec ESP32 LoRa 32(v2) [13].

#### 4.7.1. Bus de datos I2C

Como se menciona en [58], se trata de un circuito integrado interno que se lo conoce como un bus con múltiples maestros, por lo que se le puede conectar a varios chips al mismo bus y actúa como maestro, generando la comunicación y la transferencia de datos. Es un estándar que facilita la comunicación entre los microcontroladores y dispositivos como son sensores, dispositivo de IoT, para esto se hace uso de dos líneas de señal y un común, lo que permite el intercambio de información entre múltiples dispositivos a una velocidad aceptable de 100 kbits por segundos, aunque en ciertos casos especiales en los que el reloj llega hasta un máximo de 3.4 MHz.

## El Maestro

Es el encargado de controlar el cable de reloj, llamada SCL (Serial Clock), el cual se encarga de iniciar la comunicación. La información binaria serial se puede enviar solo por la línea de datos seriales llamada SDA (Serial Data) tal como se menciona en [58]. La comunicación puede funcionar de dos maneras como son , maestro-transistor y maestro receptor.

## SDA Y SCL

Son los pines de reloj y de datos I2C usado por el protocolo de comunicación serial el cual es utilizado en múltiples sensores y actuadores.

### 4.7.2. IDE Arduino para la programación HelTec ESP32

Una forma muy fácil de comenzar a codificar la placa ESP32 es usando la plataforma IDE de Arduino. Como se detalla en [13] esta IDE es de código abierto y está diseñada para la creación rápida de prototipos que se basa en microcontroladores y su entorno está creado en el lenguaje de Java.

Como menciona los autores en [59], la principal importancia de Arduino esta enfocado a la funcionalidad al momento de diseñar y programar en chips, es versátil y compatible con múltiples sensores y actuadores, posee una gran variedad de bibliotecas con las cuales se pueden trabajar para permitir las comunicaciones e interactuar con sensores, LCDs, motores, entre otros.

## 4.8. Sensor de Presión Digital BMP280

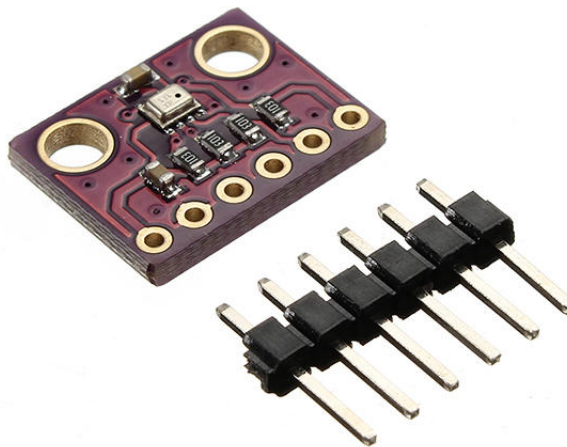
Tal como manifiesta en [60], Robert Bosch es el líder mundial del mercado en la fabricación de sensores de presión, aplicaciones automotrices y de consumo. BMP 280 es un sensor meteorológico sencillo y de buenas prestaciones que convierte con alta precisión en una señal digital la presión atmosférica, temperatura y humedad, este módulo está implementado en drones, aplicaciones móviles, estación meteorológica, GPS o relojes y se ha diseñado para ser conectado directamente a un microcontrolador a través de I2C o SPI, se puede observar en la fig. 4-16.

- **I2C (Inter Integrated Circuits - Inter Circuitos Integrados).** Es un protocolo de comunicación para Arduino usado en multitud de sensores y actuadores, su aparición se da por dos factores: con la llegada de los dispositivos IoT y el aprovechamiento de pines. Fue desarrollado por Philips a principios de los 80s.

- **SPI (Serial Peripheral Interface - Interfaz de Comunicación Serie)**. Es otro protocolo de comunicación utilizado para la intercomunicación entre si circuitos integrados, fue desarrollado por Motorola en el año de 1970 cuyo objetivo es reducir el número de pines que se necesita para conectar dos o más circuitos integrados.

Se detallan algunas de las especificaciones mas importantes de este sensor:

- Rango temperatura:  $-40\ 85^{\circ}\text{C}$
- Rango presión:  $300\ 1100\ \text{hPa}$  (hecto y pascal)
- Precisión de temperatura:  $\pm 1.0^{\circ}\text{C}$
- Presión del aire:  $\pm 1\ \text{hPa}$
- Consumo:  $2.7\ \text{A}$
- Nivel Lógico:  $3.3\text{V}$
- Interfaz: I2C, SPI



**Figura 4-16.:** Sensor de presión BMP280 [14]

# 5. MARCO METODOLÓGICO

## 5.1. Metodología de diseño de una Red LoRaWAN

Para realizar este trabajo de titulación se hace uso de dos tipos de metodologías como son el método experimental y el método deductivo.

### **Método Experimental**

Durante la formación en la universidad, los estudiantes de ingeniería en sistemas adquieren las habilidades y conocimientos necesarios para ser aplicados en el campo práctico para poder desarrollar cualquier aplicación en su campo de estudio, por lo que las pruebas experimentales se realizan en un ambiente controlado en la Universidad Politécnica Salesiana de Cuenca.

### **Método Deductivo**

La aplicación de este método nos permite realizar el estudio de los procesos sintético y analítico de la obtención de información generada por los nodos sensores, la cual puede ser visualizada mediante herramientas gráficas y generadoras de reportes.

Se deben cumplir con los siguientes pasos para el diseño de red LoRaWAN:

- **Recopilar toda la información necesaria.**  
Se debe recopilar toda la información generada por los nodos sensores, esta información nos permitirá obtener diferentes resultados.
- **Análisis de la red en el sitio a ser implementado.**  
Se debe analizar el tiempo de respuesta, tráfico, número de fallas, aplicaciones que se utilizarán, tipo de red, dispositivos a usar, etc.
- **Preparar un plan de diseño.**  
Cuáles son los puntos para que el diseño de red sea correcto con una mejor integridad, confiabilidad, costos, eficacia, etc.
- **Determinar el tamaño de la red y los servicios.**  
Se debe indicar si se pretende extender la red, por ejemplo en centros comerciales o ciudades.

- **Configuración de la red.**  
Modo de operación del hardware y software como se implementa la arquitectura y su gestión.
- **Evaluación de costos de la red.**  
Se analiza cuál es el costo de la red diseñada, así como la rentabilidad de la misma.
- **Estudios de viabilidad.**  
Hardware y Software necesarios, viabilidad económica costos y ventajas.
- **Administración.**  
Cuáles son medidas de control, la seguridad y la eficiencia de la red.

## 5.2. Diseño y Arquitectura de la red LoRaWAN con TTN y Azure IoT Hub.

Se propone implementar un prototipo de arquitectura de red como se muestra en la fig. 5-1, basado en la plataforma TTN para administrar dispositivos IoT, el cual va a utilizar como medio de transmisión inalámbrica la tecnología LoRaWAN entre los nodos y el GW, así mismo la recolección de la información o datos a ser interpretados se implementará un servidor de aplicaciones como es el caso de Azure IoT Hub, el GW se va a encargar de la conexión de los nodos LoRa tras lo cual, mediante su interconexión hacia internet, transmitirá esta información hacia el servidor IoT encargado de la administración.

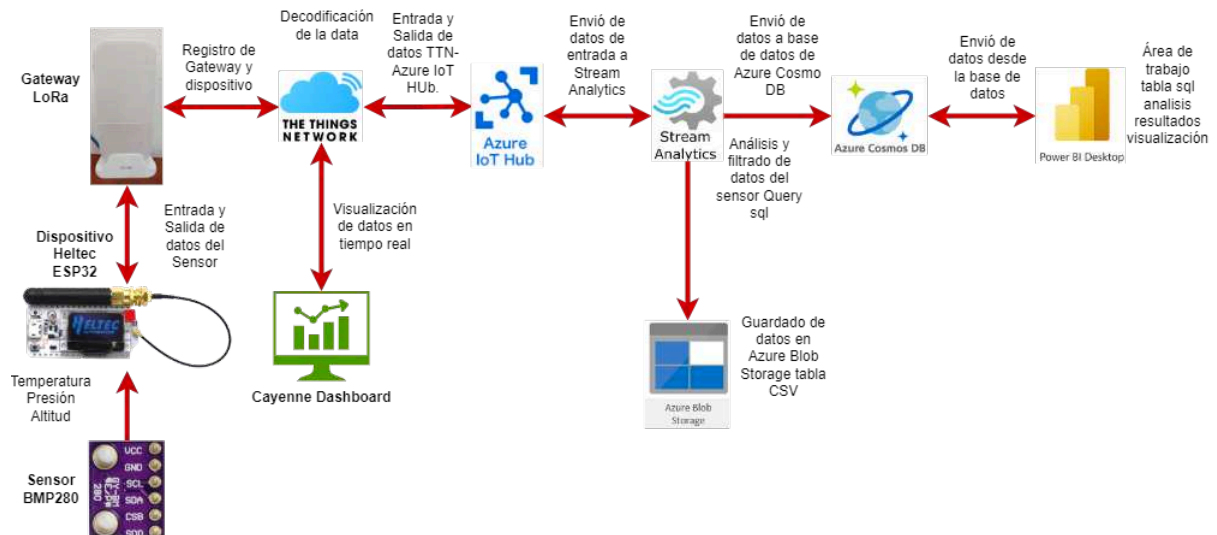


Figura 5-1.: Arquitectura de la red LoRaWAN.

**Sensor BMP280:** Se encarga de medir la temperatura, la presión atmosférica y la altitud estos datos pueden ser recopilados para proyectos basados en drones, aplicaciones móviles,

estación meteorológica, GPS o relojes entre otros.

**HelTec ESP32:** Es un dispositivo compatible con la tecnología LoRa para dispositivos IoT que cuenta con un microcontrolador de I2C y trabaja en la frecuencia de 902–928, mediante el cual podemos implementar la conexión varios dispositivos sensores mediante codificación en el IDE de Arduino para realizar diversas pruebas, el mismo que se encarga de la conexión con el sensor BMP280.

**Gateway Kit wAP LoRa8:** Este equipo se encarga de la comunicación desde el dispositivo LoRa HelTec ESP32 para poder realizar la conexión y comunicación de estos dos dispositivos deben tener conexión a Internet y trabajar en las mismas frecuencias, este GW también nos permite trabajar con diferentes servidores al mismo tiempo en este caso se implementó el servidor de red TTN.

**TTN:** Servidor de red el cual se encarga de administrar la conexión partiendo desde los nodos con el GW, para luego decodificar la data generada y enviarlo al servidor de aplicaciones Azure IoT Hub.

**Azure IoT Hub:** Es el servidor de aplicaciones que nos permitirá gestionar y monitorizar el comportamiento de los nodos, este servidor nos permite la integración con el servidor de red TTN, es un servidor que cuenta con muchos servicios adicionales que se los puede usar según sea necesario.

**Stream Analytics:** Para poder filtrar todos los datos que envía el nodo en tiempo real de múltiples dispositivos de entrada, como son: las aplicaciones, los dispositivos, los sensores, entre otros flujos de datos.

**Azure Blob Storage:** Nos permite almacenar los datos filtrados desde el Stream Analytics y almacenarlos en un formato CSV, esto mediante una consulta SQL, toda esta información se la puede analizar con diferentes herramientas.

**Azure Cosmo DB:** Es la base de datos que se encarga de almacenar los datos filtrados por el Stream Analytics, esto mediante una consulta SQL, estos datos pueden ser almacenados y ser usados para una mejor interpretación de resultados.

**Power BI:** Nos permite proporcionar una mejor experiencia para la visualización de los resultados, un mejor análisis mediante gráficos y reportes.

**Cayenne Dashboard:** Es una herramienta que viene integrado en el servidor de red de TTN, el mismo que es gratuito que posee características similares al del servicio de Power



BI, la diferencia es que este nos permite la visualización de los datos y resultados obtenidos en tiempo real.

## 5.3. Creación y configuración del GW, TTN, HelTec ESP32-Sensor BMP280, Azure IoT Hub.

### 5.3.1. Configuración del equipo wAP LR8 kit.

#### Paso 1. Configuración del equipo Mikrotik wAP LR8 kit.

- Conecte el equipo a la fuente de energía, para poder ingresar a la configuración principal del equipo debemos conectarnos a la red Wifi del equipo GW Mikrotik wAP LR8.
- Desde un navegador ingresamos al sitio web <http://192.168.88.1/> para empezar con la configuración del equipo.
- Una vez ingresado a la configuración del equipo se debe de proporcionar las credenciales de inicio de sesión como se observa en fig. 5-2.

**Login:** admin.

**Password:** admin.

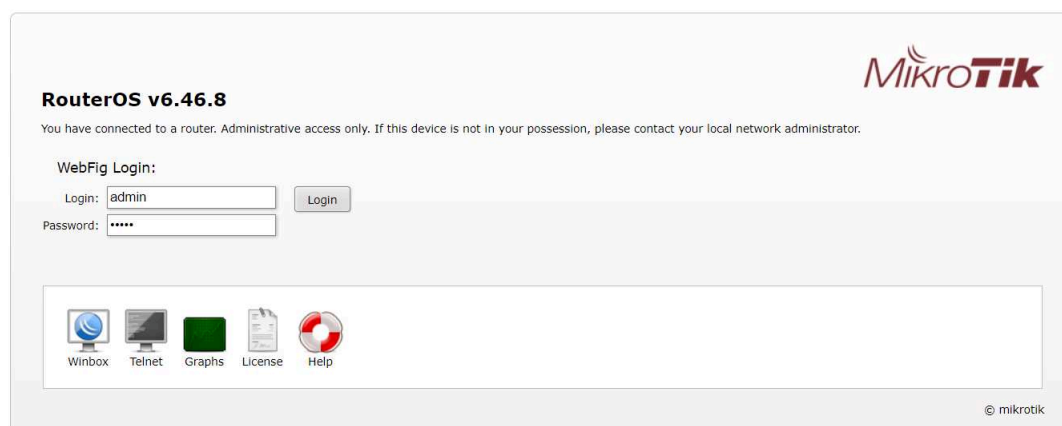


Figura 5-2.: Login del GW MikroTik wAP LR8 kit.

#### Paso 2. Configuración de un nuevo Servidor TTN en el GW MikroTik wAP LR8 kit.

- Ingresamos a la interfaz de MikroTik wAP LR8 kit y seleccionamos la opción **LoRa** como tal como se visualiza en la fig. 5-3.

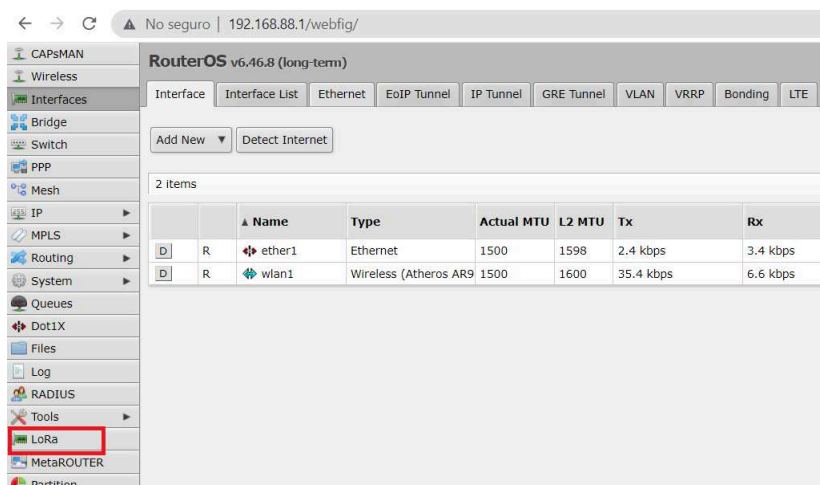


Figura 5-3.: Crear Servidor LoRa.

### Paso 3. Crear un nuevo Servidor TTN.

- Le damos Clic en **Servers** (Servidores de Red LoRaWAN), se pueden crear varios servidores LoRaWAN tal como se visualiza en la fig. 5-4.
- Luego en agregar nuevo **add New**.

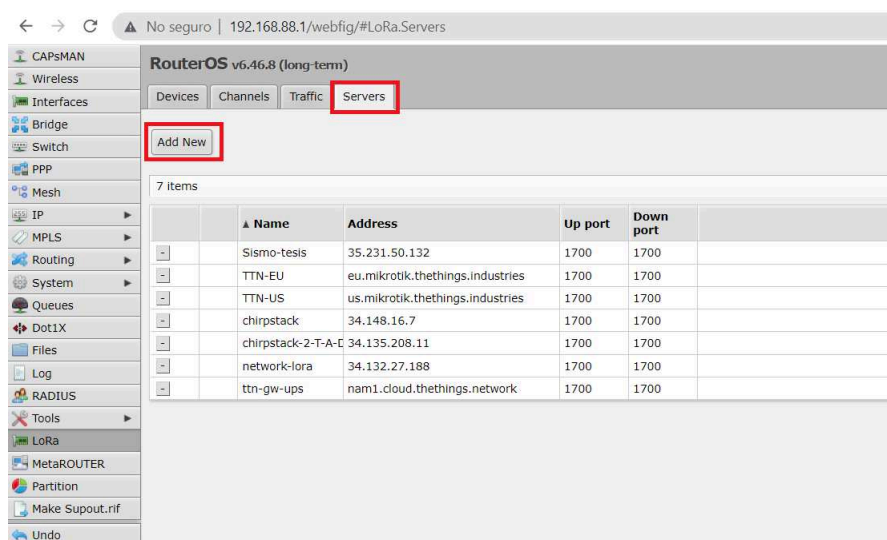


Figura 5-4.: Agregar nuevo servidor TTN.

### Paso 4. Añadir el Servidor TTN de The Things Stack V3.

- Los parámetros son:

**Name:** Nombre de nuestro GW.

**Address:** Dirección del servidor.

**Up port::** Puerto 1700 que usa el MikroTik.

**Down port:** Puerto 1700 que usa el MikroTik.

- La dirección **Address** se debe obtener de la configuración de GW en TTN.
- Una vez rellenado los campos le damos clic en **Apply** y **Ok** como se muestra en la fig. 5-5.

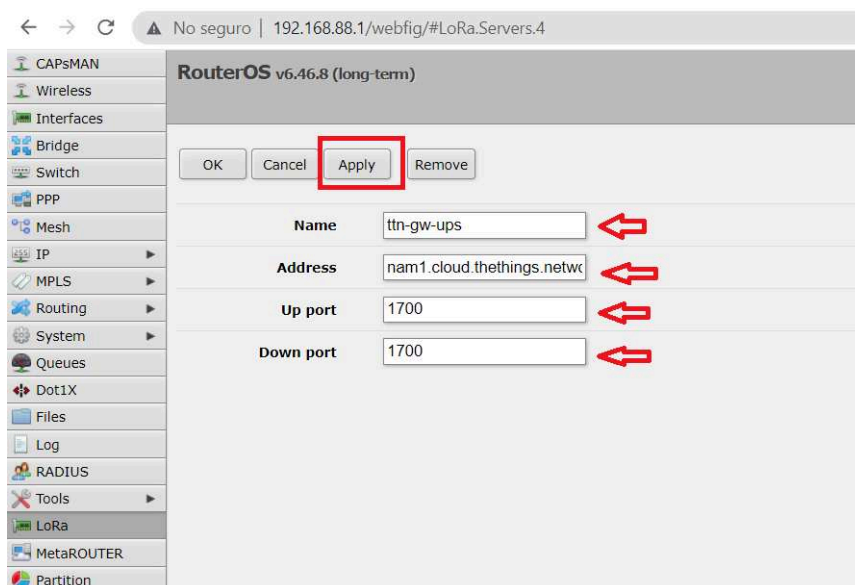


Figura 5-5.: Configurar Servidor TTN V3.

### Paso 5. Dar de alta el Servidor TTN.

- En la fig. 5-6 se observa las configuraciones realizadas en el equipo, para lo cual debemos revisar cuidadosamente que la información sea la correcta, para finalmente aplicar los cambios realizados.
- En **Network Server** debemos seleccionar el servidor que creamos en nuestro caso **ttn-gw-ups** se puede observar que se pueden dar de alta varios servidores a la vez en el mismo GW.
- **Channel plan** seleccionamos el canal de frecuencia con el que vamos a levantar nuestro servidor en nuestro caso vamos a trabajar con **US 915 Sub 1**, el Channel plan cuál debe ser el mismo también en la plataforma de TTN.

- Copiamos el **Gateway ID** el cual vamos a usar para registrar el GW en la plataforma de TTN.
- Le damos a **Apply** y debe levantarse el servidor en estado **enabled**.

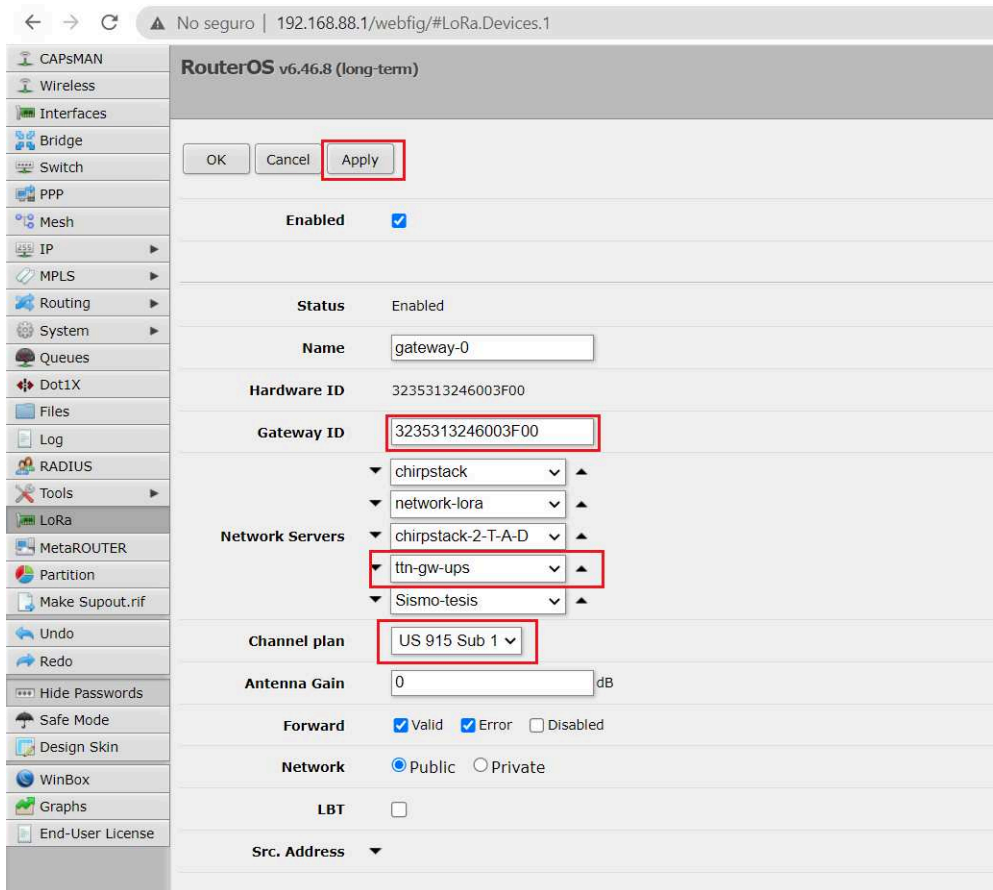


Figura 5-6.: Levantar Sevidor TTN.

### Paso 6. Revisar el estado del equipo.

- Finalmente, el estado de ejecución del equipo LoRa GW Mikrotik wAP LR8 debe estar en estado **Enabled** como se observa en la fig. 5-7, de lo contrario en la plataforma de TTN no se podrá dar de alta el dispositivo GW Mikrotik wAP LR8 kit.

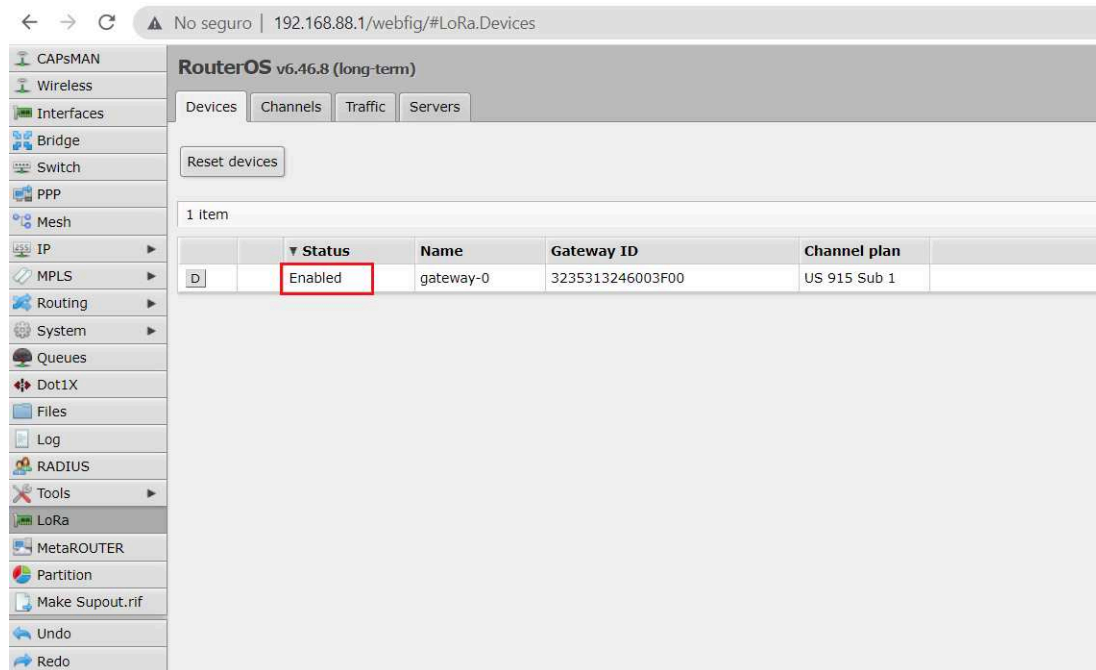


Figura 5-7.: Estado del equipo LoRa GW.

### 5.3.2. Configuración del GW en TTN.

#### Paso 1: Ingrese a la consola de TTN

- Desde un navegador ingrese a la siguiente página web: <https://www.thethingsnetwork.org/> y clic en **Start building** fig. 5-8.

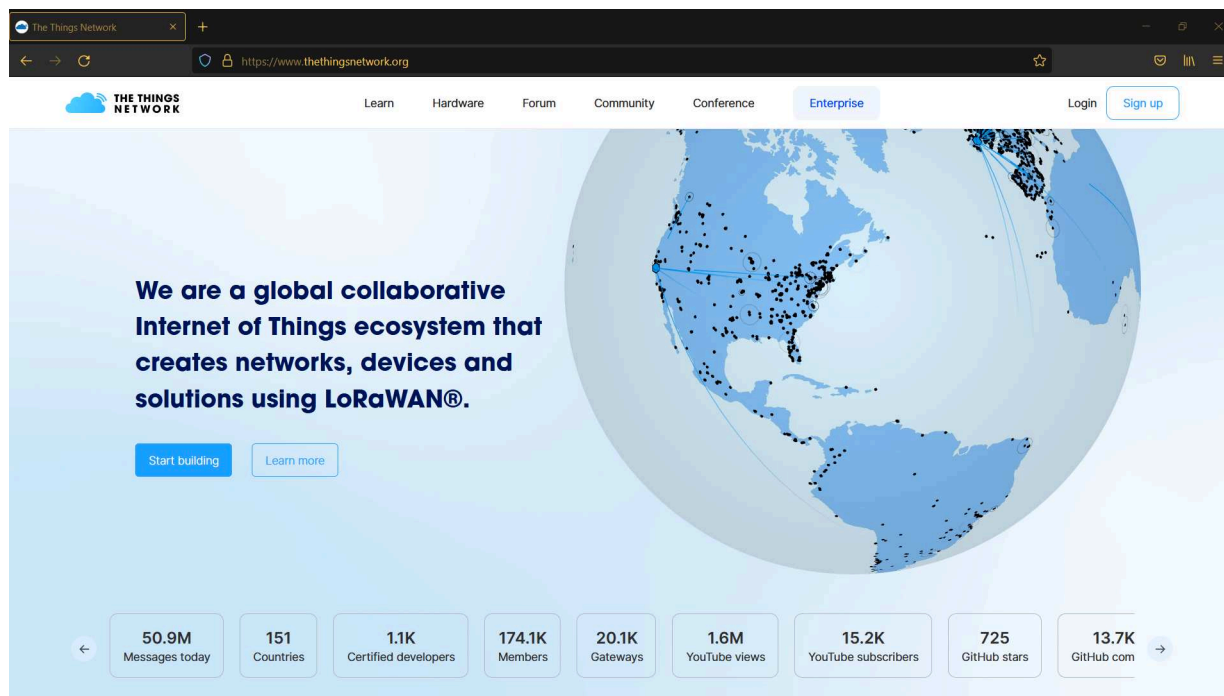


Figura 5-8.: Página de inicio de TTN

## Paso 2: Creación de una cuenta.

- Seleccionamos el modo de suscripción, en nuestro caso es el de estudiante fig. 5-9.

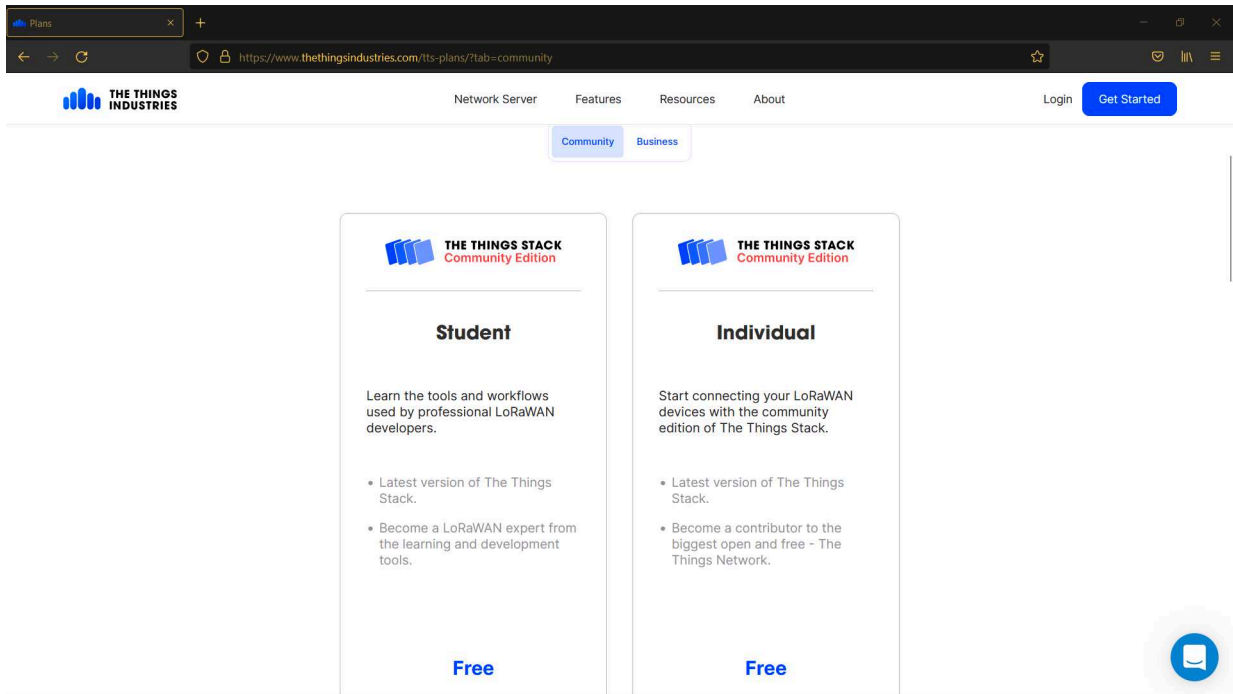


Figura 5-9.: Tipo de suscripción para TTN

### Paso 3: Seleccionamos The Things Network Cluster Picker.

- Vamos a trabajar con los servicios del Cluster de **North America 1** como se visualiza en fig. 5-10 le damos click.

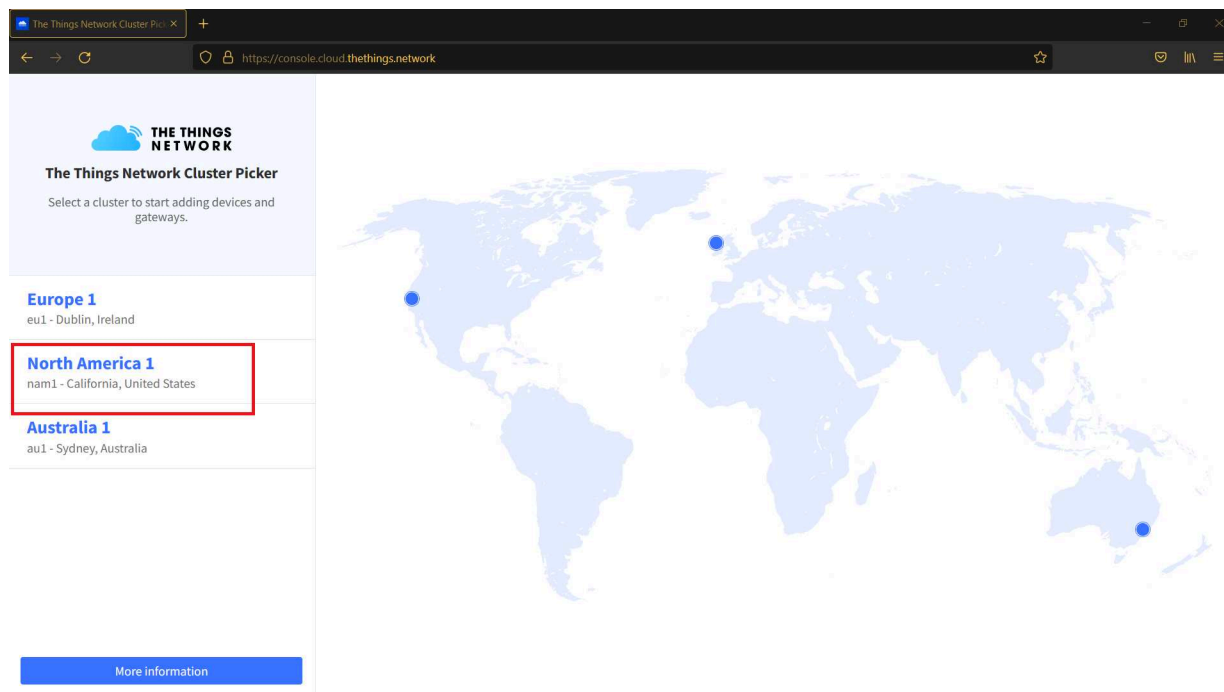


Figura 5-10.: Cluster y región de TTN.

#### Paso 4: Registrar cuenta nueva TTN.

- Debemos registrarnos primero para poder continuar.
- Seleccionamos la opción de **Register** para crear una nueva cuenta como se puede observar en la fig. 5-11.



**THE THINGS STACK**  
Community Edition

Login to The Things Stack with **The Things ID**

Username or email •

Password •

Submit

[Register](#) [Forgot password?](#)

The Things ID by The Things Industries

Figura 5-11.: Registrar una cuenta nueva.

### Paso 5: Creación de una nueva cuenta TTN.

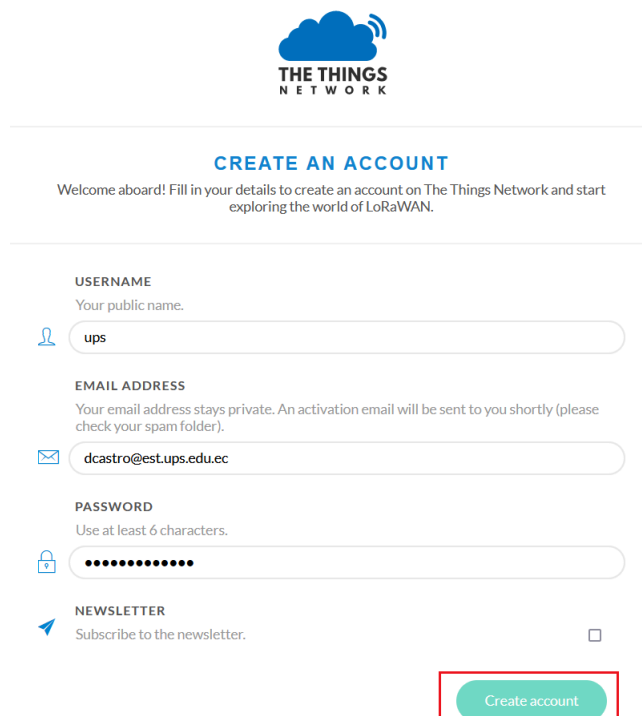
- En el siguiente formulario llenamos los campos con la información requerida tal como se ilustra en la fig. 5-12.

**Username:** Ingresamos nuestro nombre.

**Email address:** Ingresamos un correo válido, después debemos confirmar la creación en nuestro correo personal.

**Password:** Ingresamos una contraseña segura.

- Por último clic en **Create account**.



**THE THINGS NETWORK**

---

### CREATE AN ACCOUNT

Welcome aboard! Fill in your details to create an account on The Things Network and start exploring the world of LoRaWAN.

---

**USERNAME**  
Your public name.

**EMAIL ADDRESS**  
Your email address stays private. An activation email will be sent to you shortly (please check your spam folder).

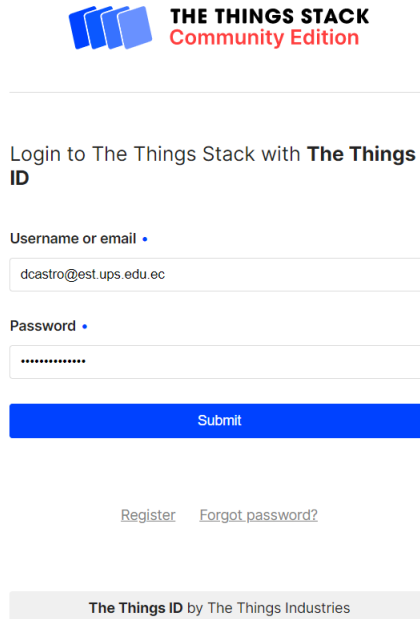
**PASSWORD**  
Use at least 6 characters.

**NEWSLETTER**  
Subscribe to the newsletter.

**Figura 5-12.:** Formulario para la crear una cuenta de TTN.

### Paso 6: Ingresar a la cuenta de TTN.

- Finalizada la creación de la cuenta, nos redirecciona a la página de Login como se muestra en la fig. 5-13, para lo cual debemos ingresar con las credenciales correspondientes.
- Ingresamos el **Email** con el cual se registró.
- Ingresamos el **Password** y finalmente clic en **Submit**.



The screenshot shows the login interface for The Things Stack Community Edition. At the top, there is a logo consisting of three blue squares of varying sizes to the left of the text "THE THINGS STACK" in bold black, with "Community Edition" in red below it. Below the logo is a horizontal line. The main heading reads "Login to The Things Stack with **The Things ID**". There are two input fields: "Username or email" containing "dcastro@est.ups.edu.ec" and "Password" containing a series of dots. A blue "Submit" button is positioned below the password field. At the bottom of the form area, there are two links: "Register" and "Forgot password?". A footer bar at the very bottom contains the text "The Things ID by The Things Industries".

Figura 5-13.: Login de ingreso a TTN.

### Paso 7: Creación de un nuevo GW.

- Seleccionamos la opción de **Go to gateways** que es la opción que nos permite registrar y configurar un nuevo dispositivo GW como se muestra en la fig. 5-14.

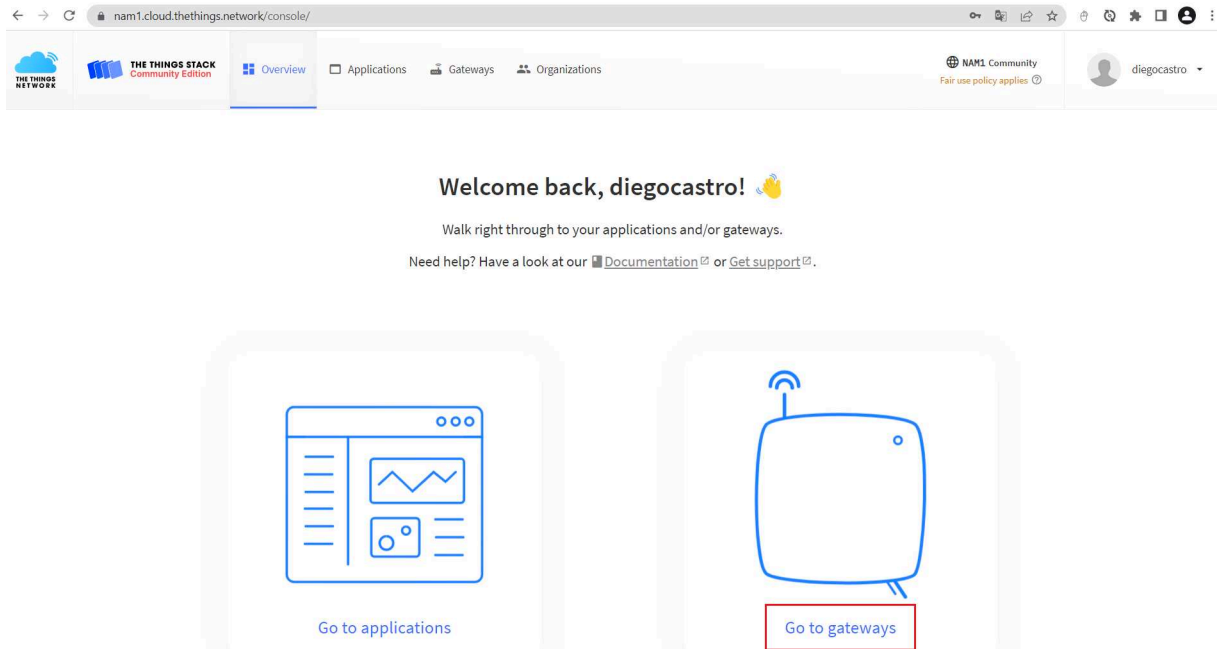


Figura 5-14.: Ingresar para crear GW.

### Paso 8: Creación de un nuevo GW.

- En la fig. 5-15 se observa la opción **Add gateway**, debemos considerar que podemos registrar varios GWs en la plataforma de TTN.
- Seleccionamos **Add gateway** para poder registrar un nuevo dispositivo GW.

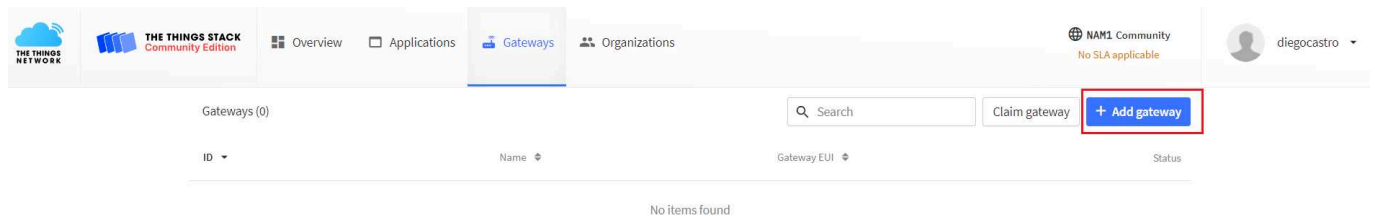


Figura 5-15.: Creación de un nuevo GW.

### Paso 9: Agregar el GW.

- Parámetros para registrar el GW fig. 5-16:
  - Gateway ID:** Este campo es el identificador de nuestro GW, se llena automáticamente.

**Gateway EUI:** Este código se obtiene de nuestro equipo LoRa GW Mikrotik wAP LR8, el mismo que fue configurado anteriormente, para lo cual debemos escribir el código EUI que le corresponde al equipo.

**Gateway name:** Registramos con cualquier nombre, este nombre debe ser el mismo que se creó en el equipo MikroTik LoRa GW wAP LR8.

**Gateway description:** Registramos con una descripción, es opcional.

**Gateway Server Address:** Se llena automáticamente, ya que depende del servidor clúster que hayamos elegido, para este caso en los pasos anteriores seleccionamos Norte América 1 y click en **Create Gateway** como se muestra en la fig. 5-17.

The screenshot shows the 'Add gateway' form in the TTN console. The form is titled 'Add gateway' and is under the 'General settings' section. It contains the following fields:

- Gateway ID**: eui-3235313246003f00
- Gateway EUI**: 32 35 31 32 46 00 3f 00
- Gateway name**: ttn-gw-ups
- Gateway description**: Integración de Mikrotik wAP LR8 kit en TTN.
- Gateway Server address**: nam1.cloud.thethings.network

Red boxes highlight the Gateway EUI and Gateway Server address fields. The Gateway Server address field has a small note below it: 'The address of the Gateway Server to connect to'.

Figura 5-16.: Formulario para el registro del GW.

← → ↻ nam1.cloud.thethings.network/console/gateways/add

Make location public  
When set to public, the gateway location may be visible to other users of the network

**Attributes** ⓘ  
+ Add attributes  
Attributes can be used to set arbitrary information about the entity, to be used by scripts, or simply for your own organization

**LoRaWAN options**

**Frequency plan** ⓘ \*

Select... | v

**Schedule downlink late** ⓘ

Enabled  
Enable server-side buffer of downlink messages

**Enforce duty cycle** ⓘ

Enabled  
Recommended for all gateways in order to respect spectrum regulations

**Schedule any time delay** ⓘ \*

530 milliseconds | v  
Configure gateway delay (minimum: 130ms, default: 530ms)

**Gateway updates**

**Automatic updates**

Enabled  
Gateway can be updated automatically

**Channel**

Stable  
Channel for gateway automatic updates

**Create gateway**

Figura 5-17.: Formulario para la registro del GW.

## Paso 10: Conexión de GW Mikrotik wAP LR8 kit en TTN.

- Si toda la configuración fue correcta se debe observar que el GW está en estado **Connected** tal como se ilustra en la fig. 5-18.
- Una vez realizado el registro del GW en TTN ya podemos implementar diferentes aplicaciones y dispositivos que sean compatibles con la plataforma de TTN.

← → ↻ nam1.cloud.thethings.network/console/gateways

THE THINGS NETWORK THE THINGS STACK Community Edition Overview Applications Gateways Organizations NAM1 Community No support plan ⓘ

Gateways (1) Search Claim gateway + Add gateway

ID	Name	Gateway EUI	Status	Created at
eui-3235513246003f00	ttn-gw-ups	32 35 31 32 46 00 3F 00	Connected	May 16, 2022

Figura 5-18.: Verificación del estado del GW.

### Paso 11: Data generada del GW.

- Finalmente, se debe de visualizar la conexión correspondiente del equipo Lora GW al servidor TTN, para poder ver la data generada nos debemos dirigir a **Live data**, tal como se visualiza en la fig. 5-19, el intercambio de los paquetes de conexión son correctos.

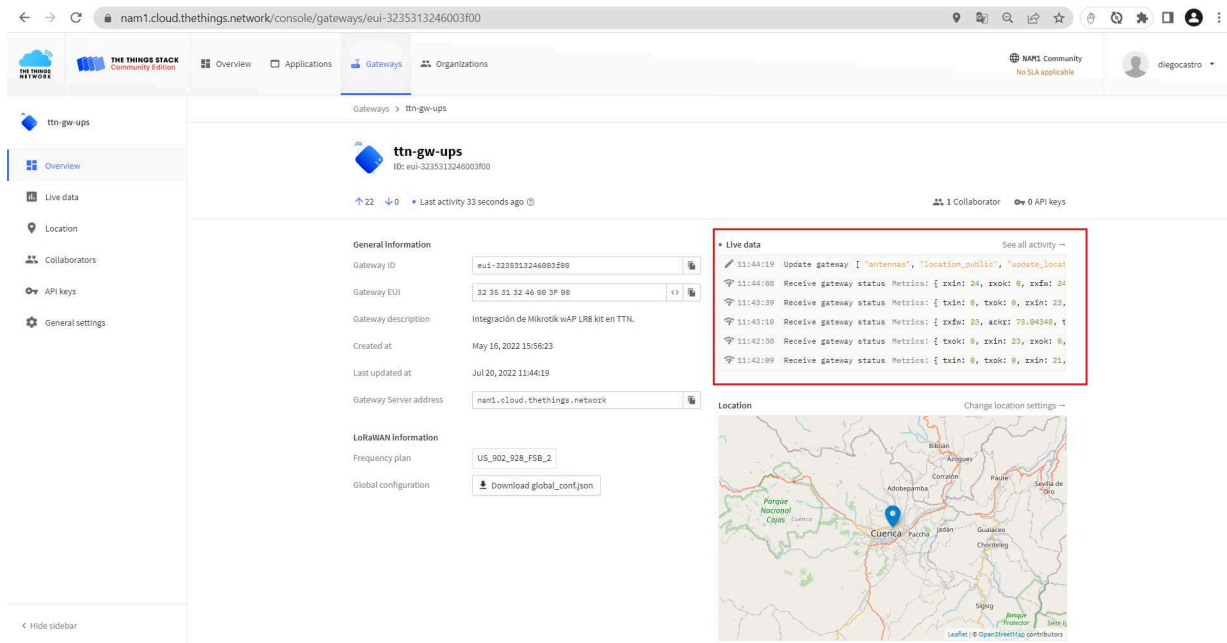


Figura 5-19.: Datos de conexión entre el GW y TTN.

## 5.4. Creación y configuración de la aplicación y el dispositivo final en TTN

### 5.4.1. Configuración de la aplicación en TTN.

#### Paso 1. Seleccionamos la opción de Aplicaciones.

- Para empezar la configuración de los dispositivos en TTN, hacemos clic en **Go to applications** como se muestra en la fig. 5-20.
- Primero vamos a crear una nueva aplicación.
- Segundo vamos a integrar un dispositivo final LoRa para recolectar información y así poder ver el comportamiento en la plataforma de TTN.

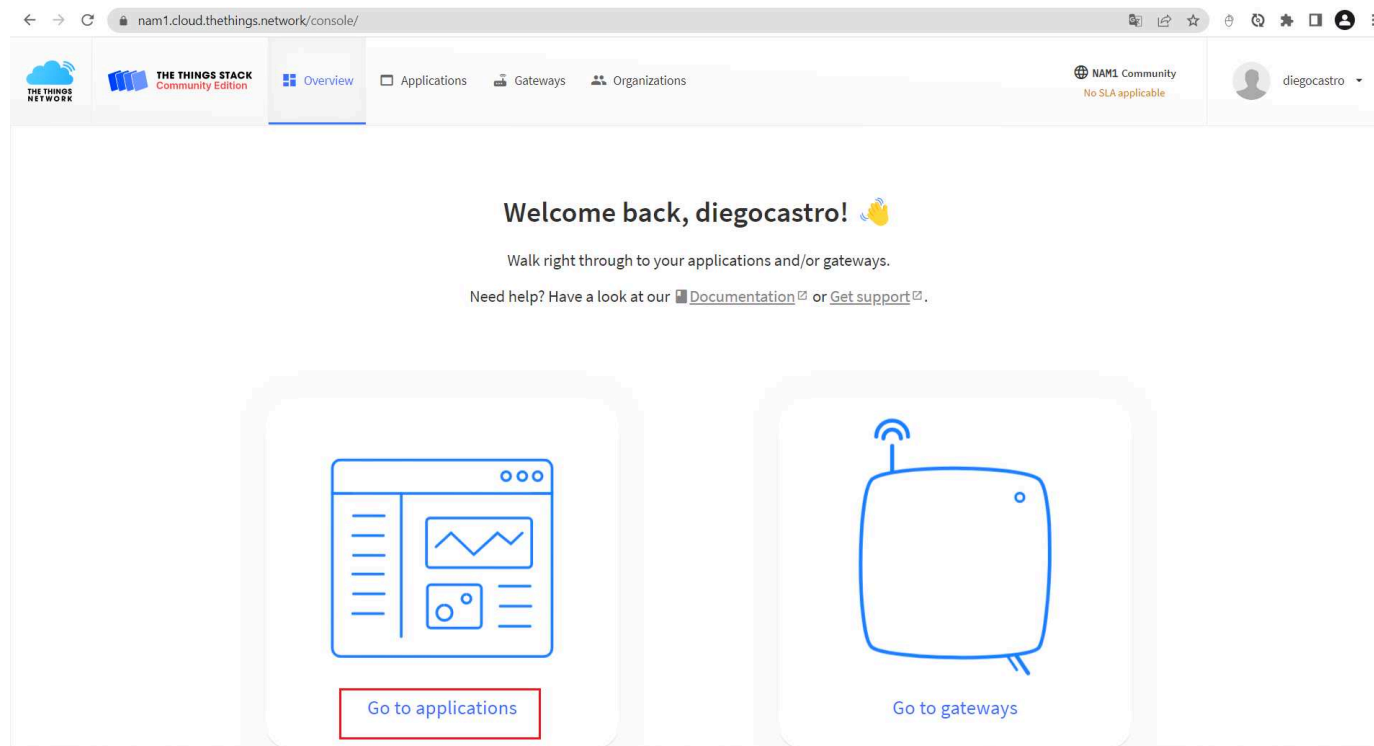


Figura 5-20.: Página de inicio para crear una aplicación.

#### Paso 2: Agregamos una aplicación.

- Procedemos a crear una aplicación le damos click en **Add application** fig. 5-21.
- Podemos crear varias aplicaciones y usarlas según sea el caso.



## 5.4 Creación y configuración de la aplicación y el dispositivo final en TTN

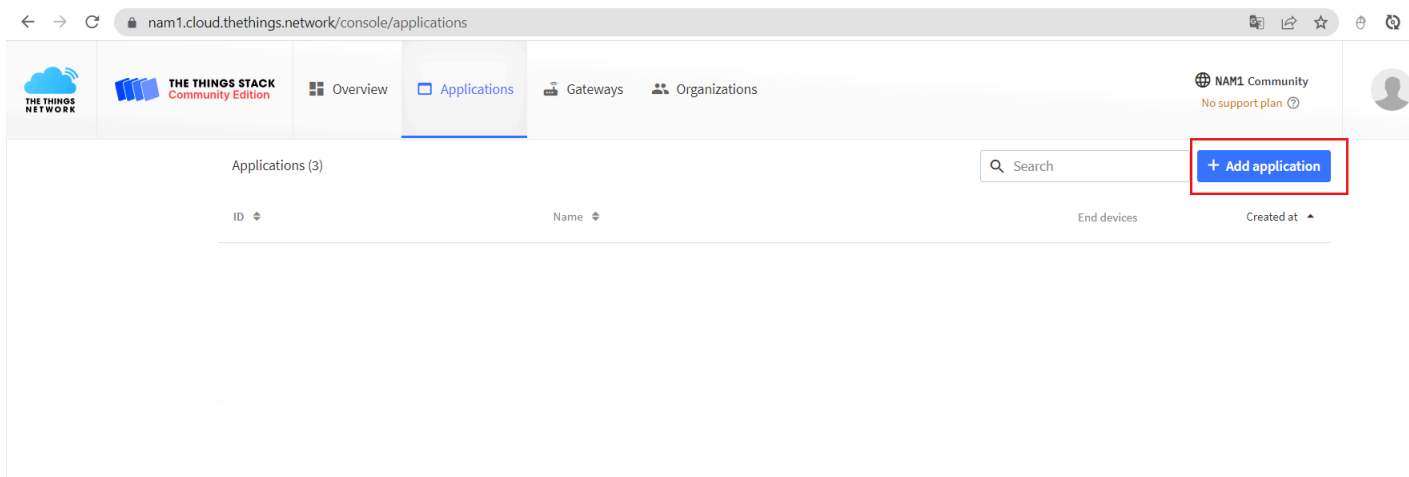
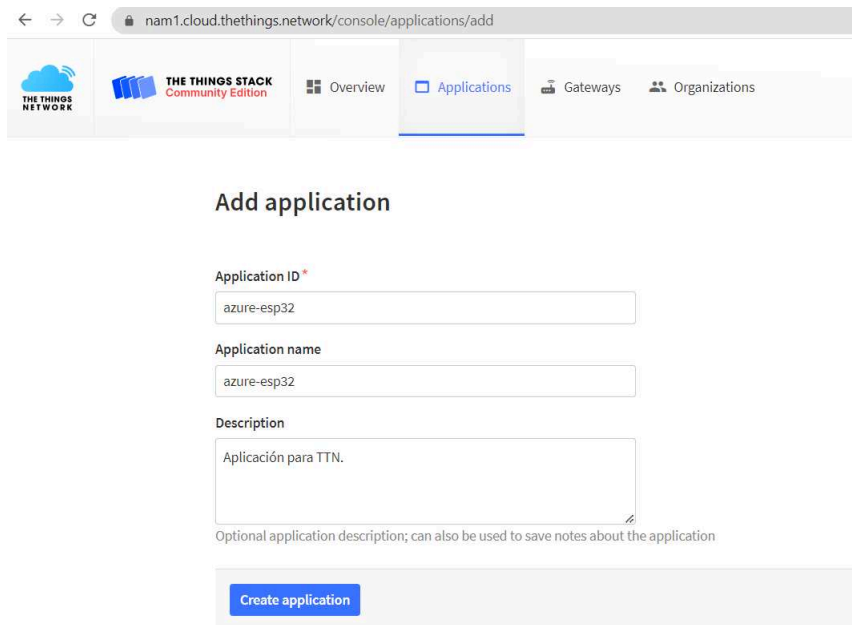


Figura 5-21.: Agregar una nueva aplicación.

### Paso 3: Creación de la aplicación en TTN.

- En el siguiente formulario llenamos los campos para la creación de una aplicación .
- Los parámetros a ingresar son fig. 5-22:
  - Application ID:** Se le agrega un nombre o código de registro único para esta aplicación.
  - Application name:** Le damos un nombre con el cual vamos a identificar a nuestra aplicación.
  - Description:** Podemos darle una descripción sobre el uso que se le va a dar a la aplicación creada.



The screenshot shows a web browser window with the URL `nam1.cloud.thethings.network/console/applications/add`. The page header includes the The Things Network logo, 'THE THINGS STACK Community Edition', and navigation tabs for 'Overview', 'Applications' (which is active), 'Gateways', and 'Organizations'. The main content area is titled 'Add application' and contains three input fields: 'Application ID\*' with the value 'azure-esp32', 'Application name' with the value 'azure-esp32', and 'Description' with the value 'Aplicación para TTN.'. Below the description field is a note: 'Optional application description; can also be used to save notes about the application'. At the bottom of the form is a blue button labeled 'Create application'.

Figura 5-22.: Formulario para la creación de la aplicación.

#### Paso 4: Registro de la aplicación creado correctamente.

- La configuración y creación de la aplicación ha sido generada correctamente.
- Para agregar un dispositivo final, damos clic en **Add end device** fig. 5-23.

## 5.4 Creación y configuración de la aplicación y el dispositivo final en TTN

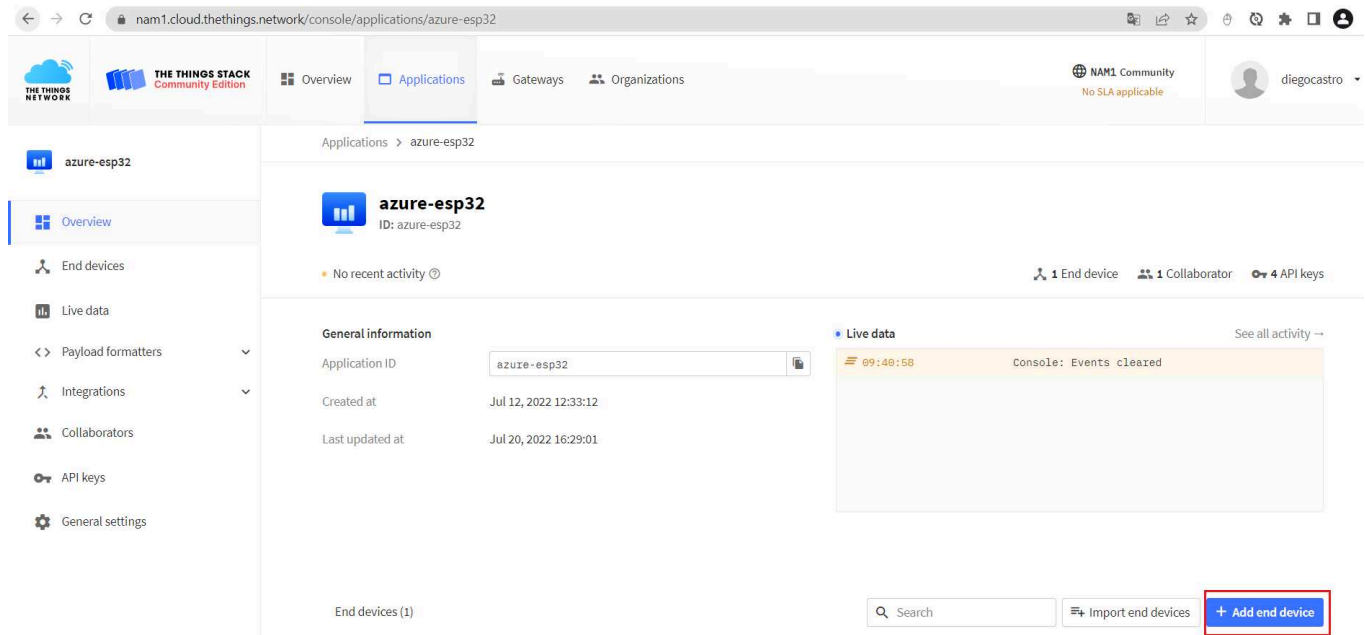


Figura 5-23.: Aplicación creada correctamente.

### 5.4.2. Creación y Configuración del dispositivo Final HelTec ESP32

#### Paso 5: Registro del dispositivo LoRaWAN

- Para el registro del dispositivo existen dos métodos; desde un repositorio (**From the LoRaWAN Device Repository**) fig. 5-24 o de manera manual (**Manually**), en este caso utilizaremos el repositorio, ya que nuestro equipo a usar (**HelTec**) consta en la lista del repositorio de TTN, en caso de no estar en la lista deberíamos optar el registro de manera manual.
- Dentro del Repositorio de dispositivos LoRaWAN de TTN contiene varios perfiles de dispositivos, la información de LoRaWAN, códecs y entre otra información para poder registrar dispositivos LoRaWAN.
- Al usar el repositorio para registrar un dispositivo en TTN el repositorio se encarga de usar automáticamente la versión correcta de LoRaWAN y la versión de parámetros como la región para nuestro dispositivo.
- Consideramos los siguientes parámetros:
  - Brand:** Seleccionamos la marca de nuestro equipo.
  - Model:** Seleccionamos el modelo del equipo que disponemos.

**Hardware versión:** Este parámetro pone a disposición el fabricante por lo que podemos encontrar disponible en el paquete del equipo o en la documentación del fabricante en internet.

**Firmware versión:** Es similar al paso del hardware.

**Profile Region:** Es igual que los pasos anteriores, disponible en el paquete del equipo o en la documentación.

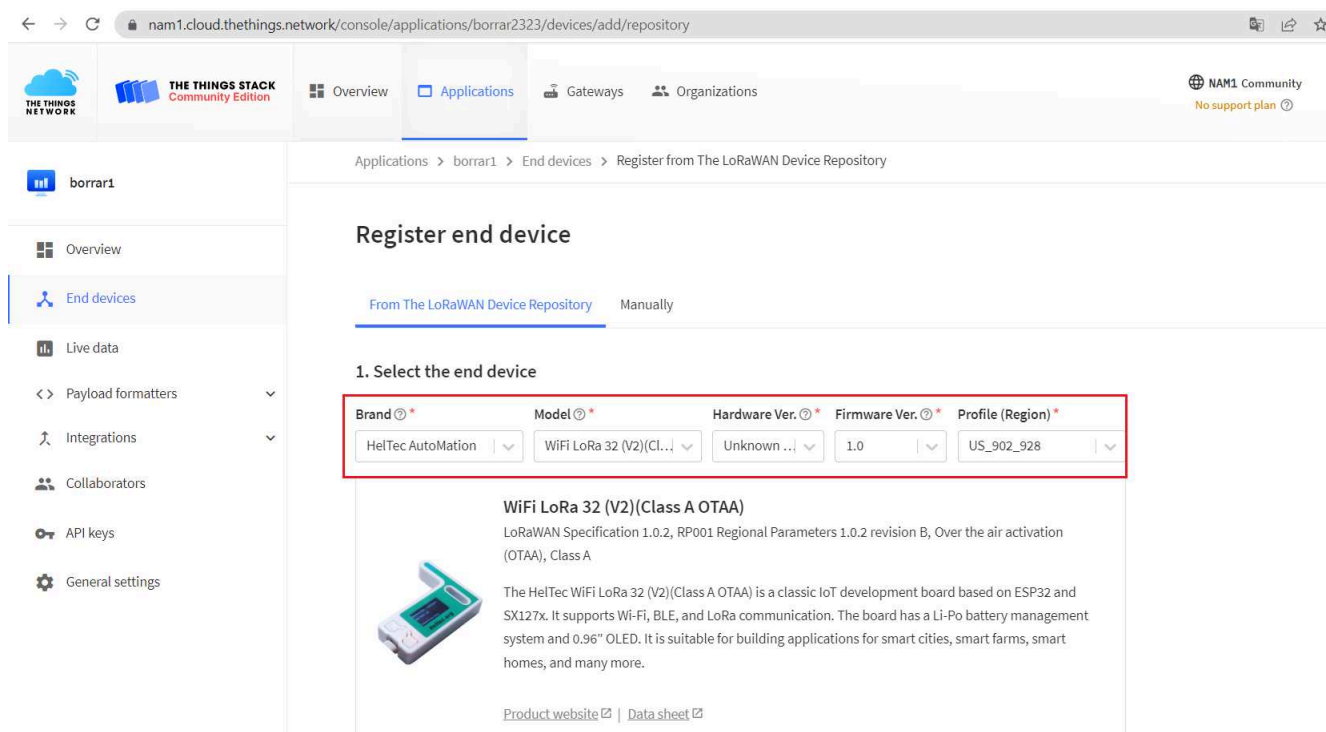


Figura 5-24.: Registro del dispositivo usando el Repositorio de TTN.

## Paso 6: Parámetros del registro del dispositivo

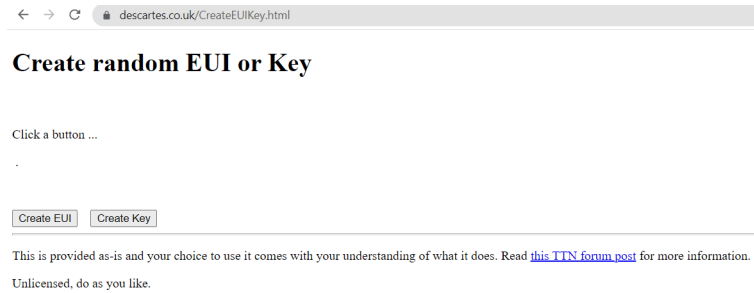
- Se deben considerar los siguientes parámetros fig. 5-26:

**Frequency plan:** Debemos seleccionar el plan de frecuencia para la región con la que vamos a trabajar, recuerde que esta región debe ser la misma en el dispositivo GW.

**AppEUI:** Si el dispositivo contiene un AppEUI se debe ingresar caso contrario se le puede generar automáticamente con ceros.

**DevEUI:** Esto se lo debe de proporcionar el fabricante del dispositivo, en el caso de los dispositivos programables como nuestro caso lo debemos generar en una página web para luego ser implementado en el código del dispositivo.

**¿Cómo crear el DevEUI?** Para crear la clave podemos ayudarnos de una página web, fig. 5-25, <https://descartes.co.uk/CreateEUIKey.html> y clic en **Create EUI**.



**Figura 5-25.:** Crear el DevEUI

**AppKey:** Si el fabricante del dispositivo lo proporciona debe agregarlo, por el contrario, lo puede generar automáticamente en el paso anterior para luego programarlo en el código del dispositivo.

**End device ID:** Es el ID con el cual se le va a reconocer a su dispositivo el cual es único.

- Por último le damos clic en **Register end device**.

The screenshot shows the '2. Enter registration data' form in the TTN console. The form includes the following fields and options:

- Frequency plan**: United States 902-928 MHz, FSB 2 (used by TTN)
- AppEUI**: 00 00 00 00 00 00 00 00 (with a 'Fill with zeros' button)
- DevEUI**: 5A D2 A0 00 D0 70 7B 9C (with a 'Generate' button and '0/50 used' indicator)
- AppKey**: F9 33 98 B6 59 40 FF 18 7D FA 67 BF 5E F4 50 77 (with a 'Generate' button)
- End device ID**: eui-5ad2a000d0707b9c (with a note: 'This value is automatically prefilled using the DevEUI')
- After registration**:
  - View registered end device
  - Register another end device of this type
- Register end device** button

Figura 5-26.: Configuración y creación del dispositivo.

### Paso 7: Registro del dispositivo creado correctamente.

- La configuración y creación del dispositivo ha sido realizado correctamente.
- En la fig. 5-27 podemos visualizar que todavía no se a establecido la conexión del dispositivo y no hay registró de actividad de datos, ya que primero debemos de programar el dispositivo para poder engancharlo a la plataforma de TTN.

## 5.4 Creación y configuración de la aplicación y el dispositivo final en TTN

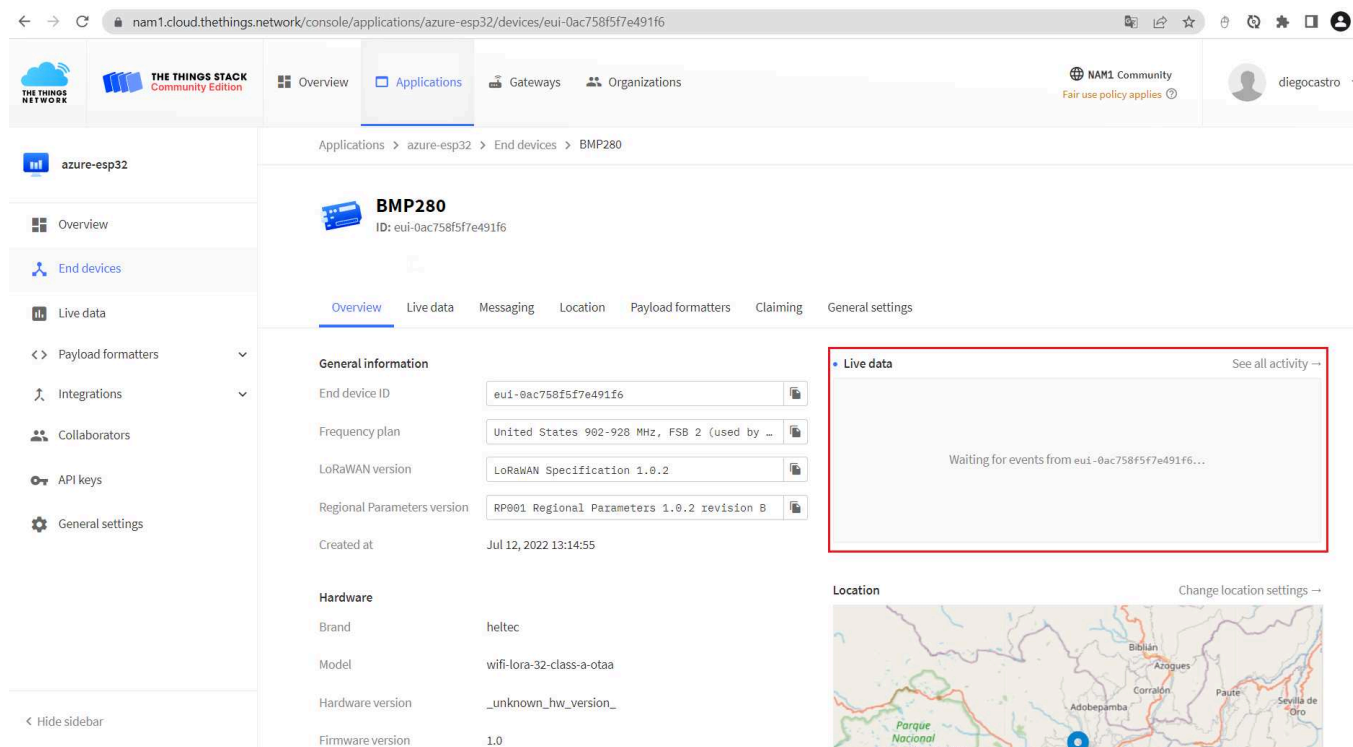


Figura 5-27.: Nodo final creada correctamente.

### 5.4.3. Programación del dispositivo HelTec ESP32 Lora 32 V2 en el IDE Arduino

En esta sección indicaremos los pasos específicos para la correcta configuración de nuestro entorno de desarrollo para el dispositivo HelTec ESP32, a continuación se detalla cada paso.

1. Abrir el IDE de Arduino, hacer clic en la pestaña **Archivo**-> luego en **Preferencias** tal como se observa en la fig. 5-28.

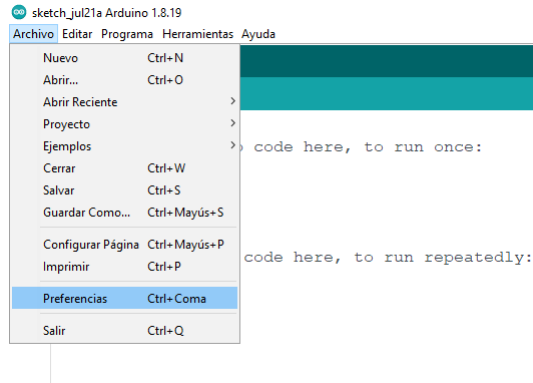


Figura 5-28.: Inicio de la configuración del IDE.

2. En la ventana de **Preferencias** se debe pegar la URL de la biblioteca de gestor de tarjetas para poder trabajar con dispositivos HelTec ESP32 en el IDE de Arduino, la siguiente dirección es: [https://github.com/HelTec-Aaron-Lee/WiFi\\_Kit\\_series/releases/download/0.0.6/package\\_HelTec\\_esp32\\_index.json](https://github.com/HelTec-Aaron-Lee/WiFi_Kit_series/releases/download/0.0.6/package_HelTec_esp32_index.json) como se ilustra en la fig. 5-29 y por último en **OK**.

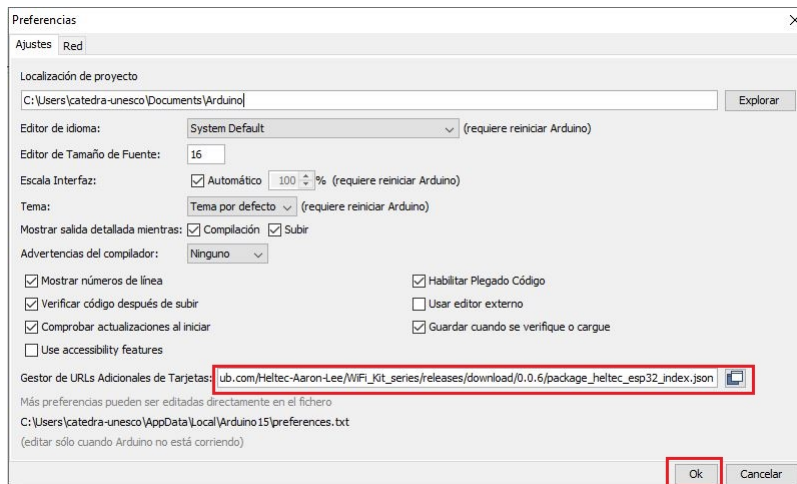


Figura 5-29.: Configuración biblioteca de gestor de tarjeta ESP32.

3. Dentro del menú **Herramientas**, seleccionamos la opción **Gestor de tarjetas** fig. 5-30 para proceder a instalar el gestor de tarjeta compatible con el dispositivo HelTec ESP32.



## 5.4 Creación y configuración de la aplicación y el dispositivo final en TTN

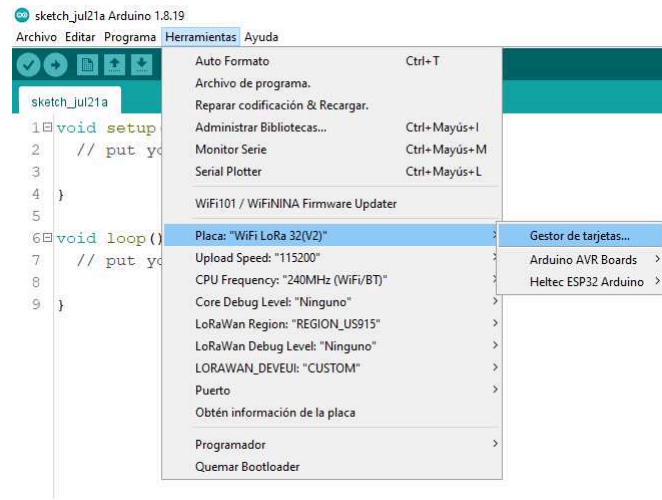


Figura 5-30.: Gestor de Tarjetas en el IDE de Arduino.

4. En la ventana del **Gestor de tarjetas** buscamos **HelTec ESP32**, procedemos a instalar el paquete gestor **HelTec ESP32 Series Dev-boards** como se observa en la fig. 5-31.

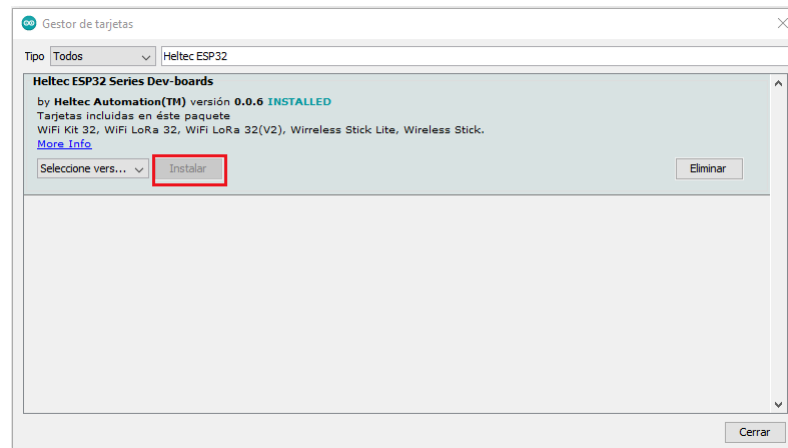


Figura 5-31.: Instalación correcta de la tarjeta.

5. Por último, tenemos que reiniciar el IDE de Arduino, nos dirigimos al menú **Herramientas** y se debe observar que la placa de HelTec ESP32 ya está instalada y configurada correctamente tal como se muestra en la fig. 5-32.

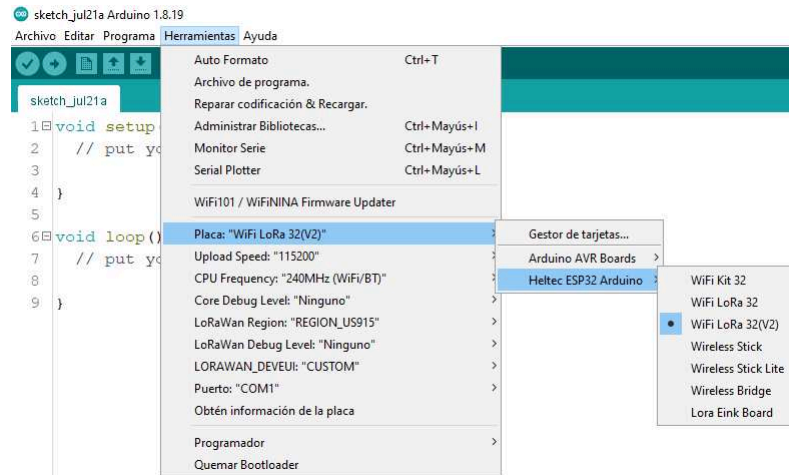


Figura 5-32.: Configuración correcta del IDE Arduino.

#### 5.4.4. Desarrollo del entorno con Arduino en la placa HelTec ESP32

1. Primero se debe de implementar algunas de las librerías que se muestran en la fig. 5-33, para poder comenzar a programar el dispositivo HelTec ESP32.

```

26 #include <Wire.h>
27 #include <ESP32_LoRaWAN.h>
28 #include <Arduino.h>
29 #include <CayenneLPP.h>
30 #include <Adafruit_Sensor.h>
31 #include <Adafruit_BMP280.h>

```

Figura 5-33.: Implementación de librerías.

- Luego nos dirigimos al menú **Programa** para **incluir librería** y en **Administrar Bibliotecas** como se visualiza en la fig. 5-34.

## 5.4 Creación y configuración de la aplicación y el dispositivo final en TTN

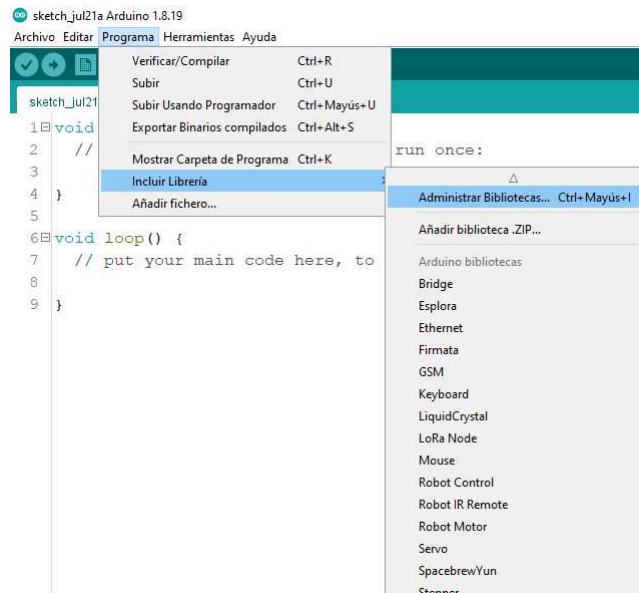


Figura 5-34.: Biblioteca de librerías para el IDE de Arduino.

- Incluimos la librería **CayenneLPP** fig. 5-35 para IoT que nos permite diseñar, crear prototipos y visualizar en tiempo real datos de dispositivos IoT. Esta librería es muy útil para visualizar datos históricos y en tiempo real, que son enviados a TTN.

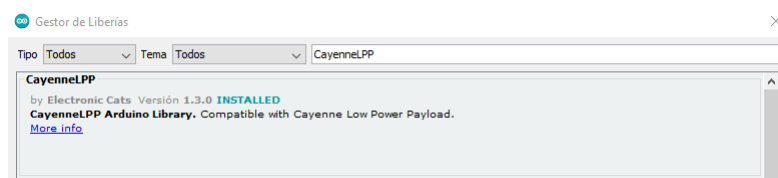


Figura 5-35.: Descarga de librería CayenneLPP.

- Descargamos la biblioteca de **Adafruit BMP280** ver fig. 5-36, esta librería la vamos a usar para poder trabajar con un sensor de Presión **BMP280** para poder recolectar datos.

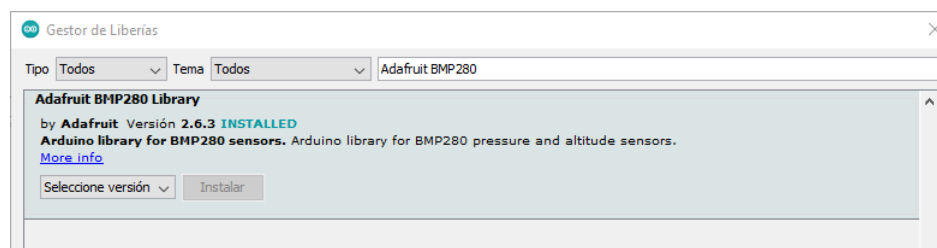


Figura 5-36.: Descarga de librería Adafruit BMP280.

- Descargamos la librería **ESP32\_LoRaWAN** fig. 5-37, desde el siguiente enlace [https://github.com/HelTecAutomation/ESP32\\_LoRaWAN](https://github.com/HelTecAutomation/ESP32_LoRaWAN) el cual contiene ejemplos de prueba para poder realizar la conexión del dispositivo HelTec ESP32 en la plataforma de TTN.

Esta biblioteca ayuda a que el protocolo de LoRaWAN 1.0.2 se ejecute en dispositivos ESP32. Es compatible con los productos fabricados por HelTec Automation(TM) y se necesita una puerta de enlace LoRa para poder realizar la integración con TTN.

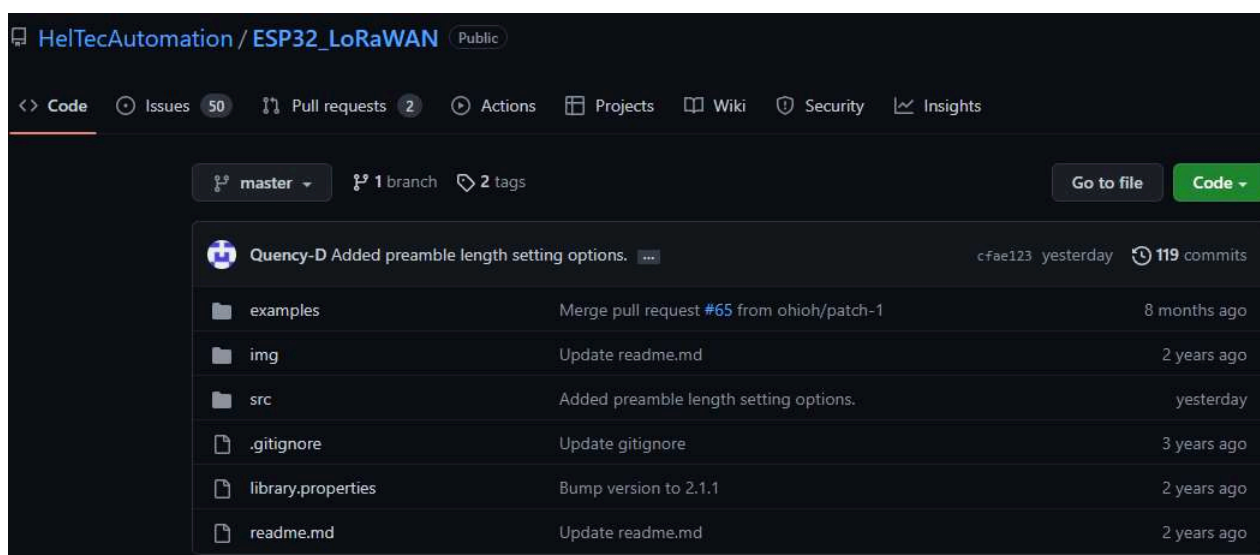


Figura 5-37.: Descarga de librería ESP32\_LoRaWAN.

- También debemos descargar la biblioteca **Adafruit\_Sensor** desde el siguiente enlace [https://github.com/adafruit/Adafruit\\_Sensor](https://github.com/adafruit/Adafruit_Sensor) como se observa en fig. 5-38, esta biblioteca nos permite trabajar con múltiples tipos de sensores como de presión, calor, humedad, temperatura, entre otros.

## 5.4 Creación y configuración de la aplicación y el dispositivo final en T165

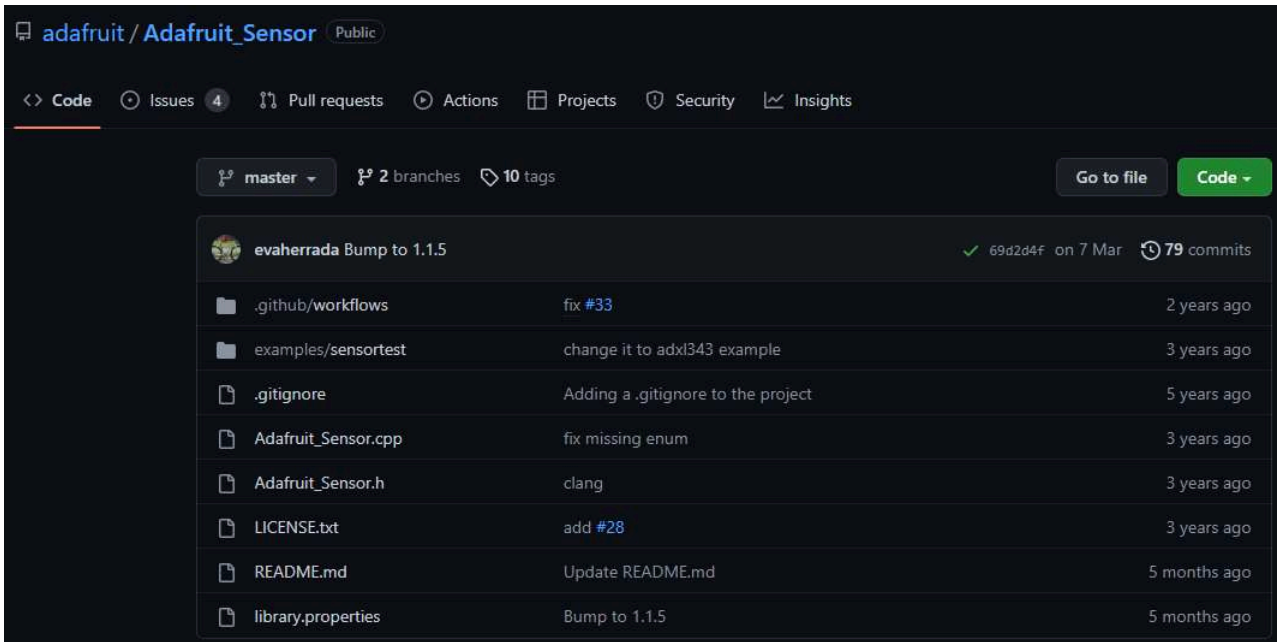


Figura 5-38.: Descarga de librería Adafruit\_Sensor.

- Por ultimo descargamos la biblioteca de **I2CScanner** fig. 5-39 esta librería nos permite conocer la dirección de Bus de datos en la que trabaja el sensor, las direcciones son únicas para cada dispositivo.

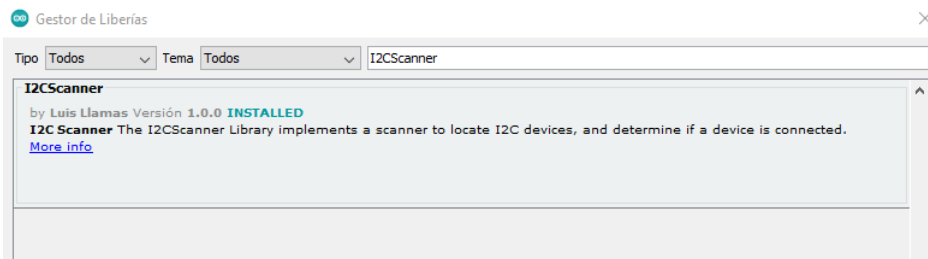


Figura 5-39.: Descarga de librería I2CScanner.

2. Finalizada la implementación y configuraciones de todas las bibliotecas, desde un ejemplo propio de Arduino empezamos a programar nuestro código para que posteriormente sea enviado por cable USB a nuestro HelTec ESP32 v2. A continuación, especificamos los pasos importantes para la correcta programación. La licencia se debe obtener desde un código que es ejecutado en Arduino.

- Cuando se ejecuta el código debemos abrir un **Monitor Serie** desde el menú **Herramientas** del IDE, ahí observamos el Chip ID de nuestro dispositivo, fig. 5-40.

```

chipID Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda
chipID
1 uint64_t chipId = 0;
2 void setup() {
3   Serial.begin(115200);
4 }
5 void loop() {
6   chipId |= (ESP.getEfuseMac() >> (40 - i)) & 0xff << i;
7   chipId=ESP.getEfuseMac();//The chip ID is essentially its MAC address(length: 6 bytes).
8   Serial.printf("ESP32ChipID=%04X", (uint16_t) (chipId>>32));//print High 2 bytes
9   Serial.printf("%08X\n", (uint32_t) chipId);//print Low 4bytes.
10
11 // Serial.printf("ESP32 Chip model = %s Rev %d\n", ESP.getChipModel(), ESP.getChipRevision());
12 // Serial.printf("This chip has %d cores\n", ESP.getChipCores());
13 // Serial.print("Chip ID: "); Serial.println(chipId);
14   delay(3000);
15 }
COM5
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
ESP32ChipID=60195BBD9E7C
Autoscroll Mestrar marca temporal

```

Figura 5-40.: Código para obtener el ChipID

- Desde un navegador abrimos la siguiente página web <https://resource.heltec.cn/search/> fig. 5-41 y pegamos el código obtenido anteriormente para que nos asigne aún licencia válida.

← → ↻ resource.heltec.cn/search/

Please enter product id:

Figura 5-41.: Pagina web para ingresar el ChipID.

- Una vez realizado el paso anterior le damos a **Confirm** y nos muestra las características del dispositivo Heltec ESP32 como son el id, licencia, entre otros fig. 5-42.

Type	id	license	Production Place	Manufacture Data
ESP32	60195bbd9e7c	0xB74F0050,0xCF79ED21,0xD9B6064E,0x2265BE61	Chengdu	2021/06/09 13:12:18

[Back](#)

[Relevant Resource](#)

Figura 5-42.: Licencia valida para nuestro dispositivo.

- La licencia obtenida pegamos en nuestro código programado en Arduino, en la línea que corresponde a la licencia como se observa en la fig. 5-43.

## 5.4 Creación y configuración de la aplicación y el dispositivo final en TTN

```
/*license for Heltec ESP32 LoRaWan, quarry your ChipID relevant license: http://resource.heltec.cn/search */  
uint32_t license[4] = { 0xB74F0050, 0xCF79ED21, 0xD9B6064E, 0x2265BE61 };
```

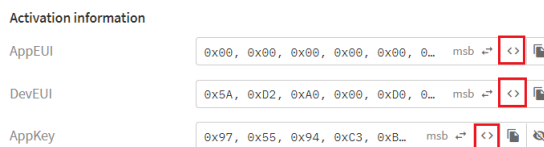
**Figura 5-43.:** Licencia valida obtenida agregada al código.

3. Para esta sección del código, estos parámetros se obtiene desde el dispositivo creado en TTN, para nuestro proyecto el dispositivo tiene por nombre **BMP280**. Para obtener el código como se muestra en la fig. **5-44** debemos hacer click en los cuadros indicados en la fig. **5-45**.

Ahora solo debemos ir copiando y pegando en los campos correspondientes del código.

```
/* OTAA parametros*/  
uint8_t DevEui[] = { 0x5A, 0xD2, 0xA0, 0x00, 0xD0, 0x70, 0x7B, 0x9C };  
uint8_t AppEui[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };  
uint8_t AppKey[] = { 0x97, 0x55, 0x94, 0xC3, 0xBF, 0xAD, 0x9E, 0x20, 0xDB, 0x15, 0x86, 0x04, 0xFE, 0x71, 0xAE, 0x3B };
```

**Figura 5-44.:** Parámetros reemplazados en el código.



**Figura 5-45.:** Parámetros de nuestro dispositivo creado en TTN.

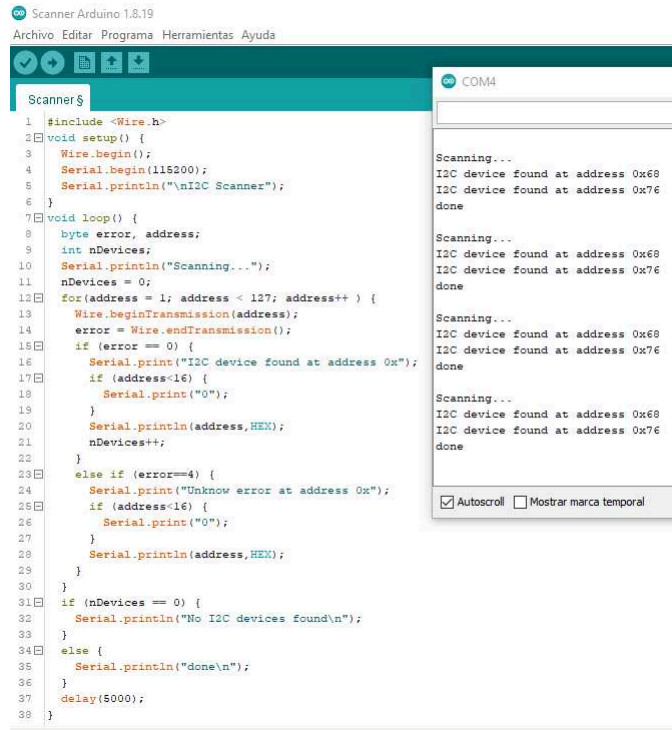
4. Considerar que la velocidad de carga que esta en la parte del código **Serial.begin(115200)** sea la misma que en el IDE del arduino, para lo cual debemos verificar en el menu **Herramientas** en la sección de **Upload Speed**. Caso contrario de no estar la misma velocidad se generaría un error al momento de cargar el código al dispositivo como se muestra en la fig. **5-46**.

```
void setup() {  
  
  Serial.begin(115200);  
  unsigned status;  
  status = bme.begin(0x76);  
  if (!status) {  
    Serial.println('Could not find a valid BME280 sensor, check wiring, address, sensor ID!');  
  }  
}
```

**Figura 5-46.:** Código de velocidad de transmisión y dirección de conexión del Bus de datos.

Para la sección de `status = bme.begin(0x76)`, ejecutamos el siguiente código, y abrimos un Monitor Serie desde el menú **Herramientas** del IDE de Arduino, aquí

podemos verificar que nos da como respuesta dos direcciones para la conexión el 0x68 y el 0x76, para nuestro código optamos por la segunda opción fig. 5-47.



```

Scanner Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda

Scanner$
1 #include <Wire.h>
2 void setup() {
3   Wire.begin();
4   Serial.begin(115200);
5   Serial.println("\nI2C Scanner");
6 }
7 void loop() {
8   byte error, address;
9   int nDevices;
10  Serial.println("Scanning...");
11  nDevices = 0;
12  for(address = 1; address < 127; address++) {
13    Wire.beginTransmission(address);
14    error = Wire.endTransmission();
15    if (error == 0) {
16      Serial.print("I2C device found at address 0x");
17      if (address < 16) {
18        Serial.print("0");
19      }
20      Serial.println(address, HEX);
21      nDevices++;
22    }
23    else if (error==4) {
24      Serial.print("Unknow error at address 0x");
25      if (address < 16) {
26        Serial.print("0");
27      }
28      Serial.println(address, HEX);
29    }
30  }
31  if (nDevices == 0) {
32    Serial.println("No I2C devices found\n");
33  }
34  else {
35    Serial.println("done\n");
36  }
37  delay(5000);
38 }
  
```

Serial Monitor (COM4):

```

Scanning...
I2C device found at address 0x68
I2C device found at address 0x76
done

Scanning...
I2C device found at address 0x68
I2C device found at address 0x76
done

Scanning...
I2C device found at address 0x68
I2C device found at address 0x76
done

Scanning...
I2C device found at address 0x68
I2C device found at address 0x76
done
  
```

Autoscroll  Mostrar marca temporal

Figura 5-47.: Código ejecutado con la librería I2C.

5. Se implementa el método que nos permite conocer si el dispositivo se conectó con la plataforma de TTN fig. 5-48, caso contrario nos debe mostrar un mensaje de error.



## 5.4 Creación y configuración de la aplicación y el dispositivo final en TTN

```
void loop() {
  switch (deviceState) {
    case DEVICE_STATE_INIT: {
      #if(LORAWAN_DEVEUI_AUTO)
        LoRaWAN.generateDeveuiByChipID();
      #endif
      LoRaWAN.init(loraWanClass, loraWanRegion);
      break;
    }
    case DEVICE_STATE_JOIN: {
      LoRaWAN.displayJoining();
      LoRaWAN.join();
      break;
    }
    case DEVICE_STATE_SEND: {
      LoRaWAN.displaySending();
      prepareTxFrame(appPort);
      LoRaWAN.send(loraWanClass);
      deviceState = DEVICE_STATE_CYCLE;
      break;
    }
    case DEVICE_STATE_CYCLE: {
      // Schedule next packet transmission
      txDutyCycleTime = appTxDutyCycle + randr(-APP_TX_DUTYCYCLE_RND, APP_TX_DUTYCYCLE_RND);
      LoRaWAN.cycle(txDutyCycleTime);
      deviceState = DEVICE_STATE_SLEEP;
      break;
    }
  }
}
```

**Figura 5-48.:** Código para la conexión al TTN.

6. Se implementa un método que nos permitirá recopilar y transmitir datos de un sensor, para este proyecto usamos un sensor de Presión BMP280 que nos permite medir la presión la temperatura y la altitud, fig. 5-49, estos datos serán recolectados por el servidor TTN y formateados con la biblioteca CayenneLPP instalado anteriormente.

```
static void prepareTxFrame(uint8_t port) {
  Serial.print("Time:");
  Serial.println(millis());
  Serial.print("Temperature = ");
  Serial.print(temp);
  Serial.println(" °C");

  Serial.print("Pressure = ");
  Serial.println(pa);
  Serial.println("hPa");
  Serial.print("Altitude = ");
  Serial.print(alt);
  Serial.println("m");
  Serial.println();
}
```

**Figura 5-49.:** Código para la recolección de datos con el sensor BMP 280.

7. Para poder cargar el código desde el IDE al HelTec ESP32 tenemos que considerar que los parámetros estén configurados correctamente como se ve en la fig. 5-50.

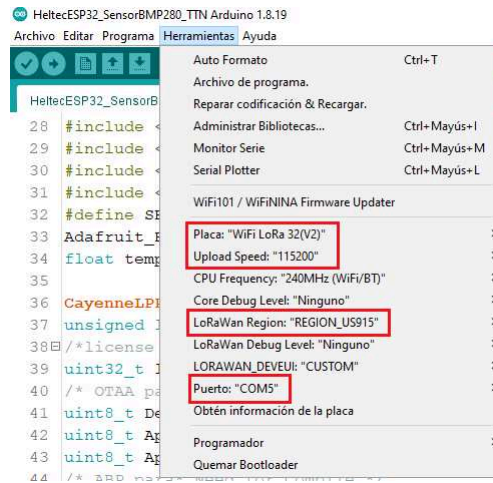
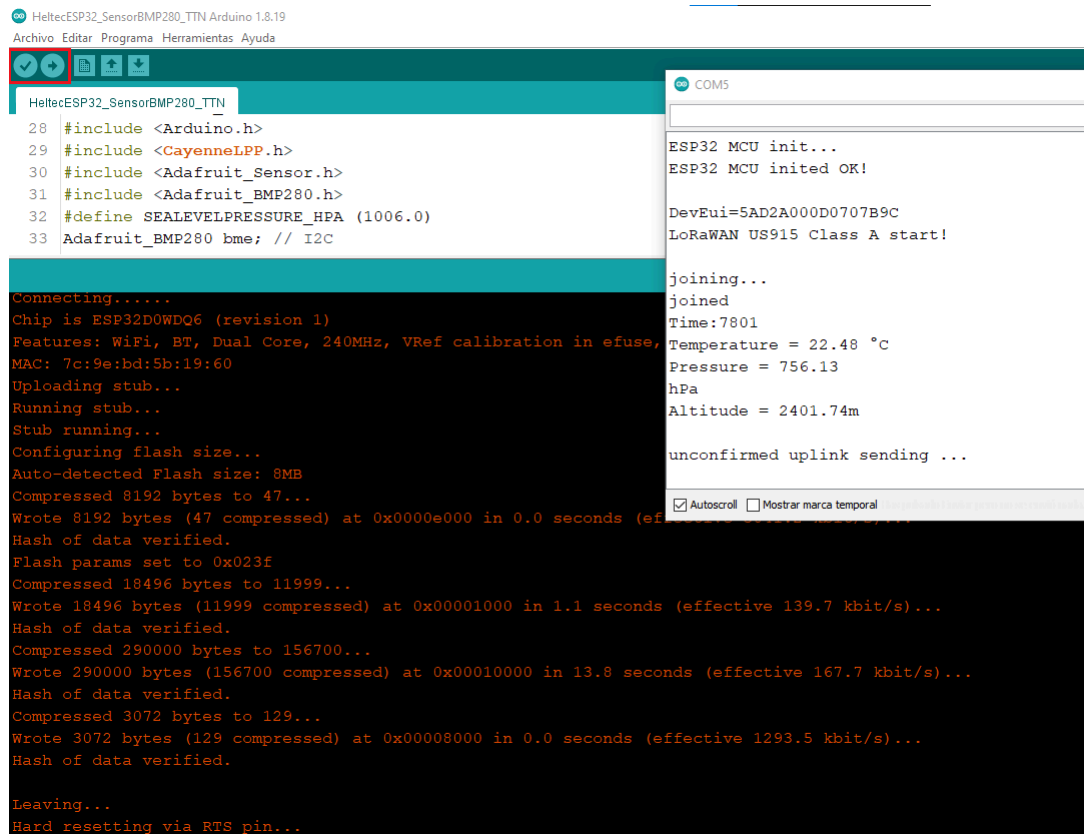


Figura 5-50.: Revisión de parámetros antes de cargar el código

8. Para cargar el código podemos usar los iconos de la parte superior izquierda, podemos hacer una **Verificación** o **Subir** nuestro código al dispositivo IoT. En la parte de la consola del IDE podemos observar los logs generados en la carga del código, en esta parte podemos verificar si existieran errores.

En el **Monitor Serie** podemos verificar los datos enviados por el sensor BMP280, eso nos indica que el código esta funcionando correctamente.

## 5.4 Creación y configuración de la aplicación y el dispositivo final en TTN



```
HeltecESP32_SensorBMP280_TTN Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda

HeltecESP32_SensorBMP280_TTN
28 #include <Arduino.h>
29 #include <CayenneLPP.h>
30 #include <Adafruit_Sensor.h>
31 #include <Adafruit_BMP280.h>
32 #define SEALEVELPRESSURE_HPA (1006.0)
33 Adafruit_BMP280 bme; // I2C

Connecting.....
Chip is ESP32D0WDQ6 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse,
MAC: 7c:9e:bd:5b:19:60
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 8MB
Compressed 8192 bytes to 47...
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.0 seconds (effective 12288 kbit/s)...
Hash of data verified.
Flash params set to 0x023f
Compressed 18496 bytes to 11999...
Wrote 18496 bytes (11999 compressed) at 0x00001000 in 1.1 seconds (effective 139.7 kbit/s)...
Hash of data verified.
Compressed 290000 bytes to 156700...
Wrote 290000 bytes (156700 compressed) at 0x00010000 in 13.8 seconds (effective 167.7 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 129...
Wrote 3072 bytes (129 compressed) at 0x00008000 in 0.0 seconds (effective 1293.5 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...

COM5
ESP32 MCU init...
ESP32 MCU init OK!

DevEui=5AD2A000D0707B9C
LoRaWAN US915 Class A start!

joining...
joined
Time:7801
Temperature = 22.48 °C
Pressure = 756.13
hPa
Altitude = 2401.74m

unconfirmed uplink sending ...

 Autoscroll  Mostrar marca temporal
```

Figura 5-51.: Carga y ejecución del código en HelTec ESP32.

9. Para revisar la conexión nos dirigimos a la página web de TTN e ingresamos en **Applications Overview** y se puede observar que la aplicación creada está activa y empieza a recibir datos de la conexión del dispositivo HelTec ESP32 como se muestra en fig. 5-52.

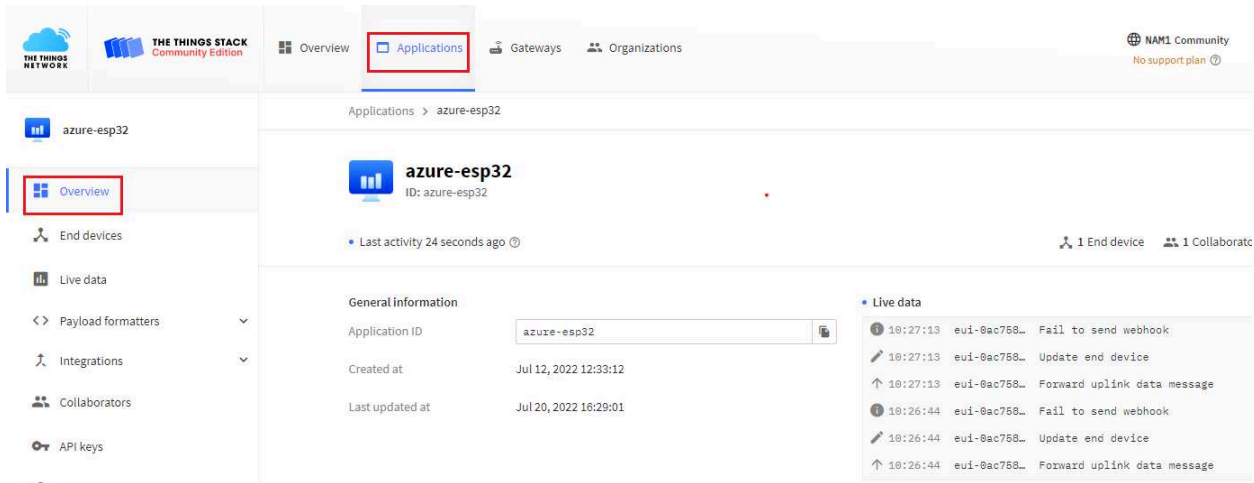


Figura 5-52.: Aplicación conectada

10. Luego nos dirigimos a **End devices** para ver nuestro dispositivo creado anteriormente, podemos observar que ya se conecto para ingresar le damos click en el nombre **BMP280** como se observa en la fig. 5-53.

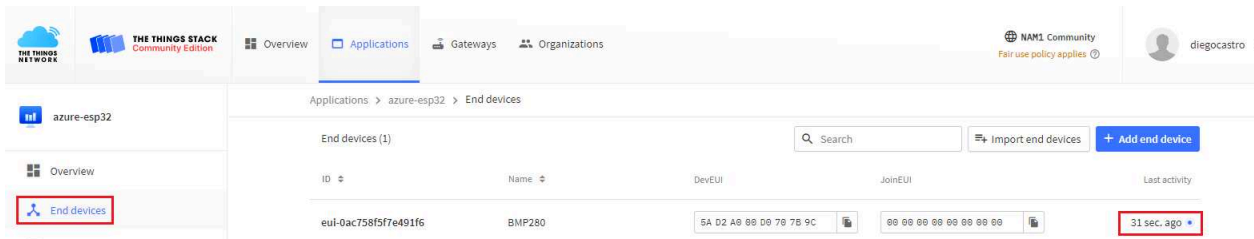


Figura 5-53.: Dispositivo HelTec ESP32 conectado

11. Una vez dentro del dispositivo creado podemos observar en la fig. 5-54 que el dispositivo creado empezó a recibir datos y ya podemos usarlo para recolectar información.

## 5.4 Creación y configuración de la aplicación y el dispositivo final en TTN

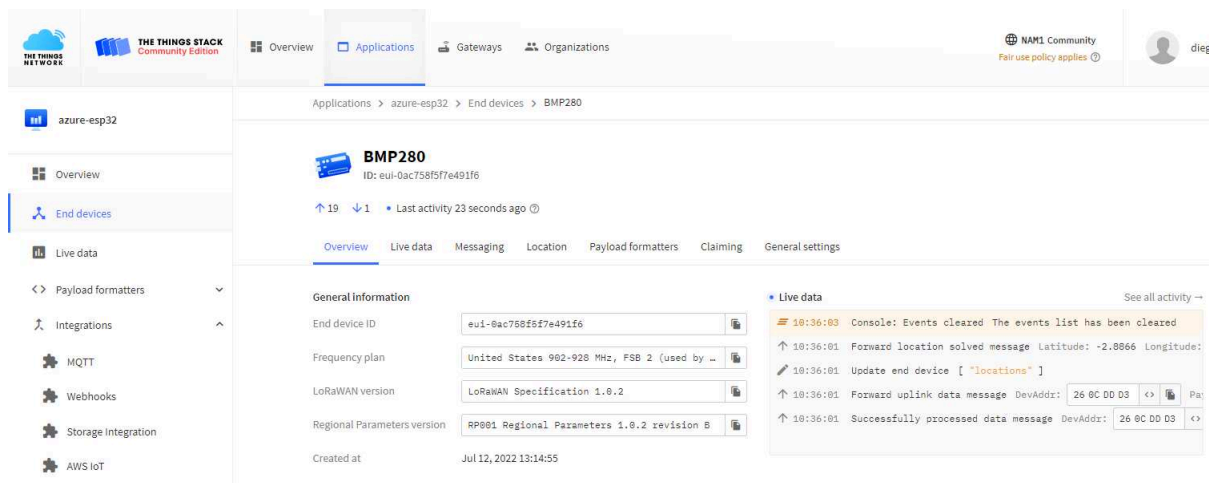


Figura 5-54.: Conexión establecida correctamente con ESP32 y BMP280 con TTN.

12. En el código de Arduino usamos un sensor de **Presión BMP280**, el cual implementamos una biblioteca de **CayenneLPP** la misma que en TTN nos da la opción de poder trabajar y poder formatear los datos que envía el sensor, esta biblioteca de **CayenneLPP** permite trabajar con dispositivos IoT y con múltiples sensores para poder formatear los datos que son enviados desde el sensor BMP280 hacia la plataforma de TTN.
13. Procedemos a entrar en la opción de **Payload Formatters**, luego a **Uplink** en **Formatter type** seleccionamos la opción **CayenneLPP** y guardamos los cambios con **Save changes** como se observa en la fig. 5-55.

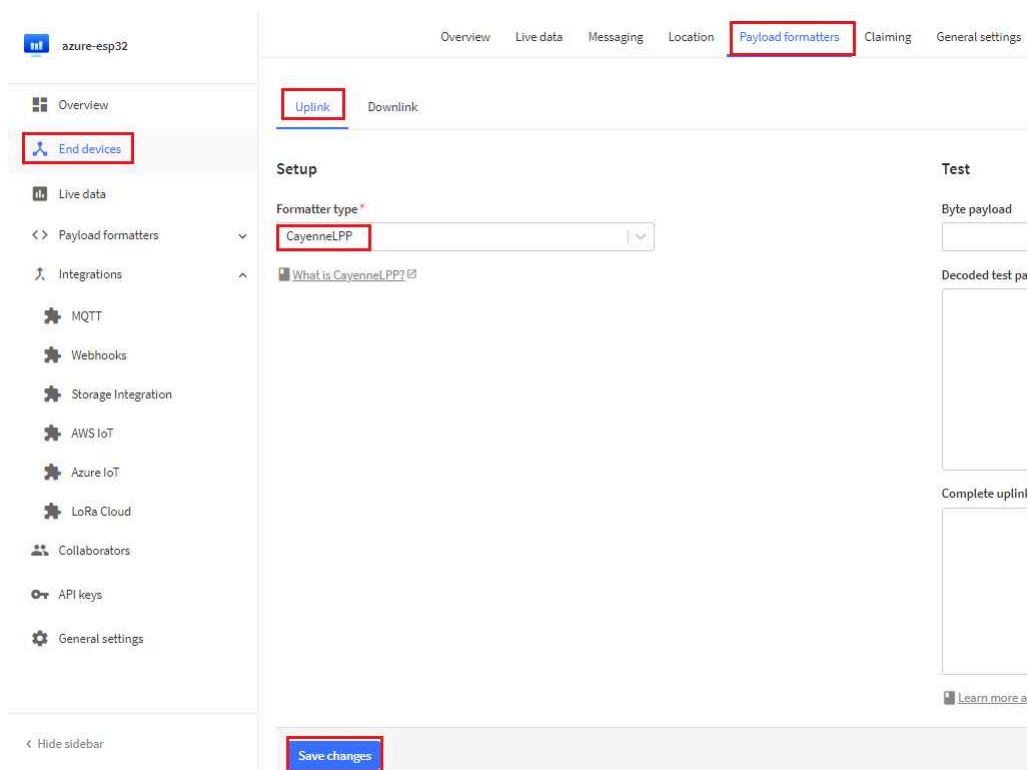


Figura 5-55.: Formateo de datos Uplink

14. Realizamos el mismo procedimiento para el Downlink, seleccionamos **Payload Formatters** luego a **Downlink** en **Formatter type** seleccionamos la opción **CayenneLPP** y guardamos los cambios en **Save changes** como se observa en la fig. 5-56.

## 5.4 Creación y configuración de la aplicación y el dispositivo final en TTN

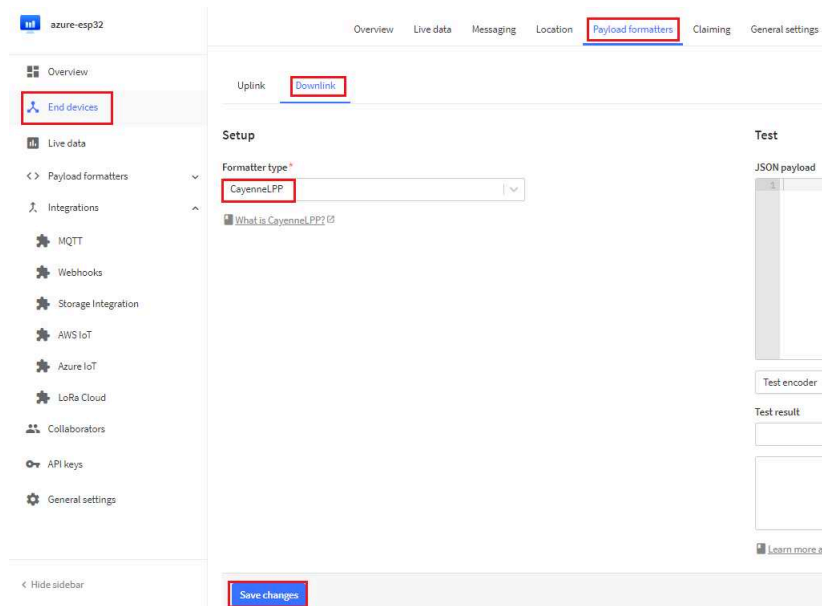


Figura 5-56.: Formateo de datos Downlink

15. Finalmente, nos dirigimos a **Live data**, se puede observar que los datos que genera el sensor se envían sin problema al TTN en formato JSON fig. 5-57, por lo que las configuración realizadas fueron correctas.

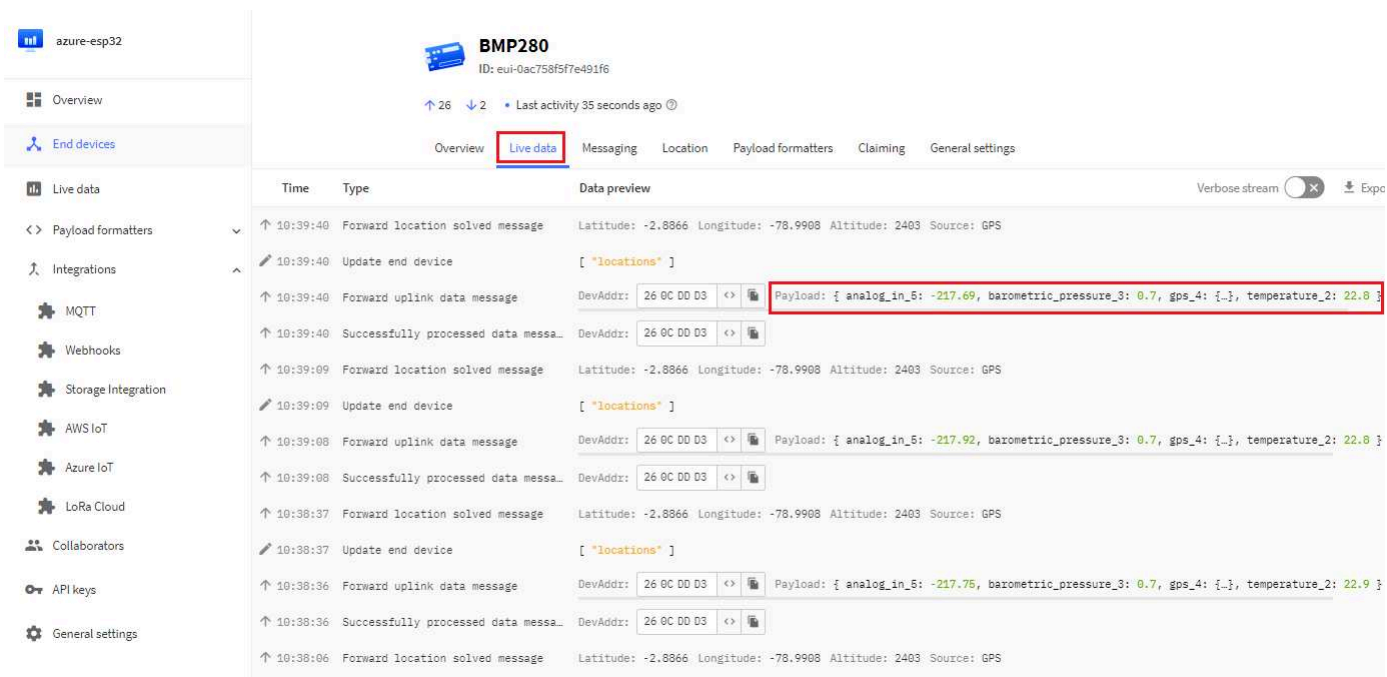


Figura 5-57.: Datos en tiempo real en TTN

## 5.5. Integración de Azure IoT Hub y TTN.

### 5.5.1. Crear cuenta en Azure IoT Hub

1. Primero ingresamos a la página oficial de **Azure IoT Hub** con la siguiente URL <https://azure.microsoft.com/en-us/services/iot-hub/> fig. 5-58, en donde para comenzar le damos click en **Start free**.

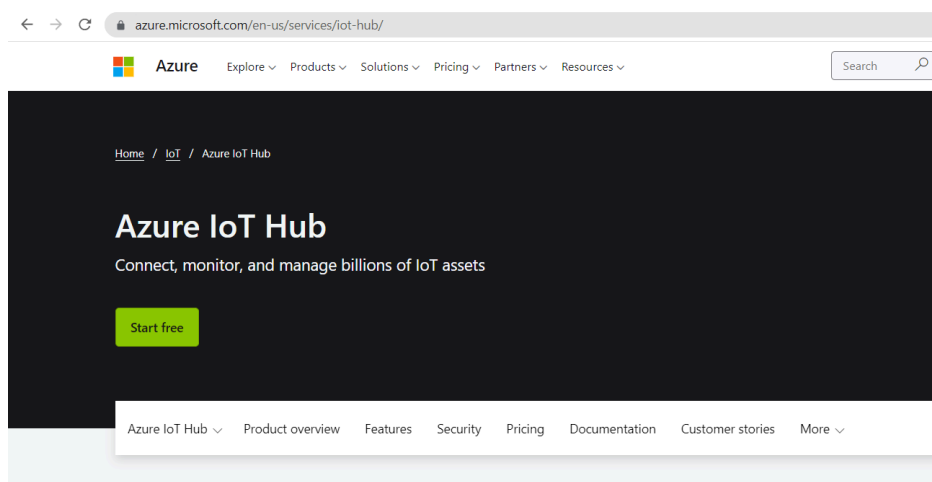


Figura 5-58.: Página de inicio Azure IoT Hub

2. Elegir un tipo de suscripción de pago como se muestra en fig. 5-59, ya que caso contrario no se puede hacer uso del centro de IoT Hub.

### Looks like you already have an Azure account

The Azure free account is only available to new users and is limited to one per customer

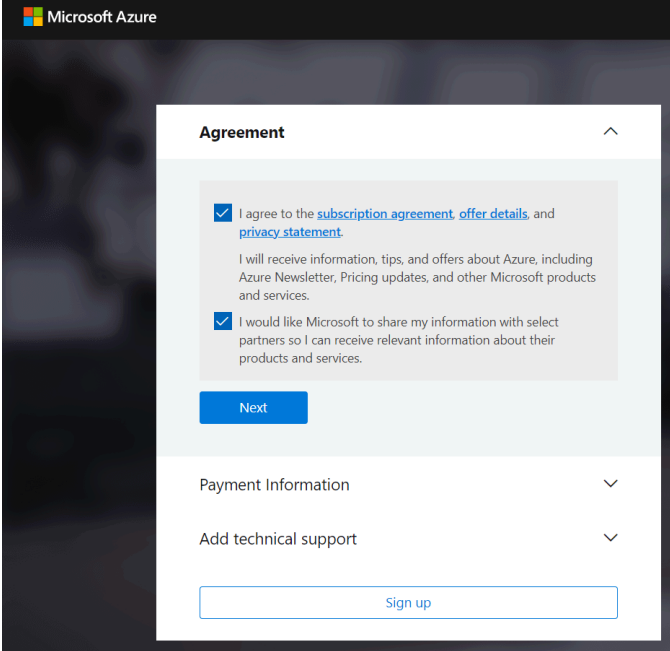
[Sign up for a Pay-As-You-Go subscription.](#)

[Use an existing subscription in your account](#)

Figura 5-59.: Suscripción de pago Azure IoT Hub.

3. Aceptamos los acuerdos de políticas para el uso de Azure IoT Hub y le damos click en **Sing up**, como se muestra en la fig. 5-60.

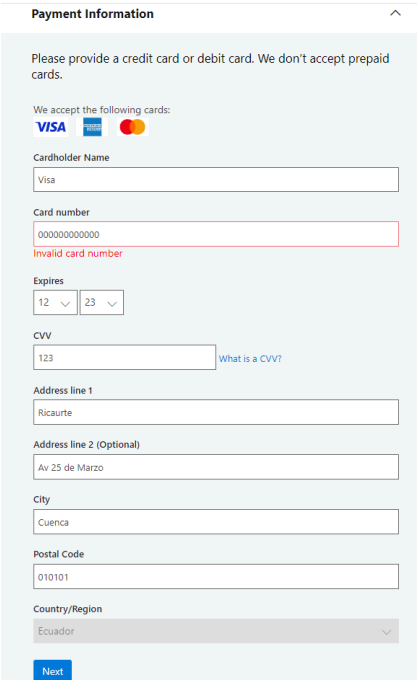




The screenshot shows the Microsoft Azure Agreement form. At the top left is the Microsoft Azure logo. The form is titled "Agreement" and contains two checked checkboxes. The first checkbox is for agreeing to the subscription agreement, offer details, and privacy statement. The second checkbox is for allowing Microsoft to share information with select partners. Below the checkboxes is a "Next" button. Underneath, there are two expandable sections: "Payment Information" and "Add technical support". At the bottom of the form is a "Sign up" button.

Figura 5-60.: Acuerdos Suscripción Azure IoT Hub.

4. En esta parte nos pedirá que ingresemos una tarjeta de crédito para la suscripción y procedemos a llenar el formulario con los datos de la tarjeta. Comprobamos que todos los datos ingresados sean correctos y le damos a **Next** como se visualiza en la fig. 5-61.



The screenshot shows the Payment Information form. It starts with a message: "Please provide a credit card or debit card. We don't accept prepaid cards." Below this, it lists the accepted cards: VISA, MASTERCARD, and AMERICAN EXPRESS. The form contains several input fields: Cardholder Name (with "Visa" entered), Card number (with "000000000000" entered and a red error message "Invalid card number" below it), Expires (with "12" and "23" selected in dropdown menus), CVV (with "123" entered and a "What is a CVV?" link), Address line 1 (with "Ricaurte" entered), Address line 2 (Optional) (with "Av 25 de Marzo" entered), City (with "Cuenca" entered), Postal Code (with "010101" entered), and Country/Region (with "Ecuador" selected in a dropdown menu). A "Next" button is located at the bottom left of the form.

Figura 5-61.: Formulario Información de Pago.

5. Debemos elegir el tipo de soporte, para nuestro caso de suscripción vamos a usar el ultimo ya que de esta manera no tenemos que hacer gastos extras. Por ultimo le damos click en **Regístrate** como se observa en la fig. 5-62.

Convenio ▼

Información del pago ▼

---

**Añadir soporte técnico** ▲

---

Para soporte técnico personalizado, necesitará un [plan de soporte](#) . Ya sea que sea nuevo en Azure o que ya esté creando aplicaciones críticas para el negocio, nuestros planes de soporte brindan orientación experta de ingenieros de Azure que trabajarán con usted para prevenir y resolver problemas. Al actualizar a un plan, acepta los [detalles de la oferta](#) .

**Primeros pasos en Azure - Desarrollador \$29/mes**  
Para las personas o los equipos que buscan comenzar a usar Azure de manera rápida y eficaz, el soporte técnico está disponible los días de semana de 9:00 a. m. a 5:00 p. m. con tiempos de respuesta iniciales de menos de 8 horas hábiles.

**Uso de producción de Azure - Estándar \$100/mes**  
Para equipos que ejecutan cargas de trabajo de producción. El tiempo de respuesta inicial es entre una hora y un día hábil, según la gravedad del caso.

**Uso crítico para la empresa de Azure - Professional Direct \$1,000/mes**  
Para equipos que ejecutan varias funciones críticas para el negocio y necesitan respuestas rápidas, servicios de asesoramiento y acceso a seminarios web de expertos de Microsoft. Tiempo de respuesta inicial entre una hora y un día hábil, según la gravedad del caso, e incluye gestión de incidentes de alta gravedad.

No tengo soporte técnico, o ya estoy cubierto por el soporte Premier de Microsoft.

[Regístrate](#)

**Figura 5-62.:** Tipo de Soporte Azure IoT Hub.

6. Una vez finalizado correctamente la suscripción ya podemos ingresar a la página principal del portal de Azure IoT Hub para lo cual le damos a **Go to the Azure portal** fig. 5-63.

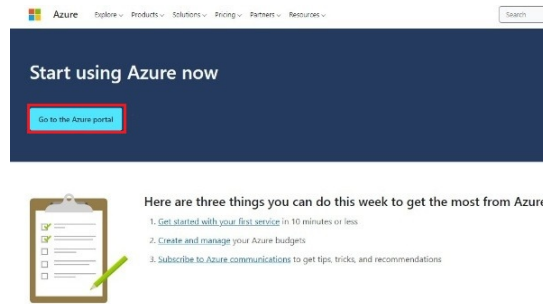


Figura 5-63.: Página de inicio a los recursos de Azure IoT Hub.

7. Podemos observar la página de inicio de Microsoft IoT Hub, donde ya podemos comenzar a trabajar creando grupos de recursos, servicios, entre otros como se ilustra en la fig. 5-64.

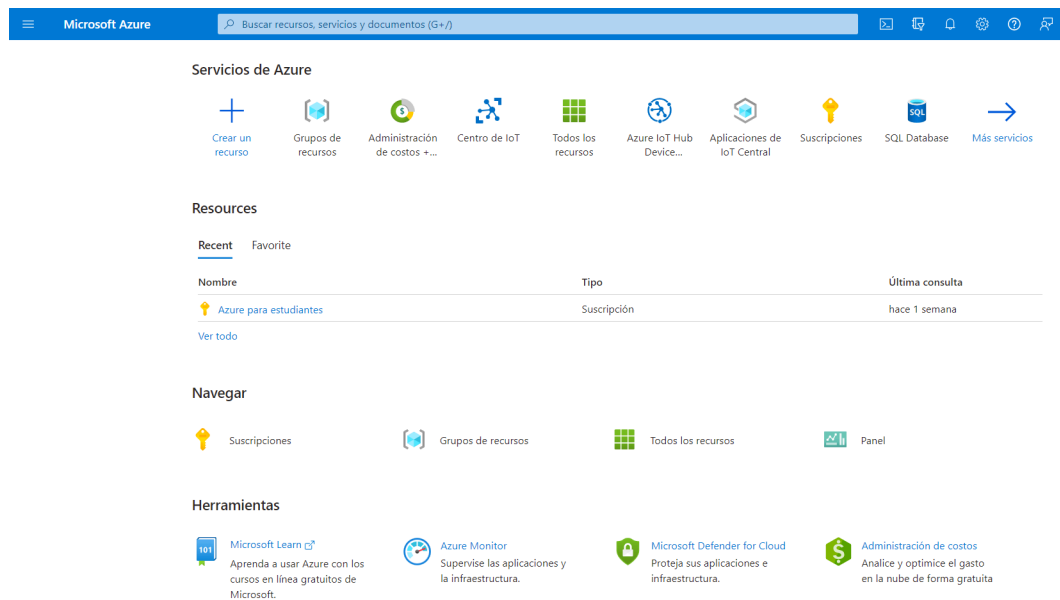


Figura 5-64.: página de inicio de Azure IoT Hub.

## 5.5.2. Integración de Azure IoT Hub con la plataforma de TTN

### Paso 1. Seleccionamos la opción de Integration en TTN.

1. Primero nos dirigimos a la parte de **Applications** en TTN.
2. Luego desplegamos la opción de **Integrations** demos seleccionar la de **Azure IoT**.
3. Luego nos dirigimos al lado izquierdo le damos click a **Expand** junta a Azure IoT Hub tal como se ilustra en la fig. 5-65.

4. Devemos darle click en **Generate API Key** ya que esta clave tenemos que insertar el la integración de la plantilla en Azure IoT Hub.

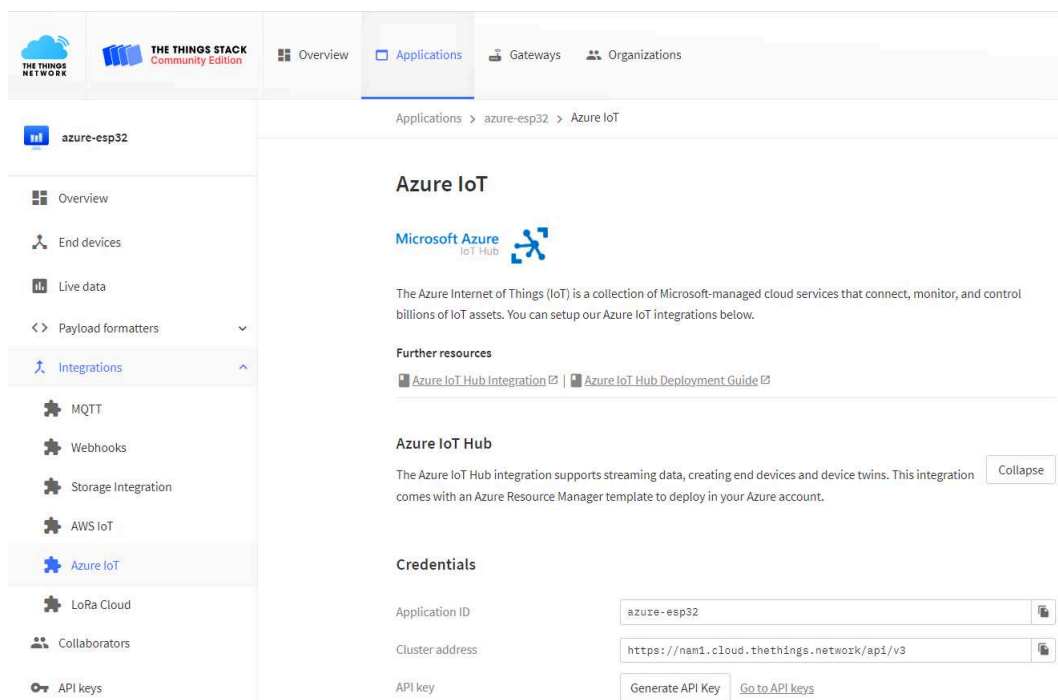


Figura 5-65.: Integración de Azure IoT Hub.

## Paso 2. Plantilla de Integración Azure IoT Hub.

1. En la guía de integración de TTN podemos encontrar la plantilla personalizada de Azure IoT Hub la cual solo debemos llenar con los datos correspondientes, esta plantilla la pueden encontrar en el siguiente Link: <https://www.thethingsindustries.com/docs/integrations/cloud-integrations/azure-iot-hub/deployment/>
2. Una vez Cargada la plantilla observamos los siguientes parámetros fig. 5-66:

**Location:** Elegir la Región en la que se va a implementar los recursos de Azure IoT Hub.

**Name Suffix:** Agregar un nombre para reconocer al recurso creado.

**Stack Cluster Address:** Nombre del cluster en nuestro caso debemos copiar el Cluster address del servidor de TTN.

**Stack Application ID:** Ingresamos la Application ID o nombre de la aplicación que fue creada en TTN.

**Stack API Key:** Copiar el APPI Key generado en el paso 1.

3. Una vez completado los campos de la plantilla para la creación del recurso de Azure IoT Hub con los datos correctos le damos click en **Revisar y Crear**.

**Implementación personalizada** ...  
Implementar desde una plantilla personalizada

Suscripción \*

Grupo de recursos \*  [Crear nuevo](#)

**Detalles de la instancia**

Región \*

Location

Name Suffix

Stack Cluster Address

Stack Application ID \*

Stack API Key \*

Event Hub Name Space SKU

IoT Hub SKU

IoT Hub Capacity

Storage Account SKU

App Service Plan SKU

App Service Plan Tier

Enable Fallback Route

[Revisar y crear](#) < Anterior Siguiente: Revisar y crear >

**Figura 5-66.:** Plantilla de Azure IoT Hub para la integración con TTN.

### Paso 3. Revisar y crear Recurso de IoT Hub.

1. Esperamos que la validación sea correcta tal como se observa en la fig. 5-67, por último le damos click en **Crear**.



Figura 5-67.: Creación del recurso Azure IoT Hub.

#### Paso 4. Creando recurso de Azure IoT Hub.

1. Debemos esperar a que finalice la creación del recurso de Azure IoT Hub esto puede tardar varios minutos, una vez que finalizado el proceso se debe observar que se completo correctamente fig. 5-68.

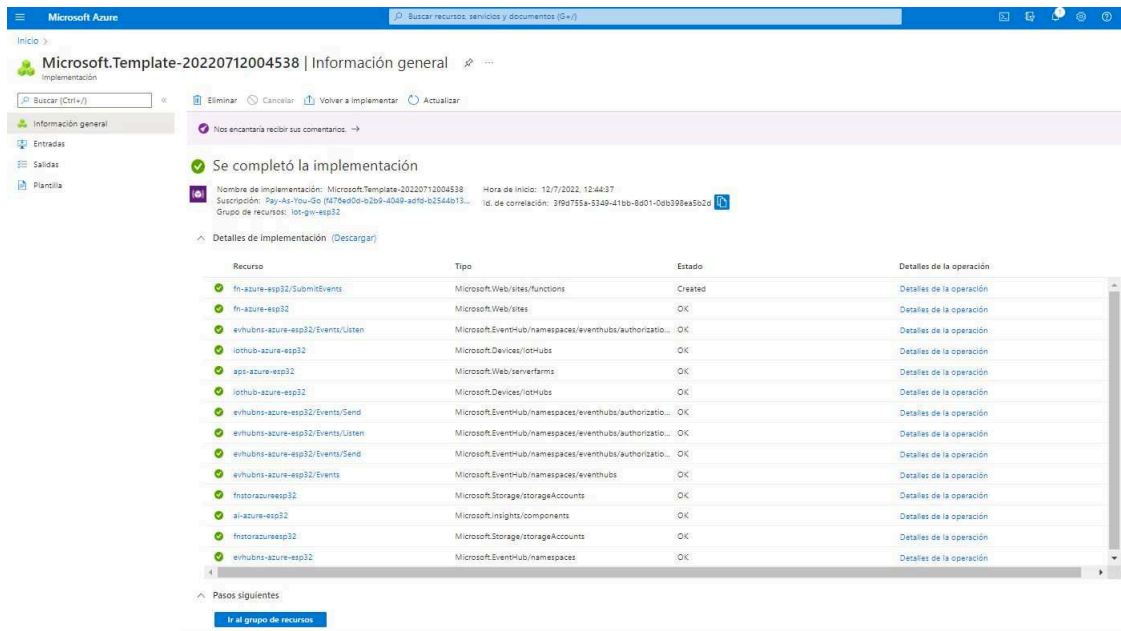


Figura 5-68.: Implementación Correcta del recurso de Azure IoT Hub.

### Paso 5. Integrar el hostname y access key de TTN en Azure IoT Hub.

1. Una vez finalizada la implementación nos dirigimos a **Salidas** fig. 5-69 y pegamos en los siguientes campos:

**Azure IoT Hub hostname:** Copiamos el hostname de la página de integración de Azure a la página integración de TTN.

**Azure IoT Hub access key:** Copiamos la clave de acceso de la página de Azure a la página de integración de TTN.



Figura 5-69.: Configuración de Salida de Azure IoT Hub .

### Paso 6. Actualizar la integración de Azure IoT Hub en TTN.

1. Por ultimo nos dirigimos a la página de Integración de TTN le damos click en **Update Azure IoT Hub integration** como se observa en la fig. 5-70.

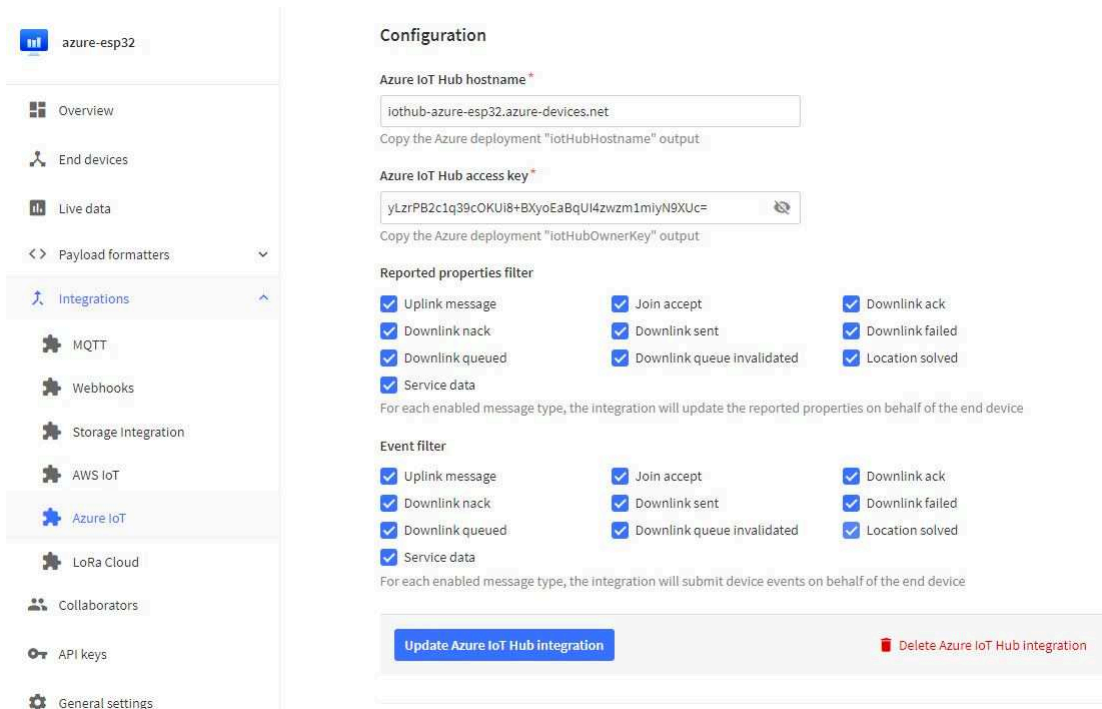


Figura 5-70.: Finalización integración Azure IoT Hub y TTN.

## Paso 7. Ver dispositivo creado en el centro de IoT Hub.

1. Para saber si la integración fue correcta en el centro IoT creado nos dirigimos a el apartado de **Dispositivos** en el recurso de Azure IoT Hub donde podemos observar fig. 5-71 que el dispositivo que fue creado en la plataforma de TTN fue reconocido automáticamente y ya podemos comenzar a gestionar y recolectar información de este dispositivo.

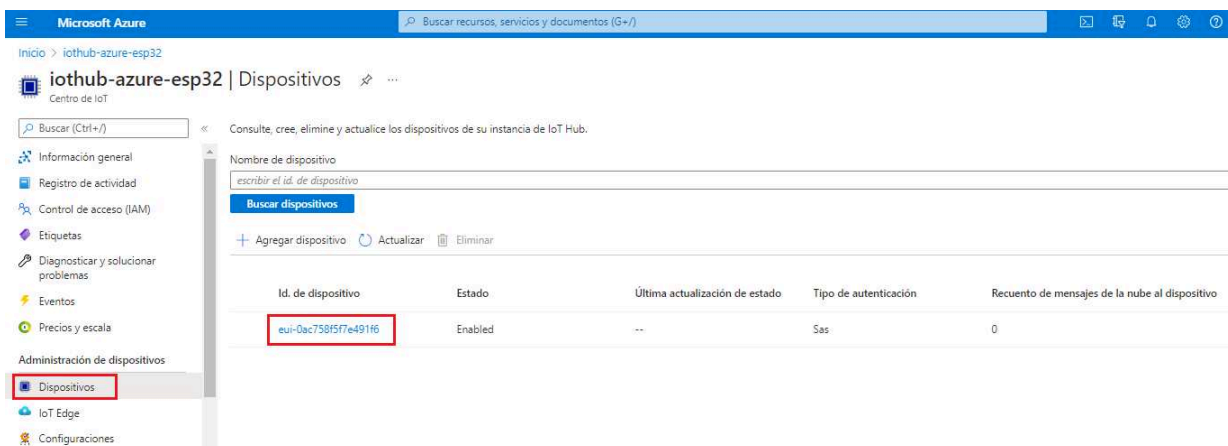


Figura 5-71.: Dispositivo detectado en Azure IoT Hub.



### 5.5.3. Implementación/Instalación de Servicios para Recolectar y Interpretar los datos del dispositivo HelTec ESP32-Sensor de Presión BMP280.

#### 5.5.4. Azure Stream Analytics

- Buscamos el servicio a implementar en el buscador de Microsoft Azure en este caso comenzamos con Trabajos de Stream Analytics como se observa en la fig. 5-72.

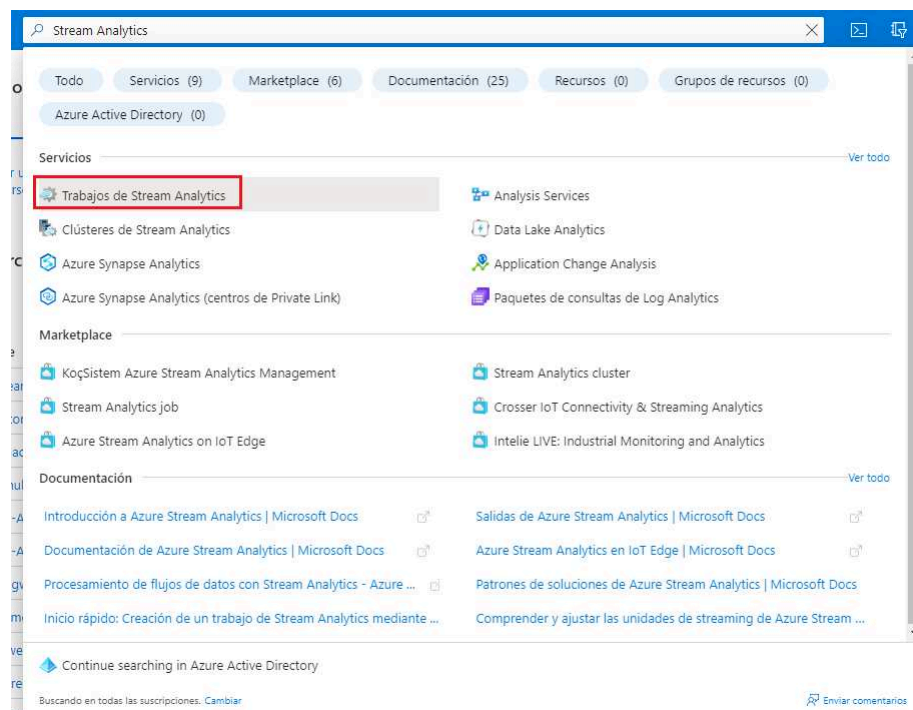


Figura 5-72.: Instalar servicio Stream Analytics.

- Dar click en **crear** para crear un nuevo servicio de Stream Analytics fig. 5-73.

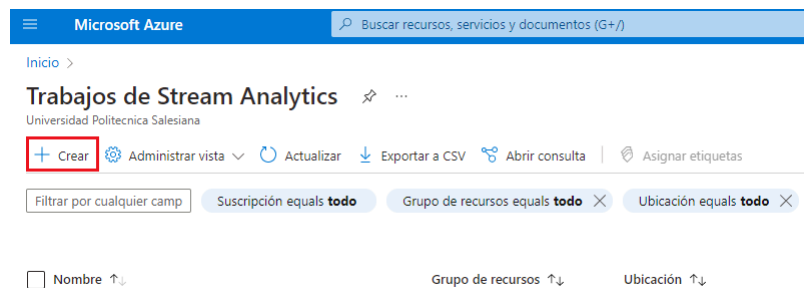


Figura 5-73.: Crear el servicio.

- Configuramos los siguientes campos fig. 5-74:

**Nombre del trabajo:** Nombre único para el servicio.

**Suscripción:** Seleccionamos la suscripción de la cuenta.

**Grupo de recursos:** Es el nombre que le damos al grupo de todos los recursos que vamos a crear, es como la carpeta principal que contendrá todo nuestros servicios y en el caso de no tener creado podemos hacerlo desde aquí.

**Ubicación:** Seleccionamos la ubicación para nuestro servicio, lo recomendable es usar la misma ubicación para todos lo servicios.

Microsoft Azure

Buscar recursos, servicios y documentos (G+)

Inicio > Trabajos de Stream Analytics >

### Nuevo trabajo de Stream Analytics

Esta operación creará un nuevo trabajo de Stream Analytics, y se le cobrará de acuerdo con el modelo de facturación de Azure Stream Analytics.

Nombre del trabajo \*

StreamDispositivo

Suscripción \*

Pay-As-You-Go (f476ed0d-b2b9-4049-adfd-b2544b...)

Grupo de recursos \*

iot-gw-esp32

Ubicación \*

South Central US

Entorno de hospedaje

Nube Perimetral

Unidades de streaming (entre 1 y 396)

3

Proteja todos los recursos de datos privados necesarios para este trabajo en su cuenta de almacenamiento.

Crear

Figura 5-74.: Crear servicio Stream Analytics

### 5.5.5. Azure Cosmos DB

- Buscamos el servicio Azure Cosmos DB en el buscador de Microsoft Azure fig. 5-75.

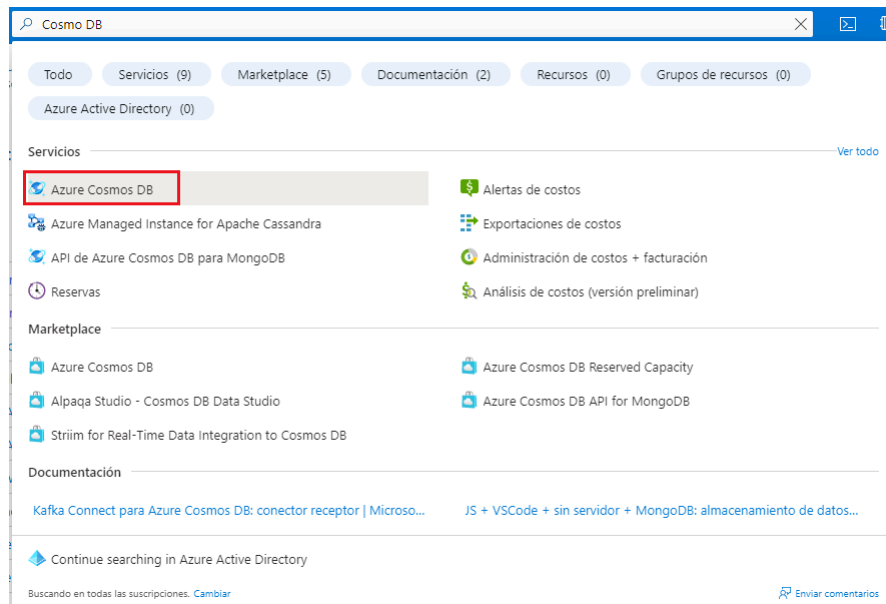


Figura 5-75.: Instalar el servicio Cosmos DB.

- Verificar que el servicio sea el correcto, dar click en **crear** para crear un servicio de Cosmos DB fig. 5-76.

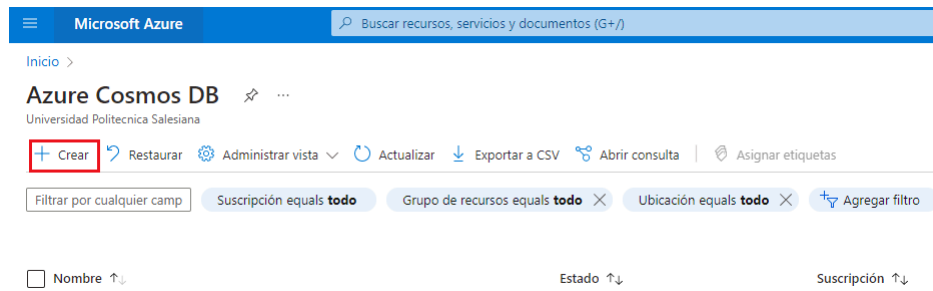


Figura 5-76.: Crear el servicio

- Seleccionamos la API de Núcleo (SQL) fig. 5-77 ya que vamos a trabajar con base de datos tipo sql.

Microsoft Azure

Inicio > Azure Cosmos DB >

### Selección de la opción de API

¿Qué API resulta más adecuada para la carga de trabajo?

Azure Cosmos DB es un servicio de base de datos NoSQL totalmente administrado para la creación de aplicaciones escalables y de alto rendimiento. [Learn more](#)

Para empezar, seleccione la API para crear una cuenta nueva. La selección de la API no se puede cambiar tras crear la cuenta.

**Núcleo (SQL): opción recomendada**

Núcleo o API nativa de Azure Cosmos DB para trabajar con documentos. Permite un desarrollo rápido y flexible con el lenguaje de consultas SQL y las bibliotecas cliente para .NET, JavaScript, Python y Java que ya conoce.

[Crear](#) [Más información](#)

**API de Azure Cosmos DB para MongoDB**

Servicio de base de datos de MongoDB totalmente administrado para aplicaciones escritas para MongoDB. Se recomienda si dispone de cargas de trabajo existentes de MongoDB que tenga previsto migrar a Azure Cosmos DB.

[Crear](#) [Más información](#)

**Cassandra**

Servicio de base de datos de Cassandra totalmente administrado para aplicaciones escritas para Apache Cassandra. Se recomienda si dispone de cargas de trabajo existentes de Cassandra que tenga previsto migrar a Azure Cosmos DB.

[Crear](#) [Más información](#)

**Tabla de Azure**

Servicio de base de datos totalmente administrado para aplicaciones escritas para el almacenamiento de Azure Table. Se recomienda si dispone de cargas de trabajo existentes del almacenamiento de Azure Table que tenga previsto migrar a Azure Cosmos DB, pero no quiere volver a escribir la aplicación para usar la API SQL.

[Crear](#) [Más información](#)

**Gremlin (grafo)**

Servicio de base de datos de grafos totalmente administrado que emplea el lenguaje de consultas Gremlin, basado en el proyecto Apache TinkerPop. Se recomienda para nuevas cargas de trabajo en las que sea necesario almacenar relaciones entre datos.

[Crear](#) [Más información](#)

Figura 5-77.: Seleccionar el API para la instalación

- Configurar lo siguientes campos fig. 5-78:
  - Suscripción:** Seleccionamos la suscripción de la cuenta.
  - Grupo de Recursos:** Seleccionamos el grupo de recursos.
  - Nombre de la cuenta:** Definimos un nombre para la cuenta.
  - Ubicación:** Seleccionamos la ubicación.
- El resto de configuraciones lo dejamos por defecto.

Microsoft Azure Buscar recursos, servicios y documentos (G+)

Inicio > Azure Cosmos DB > Selección de la opción de API >

### Creación de una cuenta de Azure Cosmos DB - Core (SQL) ...

**Detalles del proyecto**  
Seleccione la suscripción para administrar los recursos y costos implementados. Use grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción \*

Grupo de recursos \*   
[Crear nuevo](#)

**Detalles de la instancia**

Nombre de cuenta \*

Ubicación \*

Modo de capacidad ⓘ  Rendimiento aprovisionado  Serverless  
[Obtenga más información sobre el modo de capacidad.](#)

With Azure Cosmos DB free tier, you will get the first 1000 RU/s and 25 GB of storage for free in an account. You can enable free tier on up to one account per subscription. Estimated \$64/n

La suscripción que ha seleccionado ya tiene una cuenta con el nivel gratis habilitado.

Aplicar descuento del nivel Gratis  Aplicar  No aplicar

Limitar el rendimiento total de la cuenta  Limitar la cantidad total de rendimiento que se puede aprovisionar en esta cuenta  
**i** Este límite evitará cargos inesperados relacionados con el rendimiento aprovisionado. Puede actualizar o quitar este límite después de crear la cuenta.

Figura 5-78.: Formulario para crear servicio de Cosmos DB.

- Finalizamos con click en **Revisar y Crear** como se visualiza en la fig. 5-79.

Microsoft Azure

Buscar recursos, servicios y documentos (G+)

Inicio > Azure Cosmos DB > Selección de la opción de API >

## Creación de una cuenta de Azure Cosmos DB - Core (SQL) ...

✓ Validación correcta

### Hora de creación

Tiempo estimado de creación de la cuenta (en minutos) 1

La hora de creación estimada se calcula en función de la ubicación seleccionada.

### Conceptos básicos

Suscripción	Pay-As-You-Go
Grupo de recursos	iot-gw-esp32
Ubicación	South Central US
Nombre de cuenta	(nuevo) cosmo-dbdatas
API	Core (SQL)
Modo de capacidad	Rendimiento aprovisionado
Redundancia geográfica	Deshabilitar
Escrituras en varias regiones	Deshabilitar
Zonas de disponibilidad	Deshabilitar

### Directiva de copia de seguridad

Directiva de copia de seguridad	Periodic
Redundancia de almacenamiento de copias de seguridad	Almacenamiento de copias de seguridad con redundancia geográfica

**Crear** Anterior Siguiente [Descargar una plantilla para la automatización](#)

Figura 5-79.: Crear el servicio Cosmos DB

### 5.5.6. Azure Storage Account

- Escribir el servicio en el buscador de Microsoft Azure y click en crear fig. 5-80.

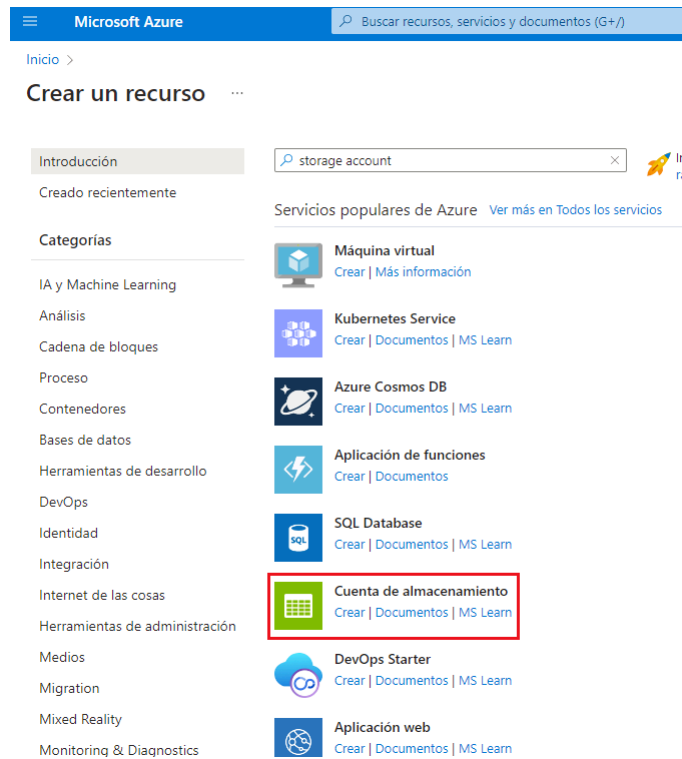


Figura 5-80.: Buscar y crear el servicio

- Configurar los siguientes campos fig. 5-81:
  - Grupo de recursos:** Seleccionamos nuestro grupo de recursos.
  - Nombre de la cuenta de almacenamiento:** Nombre para el almacenamiento.
  - Región:** Seleccionamos la región.
- El resto de configuraciones lo dejamos por defecto.

Microsoft Azure

Buscar recursos, servicios y documentos (G+)

Inicio > Crear un recurso >

### Crear una cuenta de almacenamiento

Datos básicos | Opciones avanzadas | Redes | Protección de datos | Cifrado | Etiquetas | Revisar y crear

Grupo de recursos \*  [Crear nuevo](#)

#### Detalles de la instancia

Si necesita crear un tipo de cuenta de almacenamiento heredada, haga clic en [aquí](#).

Nombre de la cuenta de almacenamiento

Región

Rendimiento  Estándar: Opción recomendada para la mayoría de los escenarios (cuenta de uso general v2)

Prémium: Se recomienda para escenarios que requieren una latencia baja.

Redundancia

Habilite el acceso de lectura a los datos en el caso de que la región no esté disponible.

[Revisar y crear](#) < Anterior Siguiente: Opciones avanzadas >

**Figura 5-81.:** Formulario para crear servicio Storage Account.

- Finalmente click en **Crear** fig. 5-82.



Microsoft Azure

Inicio > Crear un recurso >

### Crear una cuenta de almacenamiento

Validación superada

Datos básicos | Opciones avanzadas | Redes | Protección de datos | Cifrado | Etiquetas | Revisar y crear

#### Datos básicos

Suscripción	Pay-As-You-Go
Grupo de recursos	iot-gw-esp32
Ubicación	southcentralus
Nombre de la cuenta de almacenamiento	almacenamientosbmp280
Modelo de implementación	Resource Manager
Rendimiento	Standard
Replicación	Almacenamiento con redundancia geográfica con acceso de lectura (RA-GRS)

#### Opciones avanzadas

Transferencia segura	Habilitado
Permitir el acceso a la clave de la cuenta de almacenamiento	Habilitado
Permitir replicación entre espacios empresariales	Habilitado
Usar la autorización de Azure Active Directory como predeterminada en Azure Portal	Deshabilitado

**Crear** < Anterior Siguiente > [Descargar una plantilla para la automatización](#)

Figura 5-82.: Crear el servicio

- Concluida la instalación del servicio, no dirigimos al servicio creado, en el menú seleccionamos la opción de **Contenedores** y **Agregamos un contenedor** tal como se muestra en la fig. 5-83.

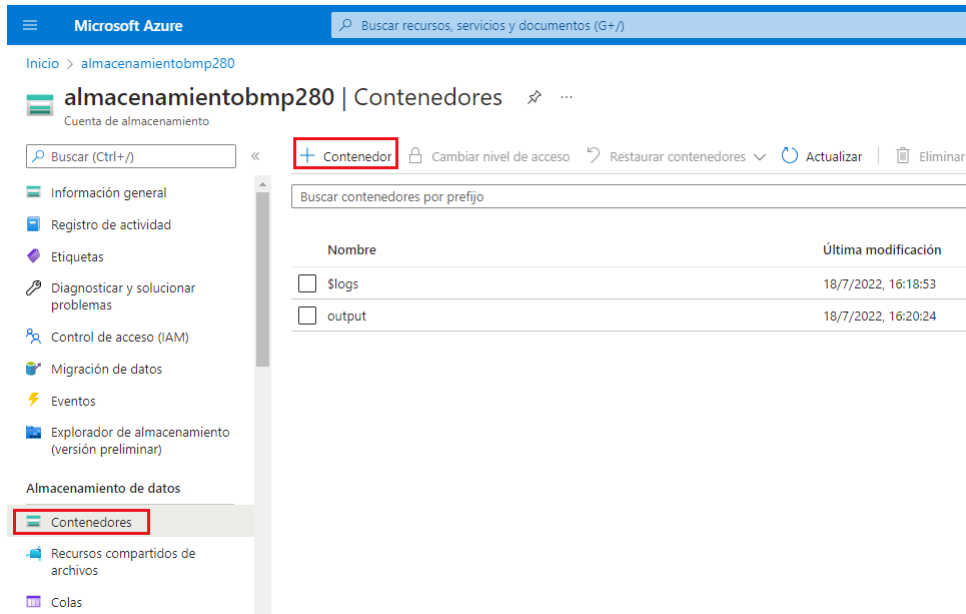


Figura 5-83.: Crear un contenedor de almacenamiento.

- En la fig. 5-84 vamos a crear un nuevo contenedor para o cual procedemos a ingresar el nombre al contenedor y el nivel de acceso sera privada. Finalmente click en **Crear**.

Nuevo contenedor ×

Nombre \*  
output1 ✓

Nivel de acceso público ⓘ  
Privada (sin acceso anónimo) ▼

▼ Avanzado

Crear Descartar

Figura 5-84.: Completar el formulario para el contenedor

### 5.5.7. Azure Power BI

- Nos dirigimos al siguiente link: <https://azure.microsoft.com/es-es/services/developer-tools/power-bi/> fig. 5-85, escogemos la opción gratis.

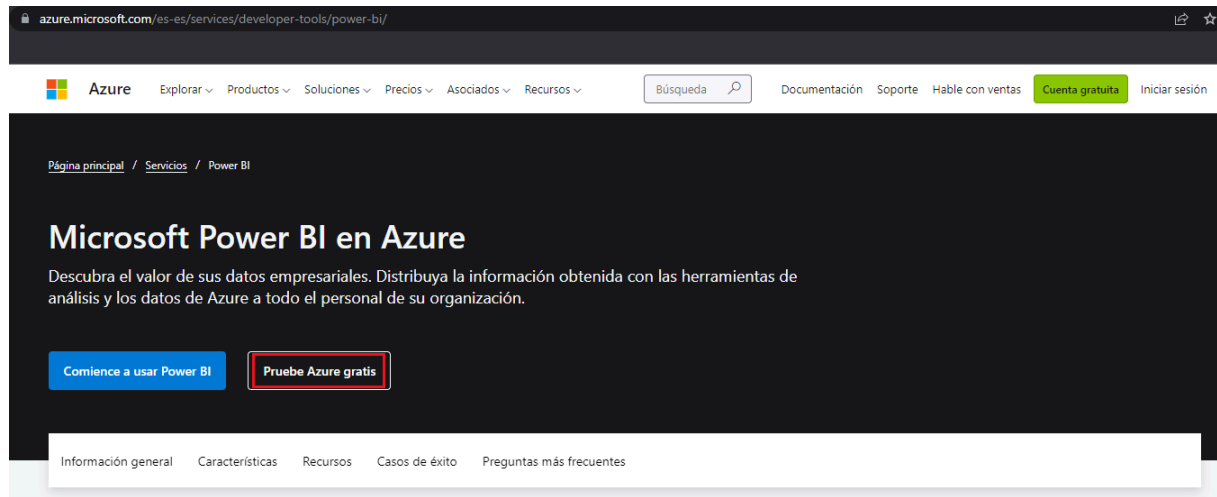


Figura 5-85.: página de inicio de Power BI

- Seleccionamos la primera opción para la suscripción de uso por Pago fig. 5-86.



Figura 5-86.: Tipo de suscripción

- Buscamos el servicio de Power BI en el buscador de Microsoft Azure fig. 5-87.

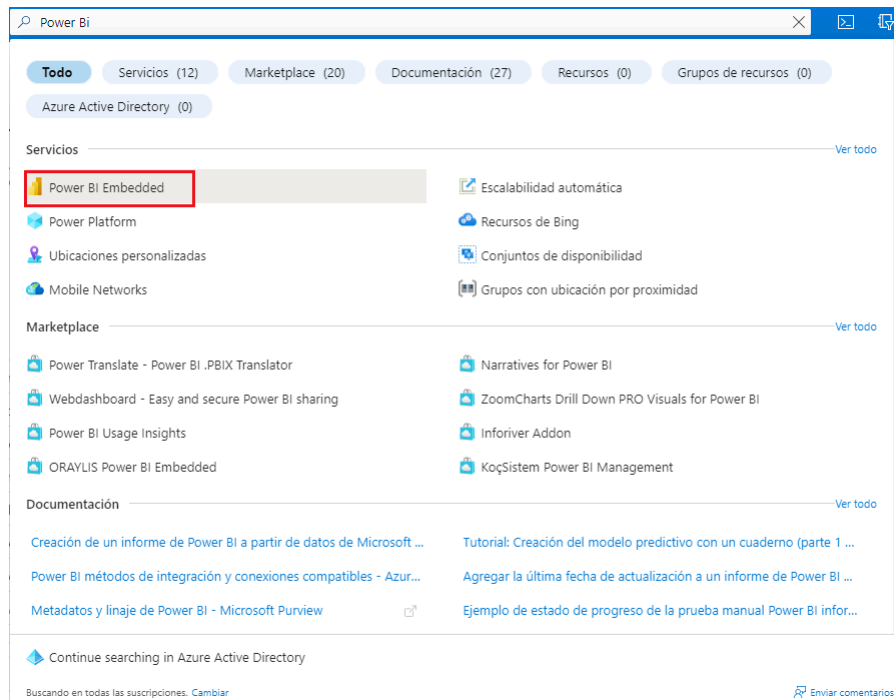


Figura 5-87.: Instalar el servicio Power BI.

- Click en **Crear** fig. 5-88 para un nuevo servicio de Power BI.

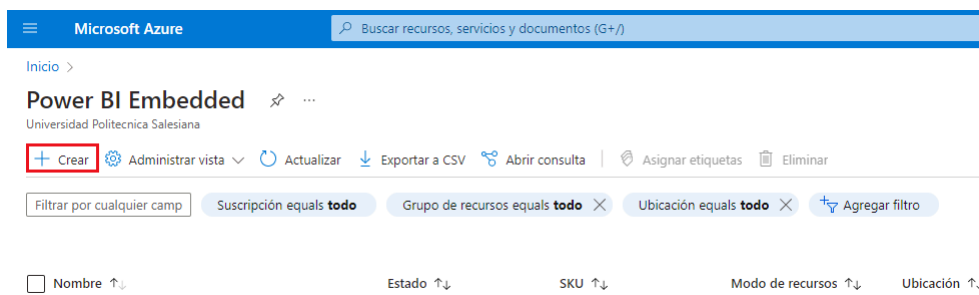


Figura 5-88.: Crear el servicio.

- Configuramos los siguientes campos fig. 5-89:
  - Suscripción:** Seleccionamos el tipo de suscripción de la cuenta.
  - Grupo de recursos:** Seleccionamos el grupo de recursos.
  - Nombre del recurso:** Asignamos un nombre cualquiera.
  - Ubicación:** Seleccionamos la ubicación.
- El resto de configuraciones lo dejamos por defecto.

Microsoft Azure

Buscar recursos, servicios y documentos (G+/)

Inicio > Power BI Embedded >

## Power BI Embedded

\* Datos básicos Etiquetas Revisar y crear

### DETALLES DEL PROYECTO

Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción \*

Grupo de recursos \*  [Crear nuevo](#)

### DETALLES DEL RECURSO

Nombre de recurso \*

Ubicación \*

Tamaño **A1**  
1 núcleo virtual y 3 GB de memoria  
[Cambiar el tamaño](#)

Administrador de capacidad de Power BI  [Seleccionar](#)

**Revisar y crear** Siguiente: Etiquetas >

Figura 5-89.: Completar los campos del formulario

- Finalmente en **Revisar y crear** el servicio fig. 5-90.
- Para una mejor experiencia de usuario se recomienda instalar el Power BI para escritorio.

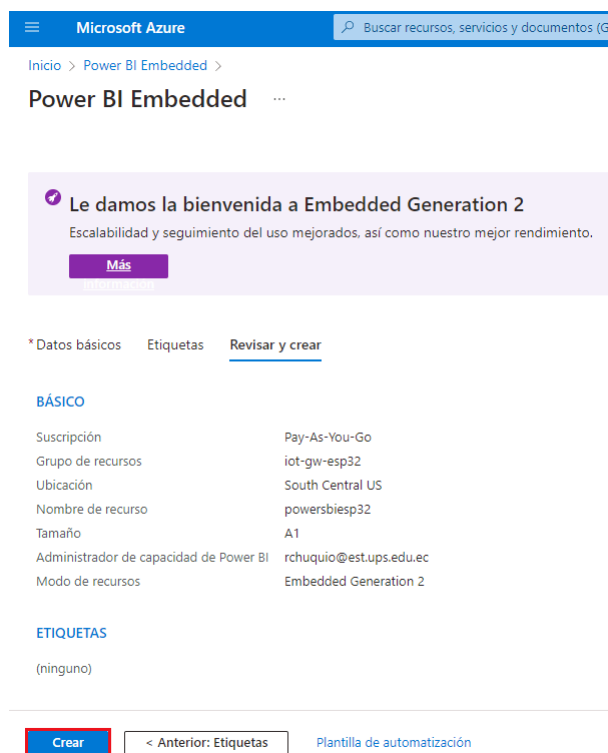


Figura 5-90.: Crear el servicio.

### 5.5.8. Integración de Cosmos DB, Storage Account y Power BI en Azure Stream Analytics.

#### 1. Configuración del explorador de datos Cosmos DB.

- Para configurar al servicio debemos acceder a nuestro grupo de recursos, para nuestro ejemplo es el **iot-gw-esp32**, aquí estarán guardados todos nuestros servicios, seleccionamos el **cosmo-dbdata**, una vez ingresado a este servicio seleccionamos la opción de **Explorador de datos** como se puede observar en la fig. 5-91.

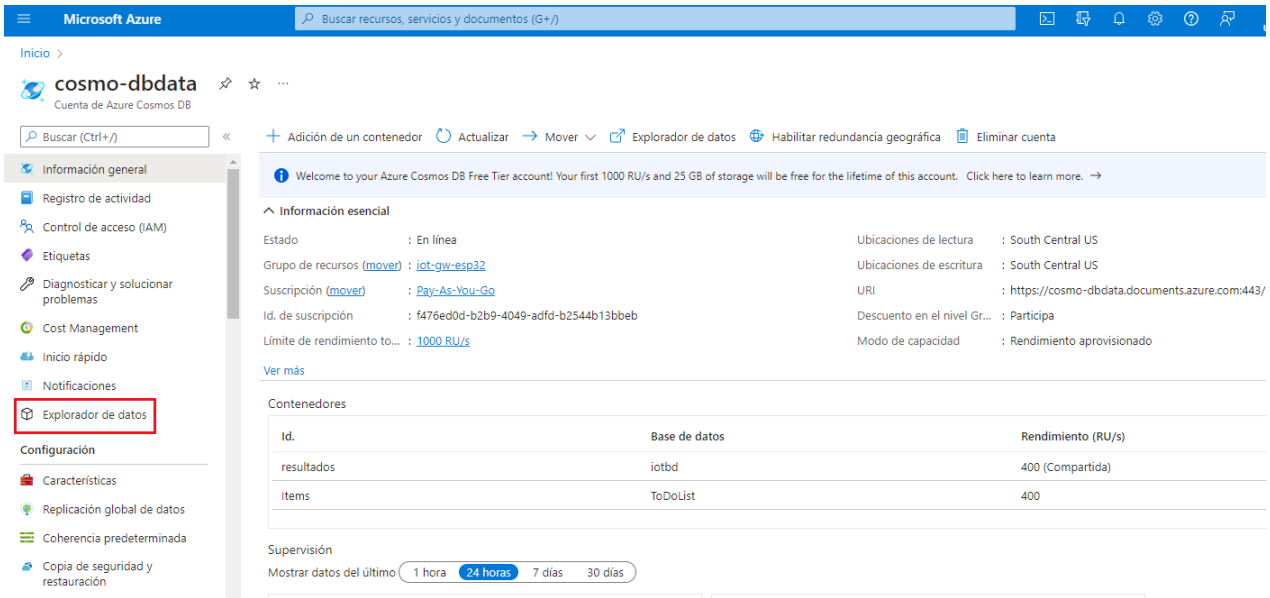


Figura 5-91.: Explorador de datos Cosmos DB.

2. Procedemos a crear un nuevo Contenedor seleccionamos **New Container**, y procedemos a crear una nueva base de datos que va a almacenar los datos.
3. Llenamos todos los campos con los datos correspondientes como se observa en la fig. 5-92.

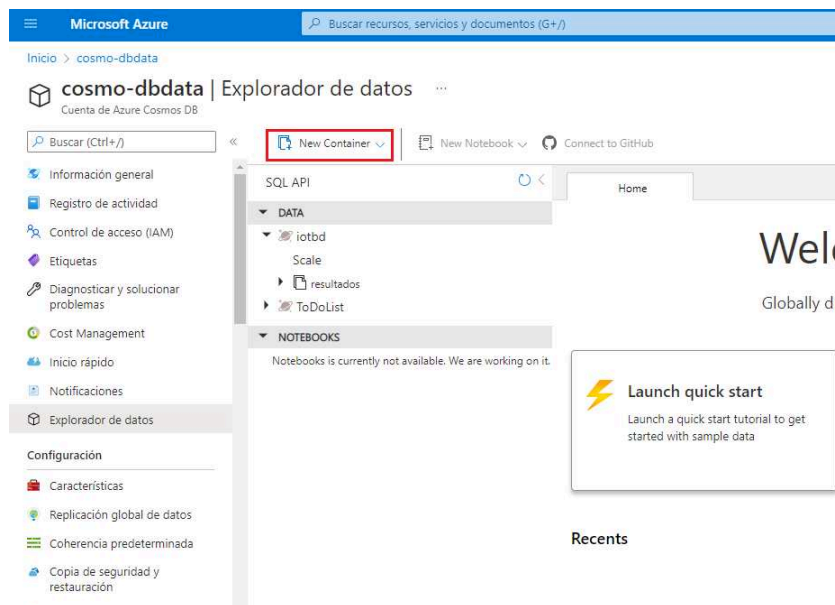


Figura 5-92.: Crear container en Cosmos DB.

4. Llenamos el formulario correspondiente fig. 5-93.

**New Container**

\* Database id  Create new  Use existing

Share throughput across containers

\* Database throughput (400 - unlimited RU/s)  
 Autoscale  Manual  
 Estimate your required RU/s with [capacity calculator](#).

Estimated cost (USD): **\$0.016 hourly / \$0.38 daily / \$11.68 monthly** (1 region, 200RU/s, \$0.00008/RU)

\* Container id

\* Indexing  
 Automatic  Off  
 All properties in your documents will be indexed by default for flexible and efficient queries. [Learn more](#)

\* Partition key

Unique keys

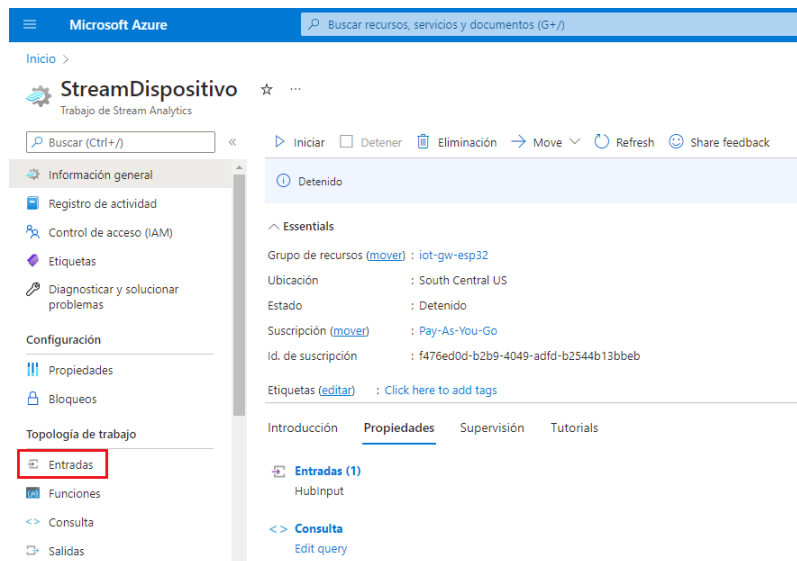
+ Add unique key

Analytical store  
 On  Off  
 > Advanced

**OK**

**Figura 5-93.:** Formulario para crear un container en Cosmos DB.

5. Regresamos a nuestro grupo de recursos y procedemos a seleccionar el servicio de **StreamDispositivo**, seleccionamos la opción de **Entradas** fig. 5-94 del menú correspondiente.



**Figura 5-94.:** Configurar entrada de Stream Analytics.

6. Debemos agregar una entrada para captar el flujo de datos que están ingresando en el recurso de IoT Hub para lo cual en Agregar entrada seleccionamos **Centro de IoT** como se puede observar en la fig. 5-95.



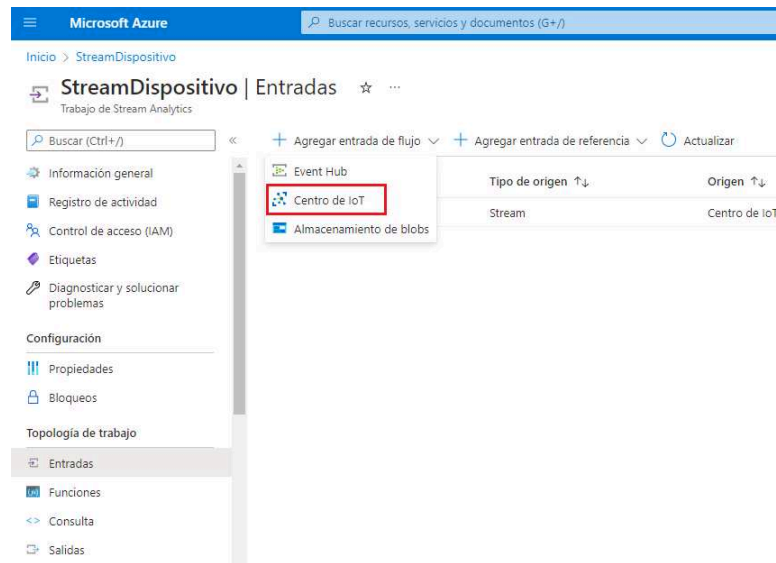


Figura 5-95.: Crear una entrada de IoT.

7. Se nos despliega el siguiente formulario fig. 5-96 procedemos a llenar los campos siguientes:

**Alias de entrada:** Asignamos un nombre cualquiera y que este disponible.

**Centro de IoT:** Seleccionamos nuestro servicio creado anteriormente.

**Nombre de directiva de acceso compartido:** Seleccionamos la opción que corresponde.

**Formato de serialización de eventos:** El formato es de tipo JSON para el envío de datos y luego en crear.

**Detalles de entrada**

HubInput

Probar Eliminar Open Centro de IoT

Alias de entrada  
HubInput

Proporcionar configuración de Centro de IoT de forma manual  
 Seleccionar Centro de IoT de las suscripciones

Suscripción  
Pay-As-You-Go (f476ed0d-b2b9-4049-adf6-b2544b13b...

Centro de IoT \*  
iothub-azure-esp32

Grupo de consumidores \*  
\$Default

Nombre de directiva de acceso compartido \*  
iothubowner

Clave de directiva de acceso compartido  
\*\*\*\*\*

Punto de conexión  
Mensajes

Clave de partición

Formato de serialización de eventos \*  
JSON

Codificación  
UTF-8

Guardar

Figura 5-96.: Crear Input centro de IoT.

8. Continuamos y seleccionamos la opción de Salidas del menú y la opción de **Agregar** seleccionamos los siguientes servicios: Almacenamiento de blobs, Cosmos DB y Power BI fig. 5-97, cada servicio tiene su propia configuración que se detalla a continuación:

Microsoft Azure

Inicio > StreamDispositivo

StreamDispositivo | Salidas

Trabajo de Stream Analytics

Buscar (Ctrl+/) Actualizar

Almacenamiento de blobs

Almacenamiento de tablas

Azure Data Explorer (versión preliminar)

Azure PostgreSQL (versión preliminar)

Azure Synapse Analytics

Cola del bus de servicio

Cosmos DB

Data Lake Storage Gen1

Event Hub

Función de Azure

Power BI

SQL Database

Tema del bus de servicio

Receptor	Modo de autenticación
Almacenamiento de blobs	Identidad administrada
Azure Data Explorer (versión preliminar)	Identidad administrada
Cosmos DB	Cadena de conexión
Power BI	Token de usuario

Figura 5-97.: Servicios de Salidas en Stream Analytics.

9. Se debe configurar la creación de Salidas del Stream Analytics para el servicio de Blob Storage, seleccionamos cuenta de almacenamiento, contenedor y seleccionamos el tipo

de formato por CSV separado por comas para que los datos se guarden en ese formato tal como se muestra en la fig. 5-98.

The screenshot shows the 'Detalles de salida' (Output Details) configuration window for Blob Storage. The window is titled 'Detalles de salida' and has a close button (X) in the top right corner. Below the title, there are three buttons: 'Probar' (Test), 'Eliminar' (Delete), and 'Open Blob Storage/ADLS Gen2'. The 'Alias de salida' (Output Alias) field is set to 'blobOutput'. Below this, there are two radio buttons: 'Proporcionar configuración de Blob Storage/ADLS Gen2 de forma manual' (Provide Blob Storage/ADLS Gen2 configuration manually) and 'Seleccionar Blob Storage/ADLS Gen2 de las suscripciones' (Select Blob Storage/ADLS Gen2 from subscriptions). The 'Suscripción' (Subscription) dropdown is set to 'Ray-As-You-Go (6476e00b-b2b9-4049-adf0-b2544b13b...)'. The 'Cuenta de almacenamiento' (Storage Account) dropdown is set to 'almacenamientobmp280'. Below this, there are two radio buttons: 'Crear' (Create) and 'Usar existente' (Use existing). The 'Contenedor' (Container) dropdown is set to 'output'. The 'Modo de autenticación' (Authentication Mode) dropdown is set to 'Identidad administrada: Asignado por el sistema' (Managed Identity: Assigned by the system). Below this, there is a note: 'La identidad administrada tiene permisos para el Blob Storage/ADLS Gen2 seleccionado.' (The managed identity has permissions for the selected Blob Storage/ADLS Gen2). The 'Patrón de ruta de acceso' (Access Path Pattern) dropdown is set to '{date}'. The 'Formato de fecha' (Date Format) dropdown is set to 'YYYY/MM/DD'. The 'Formato de hora' (Time Format) dropdown is set to 'HH'. At the bottom of the window, there is a 'Guardar' (Save) button highlighted with a red box.

Figura 5-98.: Salidas del Stream Analytics Blob Storage.

10. Luego se debe configurar las Salidas del Stream Analytics para el servicio de Cosmos DB, le damos un nombre de Salidas, id de la cuenta, nombre de la base de datos y el nombre del contenedor donde se va almacenar los datos tal como se visualiza en la fig. 5-99.

**Detalles de salida** ×

OutputCosmo

Probar Eliminar Open Cosmos DB

Alias de salida  
OutputCosmo

Proporcionar configuración de Cosmos DB de forma manual  
 Seleccionar Cosmos DB de las suscripciones

Suscripción  
Pay-As-You-Go (f476ed0d-b2b9-4049-adfd-b2544b13b...)

Id. de cuenta \*  
cosmo-database

Base de datos \*  
 Crear  Usar existente  
lotbd

Nombre de contenedor \*  
 Crear  Usar existente  
resultados

Modo de autenticación  
Cadena de conexión

Clave de cuenta  
.....

Id. de documento

Guardar

**Figura 5-99.:** Salidas del Stream Analytics s DB.

- Configuramos la creación de Salida del Stream Analytics para el servicio de Power BI, le damos un nombre de Salidas, base de almacenamiento y la tabla como se puede ver en la fig. 5-100.

**Detalles de salida** ×

PowerBIOutput

Probar Eliminar

Alias de salida  
PowerBIOutput

Proporcionar configuración de Área de trabajo de grupo de forma manual  
 Seleccionar Área de trabajo de grupo de las suscripciones

Área de trabajo de grupo \*  
Mi área de trabajo

Modo de autenticación  
Token de usuario

⚠ En el caso de que el conjunto de datos o la tabla ya existan en la suscripción de Microsoft Power BI, se sobrescribirán.

Nombre del conjunto de datos \*  
est32

Nombre de tabla \*  
bmp280

Autorizado actualmente como Est. Ricardo Marcelo Chuqui Ortega (rchuquio@est.ups.edu.ec)

**Autorización**  
Haga clic en el botón que hay a continuación si quiere renovar la autorización, autorizar con otra cuenta o modificar el área de trabajo.

Renovar autorización

Guardar

**Figura 5-100.:** Salidas del Stream Analytics Power BI.

- Creamos una consulta SQL para mandar a almacenar los datos en nuestro servicio implementado de Salidas en la base de datos de Cosmos DB fig. 5-101.

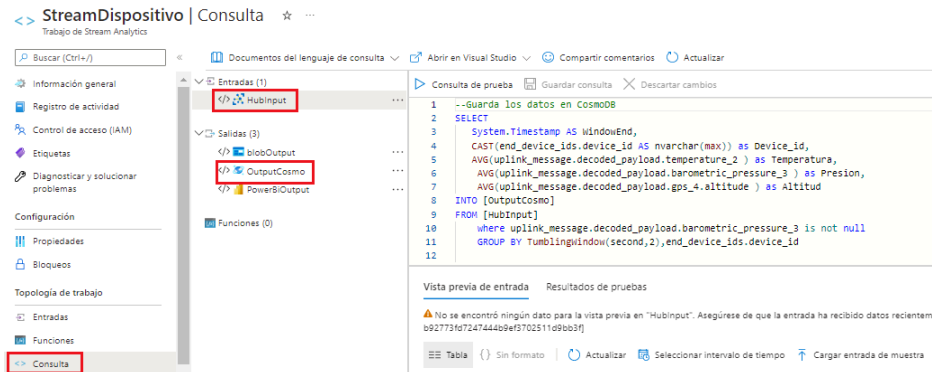


Figura 5-101.: Consulta SQL de Salidas Cosmos DB.

13. Creamos una consulta SQL para mandar a guardar los datos en nuestro servicio implementado de Salidas Blob Storage fig. 5-102.

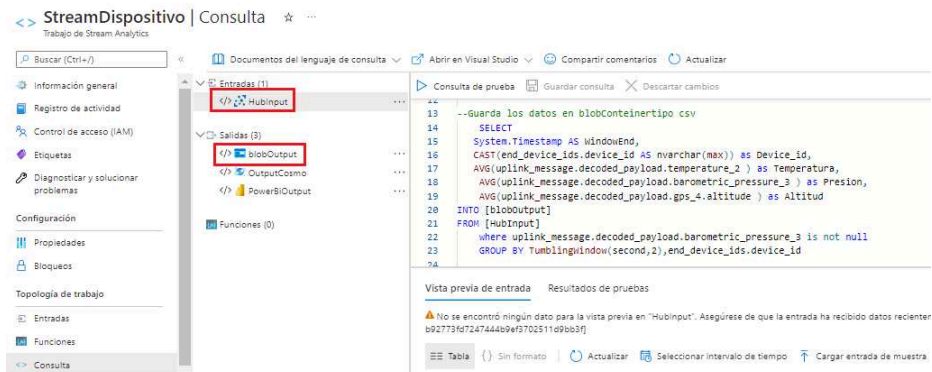
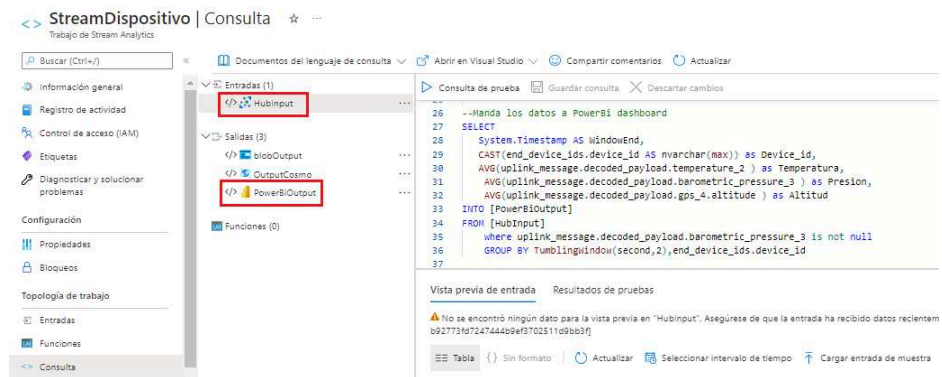


Figura 5-102.: Consulta SQL de Salidas Blob Storage.

graficar

14. Creamos una consulta SQL para mandar a visualizar y graficar los datos en nuestro servicio implementado de Salidas Power BI fig. 5-103.



The screenshot displays the Microsoft Stream Analytics console interface. On the left, the navigation pane shows the 'Consultas' (Queries) section. Under 'Salidas' (Outputs), three outputs are listed: 'HubInput', 'blobOutput', and 'PowerBIOutput', with 'PowerBIOutput' highlighted. The main area shows a SQL query in the 'Consulta de prueba' (Test Query) editor. The query is as follows:

```
26 --Manda los datos a PowerBI dashboard
27 SELECT
28     System.Timestamp AS WindowEnd,
29     CAST(end_device_ids.device_id AS nvarchar(max)) AS Device_id,
30     AVG(uplink_message.decoded_payload.temperature_2 ) AS Temperatura,
31     AVG(uplink_message.decoded_payload.barometric_pressure_3 ) AS Presion,
32     AVG(uplink_message.decoded_payload.gps_4.altitude ) AS Altitud
33 INTO [PowerBIOutput]
34 FROM [HubInput]
35 WHERE uplink_message.decoded_payload.barometric_pressure_3 is not null
36 GROUP BY TumblingWindow(second,2),end_device_ids.device_id
37
```

Below the query editor, there is a 'Vista previa de entrada' (Input Preview) section. It contains a warning message: 'No se encontró ningún dato para la vista previa en "HubInput". Asegúrese de que la entrada ha recibido datos recientemente b92773f97247444b9f5702511d9b63f'. At the bottom, there are controls for 'Tabla' (Table), 'Actualizar' (Refresh), 'Seleccionar intervalo de tiempo' (Select time interval), and 'Cargar entrada de muestra' (Load sample input).

Figura 5-103.: Consulta SQL de Salidas Power BI.

# 6. RESULTADOS

## 6.1. Pruebas y resultados con Azure IoT Hub

A continuación, se presenta las diferentes pruebas y resultados una vez finalizado todas las respectivas configuraciones. Implementamos un diagrama de conexión el cual se ilustra en la fig. 6-1, se puede visualizar la conexión de los pines del dispositivo HelTec ESP32 y el sensor de presión BMP280.

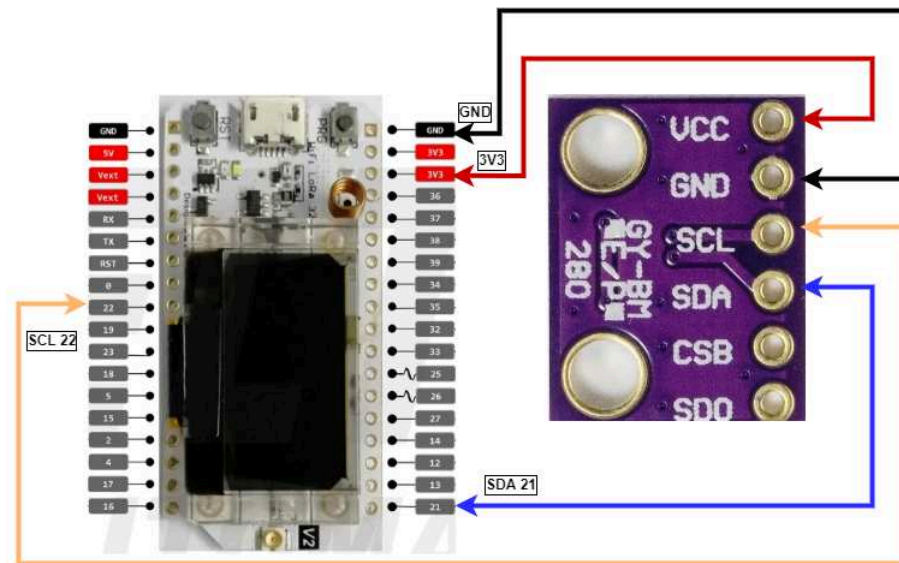
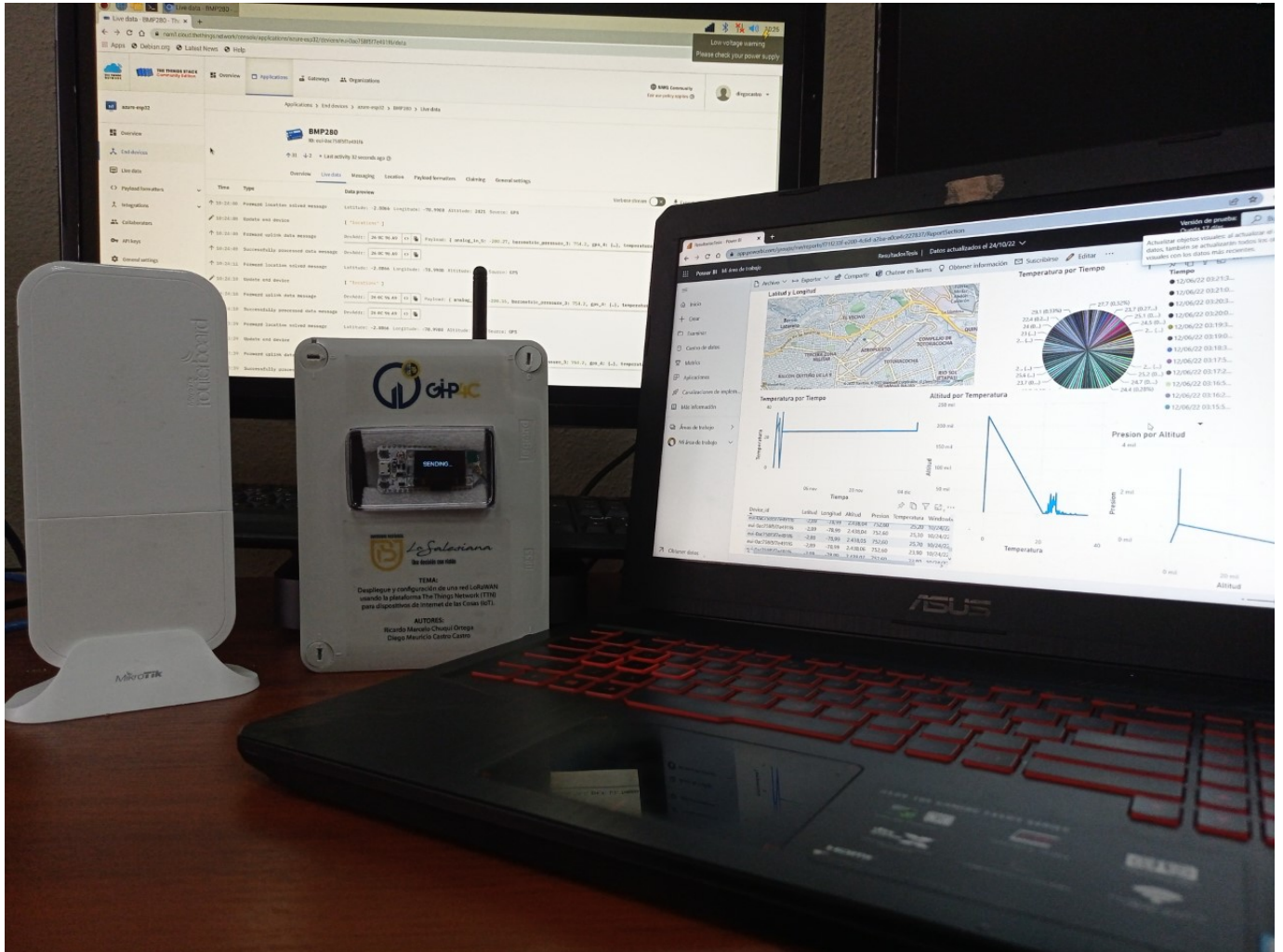


Figura 6-1.: Diagrama de conexión HelTec ESP32 y el Sensor BMP280.

BMP280	ESP32
V <sub>in</sub>	3.3v
GND	GND
SCL	Pin 22
SDA	Pin 21

Tabla 6-1.: Pines de Conexión.

1. En la fig. 6-2 se muestra el dispositivo GW y el dispositivo final (HelTec ESP32 - Sensor BMP280) los cuales deben estar conectados y configurados correctamente para proceder a realizar las diferentes pruebas.



**Figura 6-2.:** GW y HelTec ESP32 - Sensor BMP280.

2. Se debe verificar que el sensor implementado esté conectado y enviando datos a la plataforma de TTN como se visualiza en la fig. 6-3, para realizar diversas pruebas de funcionamiento.



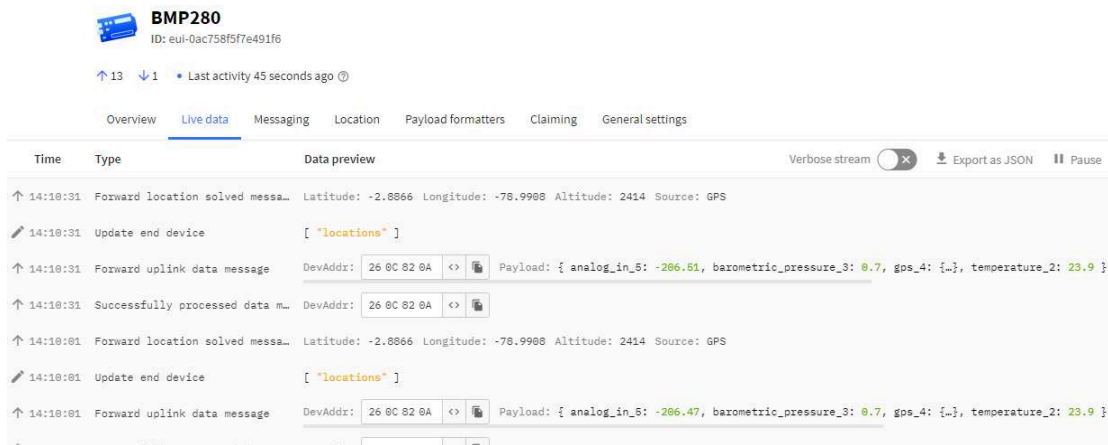


Figura 6-3.: Dispositivo conectado al TTN.

- Se procede a realizar una **Consulta sql** como se observa en la fig. 6-4, para poder visualizar los datos que envía el dispositivo ESP32 con el sensor (BMP280) al servidor de aplicaciones de Azure IoT Hub y poder filtrar únicamente los datos que sean necesarios. Creada la consulta se debe de ejecutar una prueba, luego se selecciona **guardar consulta** con esto ya se puede filtrar y almacenar estos datos para presentar en tablas, documentos, gráficos, etc.

```

1 --Guarda los datos en CosmosDB
2 SELECT
3   System.Timestamp AS WindowEnd,
4   CAST(end_device_ids.device_id AS nvarchar(max)) as Device_id,
5   AVG(uplink_message.decoded_payload.temperature_2 ) as Temperatura,
6   AVG(uplink_message.decoded_payload.barometric_pressure_3 ) as Presion,
7   AVG(uplink_message.decoded_payload.gps_4.latitude ) as Latitud,
8   AVG(uplink_message.decoded_payload.gps_4.longitude ) as Longitud,
9   AVG(uplink_message.decoded_payload.gps_4.altitude ) as Altitud
10 INTO [SalidaCosmosDB]
11 FROM [EntradaStream]
12   where uplink_message.decoded_payload.barometric_pressure_3 is not null
13   GROUP BY TumblingWindow(second,2),end_device_ids.device_id
14
15 --Guarda los datos en blobConteintipo csv
16 SELECT

```

Vista previa de entrada **Resultados de pruebas**

Mostrando 1 filas de "SalidaBlob". [Descargar resultados](#)

WindowEnd datetime	Device_id string	Temperatura float	Presion float	Latitud float	Longitud float	Altitud float
"2022-12-06T15:48:04.000000..."	"eui-0ac758f5f7e491f6"	27.145833333333333	754.3416666666667	-2.8866000000000001	-78.990800000000002	2420.0670833333333

Figura 6-4.: Consulta SQL para filtrar los datos.

- Luego de haber creado la consulta de filtrado se debe de inicializar el servicio de Stream Analytics para este paso tenemos que dirigirnos a **información general** luego a **iniciar** fig. 6-5 y el servicio comenzara a ejecutarse.

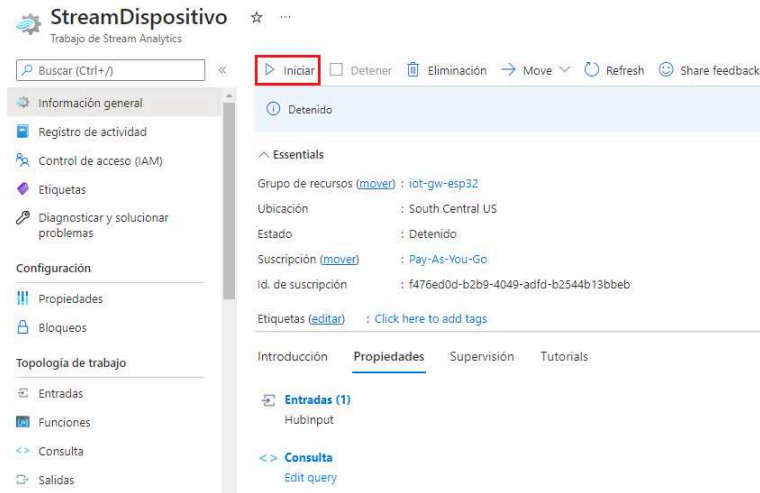


Figura 6-5.: Ejecutar servicio Stream Analytics.

- Esperamos unos minutos hasta que se ejecute correctamente el servicio de Stream Analytics, si no hay ningún error se mostrará un mensaje de ejecución correcta como se observa en la fig. 6-6, o también podemos verificar cualquier error desde el menú de notificaciones del menú superior derecho.

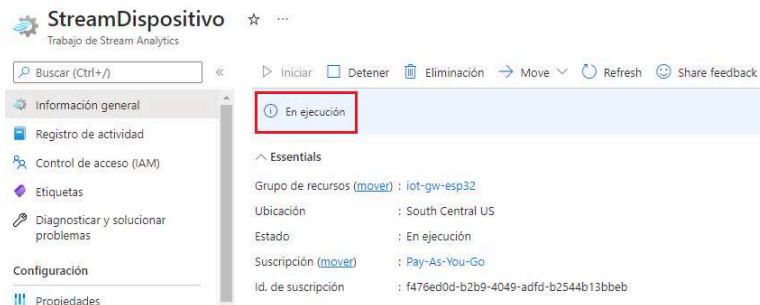


Figura 6-6.: Ejecución del servicio Stream Analytics.

- Desde el grupo de recursos **iot-gw-esp32** creado anteriormente, se debe seleccionar el servicio de **Cuenta de Almacenamiento** que en este servicio se le asigno con el nombre de **almacenamientobmp280**, posteriormente, en el menú izquierdo seleccionamos la opción de **Contenedores** y abrimos el contenedor creado anteriormente que se le nombro como **output**. En la fig. 6-7 se visualiza una carpeta raíz con todas las sub-carpetas creadas por orden de mes y fecha, buscamos y seleccionamos la más actual y se puede visualizar un archivo de tipo csv con toda la data generada por el dispositivo.

Seleccionar el archivo csv, buscamos en el menú superior la opción de **Editar** y seleccionamos la **Vista previa** para una mejor visualización de los datos. En la tabla

generada se ve la información filtrada por la consulta SQL.

WindowEnd	Device_id	Temperatura	Presion	Latitud	Longitud	Altitud
2022-12-06T15%3A09%3A08.000000Z	eui-0ac758f57e491f6	24.3	754.4	-2.8866	-78.9908	2419.21
2022-12-06T15%3A09%3A42.000000Z	eui-0ac758f57e491f6	26	754.4	-2.8866	-78.9908	2419.02
2022-12-06T15%3A11%3A44.000000Z	eui-0ac758f57e491f6	25.8	754.4	-2.8866	-78.9908	2419.26
2022-12-06T15%3A11%3A44.000000Z	eui-0ac758f57e491f6	25.7	754.4	-2.8866	-78.9908	2419.59
2022-12-06T15%3A12%3A16.000000Z	eui-0ac758f57e491f6	25.6	754.4	-2.8866	-78.9908	2419.98
2022-12-06T15%3A12%3A48.000000Z	eui-0ac758f57e491f6	25.6	754.3	-2.8866	-78.9908	2420.76
2022-12-06T15%3A13%3A18.000000Z	eui-0ac758f57e491f6	25.5	754.3	-2.8866	-78.9908	2420.28

Figura 6-7.: Tabla de datos en archivo csv.

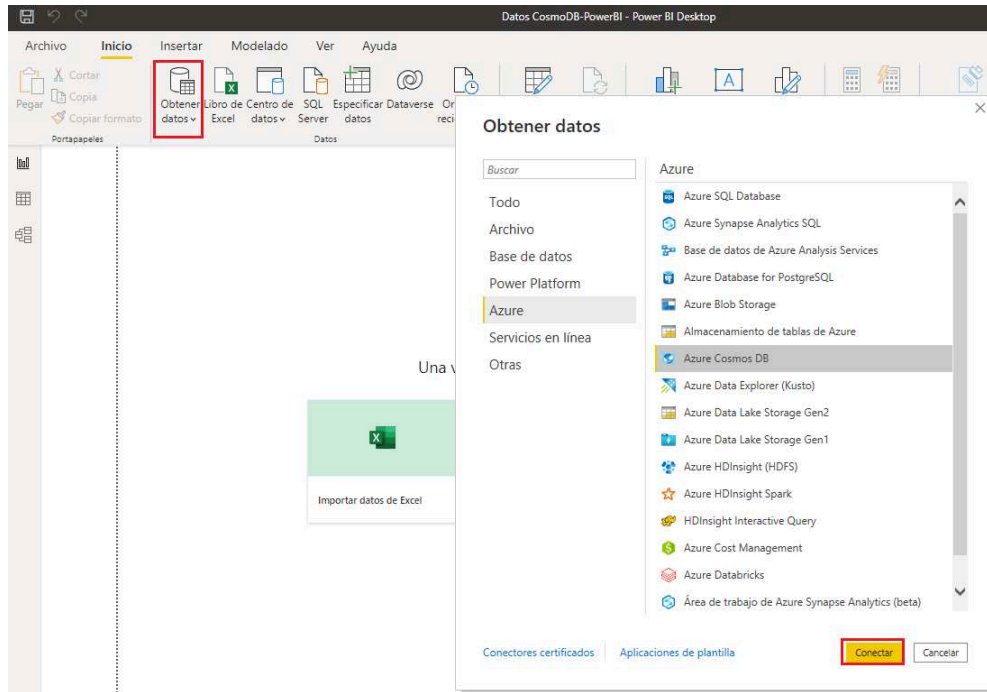
- En este punto vamos a verificar la información guardada en una base de datos desde el servicio de Cosmos DB, por tanto, desde nuestro grupo de recursos seleccionamos el servicio mencionado y desde el menú de la parte izquierda seleccionamos **Explorador de datos**, aquí se listara la base creada, desplegamos y buscamos nuestra tabla, se mostrara la data enviada desde el dispositivo fig. 6-8, debemos actualizar la base en caso de ser necesario.

id	/WindowEnd
replace_with_new_document...	2022-07-19T19:47:16.000000...
f01df9d9-735f-4611-8623-6...	2022-07-19T20:50:20.000000...
d3ae820b-c6f1-4f90-be0f-6c...	2022-07-19T20:50:50.000000...
ab226a59-1e4a-47e7-9a37-...	2022-07-19T20:51:20.000000...
723dcb6c-2f0f-4427-9448-8...	2022-07-19T20:51:52.000000...
2004559d-eded-4c15-84d1-...	2022-07-19T20:52:22.000000...
14b9c267-7b63-48c2-ba83-f...	2022-07-19T20:52:54.000000...
82525b78-b130-46f6-8d26-...	2022-07-19T20:53:26.000000...
e53f358c-6080-40b5-b500-b...	2022-07-19T20:53:58.000000...
206fd9ca-8298-483f-8c6a-a...	2022-07-19T20:54:30.000000...
6ac94247-bada-4e3c-855b-5...	2022-07-19T20:55:00.000000...
392f5f3d-57e9-4709-b281-b...	2022-07-19T20:55:32.000000...

Figura 6-8.: Base de datos de Cosmos DB.

- Para generar gráficos con la información guardada en nuestra base de datos utilizaremos el servicio de **Power BI**, se recomienda usar la aplicación de escritorio. Para construir nuestros gráficos tenemos que importar la información, por lo que, desde el

menú superior seleccionamos **Obtener datos**, de la lista desplegada seleccionamos **Azure** y seleccionamos el servicio de **Cosmos DB** y finalmente **Conectar** tal como se ilustra en la fig. 6-9.



**Figura 6-9.:** Conectar los servicios de Cosmos DB y Power BI.

9. Siguiendo con la configuración de Power BI, desde nuestro servicio de Cosmos DB copiamos la **URI** y la **PRIMARY KEY** como se muestra en la fig. 6-10 de la configuración del servicio Cosmos DB. Desde la pestaña de **Home** seleccionamos la opción de **Connect** y podemos visualizar la información que necesitamos.

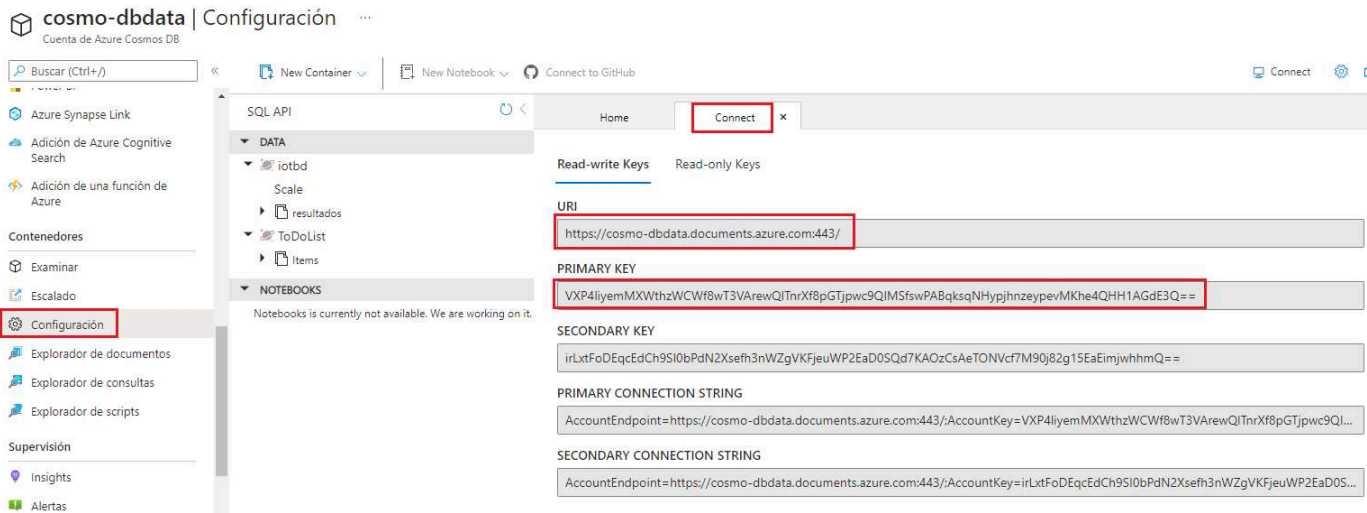


Figura 6-10.: Datos de conexión del servicio de CosmosDB.

- Se debe copiar los datos anteriores y pegar en los campos correspondientes para conectar el servicio Cosmos DB y Power BI, además de que se debe especificar el nombre de la base creada en Cosmos DB como se ilustra en la fig. 6-11, cuando se conecta por primera vez también pide ingresar la clave PRIMARY KEY.



Figura 6-11.: Conexión de CosmosDB y PowerBI.

- Si la conexión es correcta se desplegará una tabla la misma que fue generada en Cosmos DB, en nuestro caso la tabla se llama resultados, dicha tabla debe ser formateada para obtener solo los campos requeridos, para esto se procede a dar click en **Transformar datos** como se observa en la fig. 6-12.

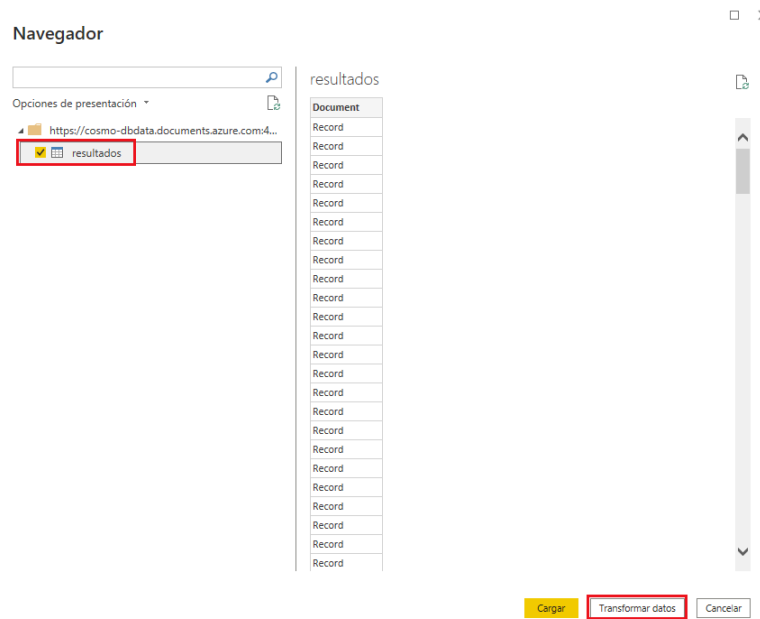


Figura 6-12.: Transformar datos en Power BI.

12. En este paso se desplegará un sub-menú fig. 6-13 y se debe seleccionar las columnas que deseamos visualizar le damos a **Aceptar** y se genera una tabla en Power BI con los datos enviados desde Cosmos DB.

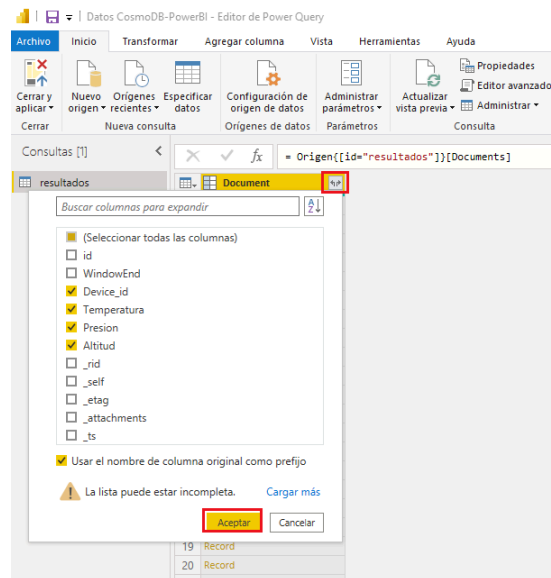


Figura 6-13.: Filtrado de datos específicos en Power BI.

13. Se genera la siguiente tabla fig. 6-14, con los datos de esta tabla ya se puede crear gráficas para su visualización y análisis de resultados.

Document.Id	Document.WindowEnd	Document.Device_id	Document.Temperatura	Document.Presion	Document.Altitud	Document.Latitud	Document.Longitud
1d19f328-d1c0-43da-ace1-3d90e5f7dec8	2022-08-02T20:50:52.0000000Z	eu1-0ac758f5f7e491f6	25,4	751,6	2448,99	-2,8866	-78,9908
eb5e6c98-bd90-49ff-8d38-e3178b20e36c	2022-08-02T20:51:22.0000000Z	eu1-0ac758f5f7e491f6	25,5	751,6	2449,13	-2,8866	-78,9908
c3b8745c-2c04-4537-bfa6-92a7cc01c21e	2022-08-02T20:51:52.0000000Z	eu1-0ac758f5f7e491f6	25,5	751,6	2449,35	-2,8866	-78,9908
d4c4b1f6-7f85-410e-819e-7830592M975	2022-08-02T20:52:24.0000000Z	eu1-0ac758f5f7e491f6	25,5	751,6	2448,81	-2,8866	-78,9908
c9ecfd4f-d52a-49ac-a994-f3774327777a	2022-08-02T20:52:54.0000000Z	eu1-0ac758f5f7e491f6	25,5	751,6	2449,02	-2,8866	-78,9908
d7f483cb-38e2-42ce-9488-e70422385492	2022-08-02T20:53:26.0000000Z	eu1-0ac758f5f7e491f6	25,5	751,6	2449,08	-2,8866	-78,9908
d07f6524-fd9e-47f9-96f4-631e56d13a58	2022-08-02T20:53:56.0000000Z	eu1-0ac758f5f7e491f6	27,2	751,6	2448,79	-2,8866	-78,9908
05b45256-f117-4821-b7a6-4825dbb74c0a	2022-08-02T20:54:28.0000000Z	eu1-0ac758f5f7e491f6	27,5	751,6	2448,97	-2,8866	-78,9908
21dd54aa-7164-498b-a8e0-8dd9b261e89f	2022-08-02T20:55:00.0000000Z	eu1-0ac758f5f7e491f6	28,4	754,3	2420,61	-2,8866	-78,9908
4a814d51-a685-4323-9fa2-6ed0b01e4d240	2022-08-02T20:55:30.0000000Z	eu1-0ac758f5f7e491f6	31,7	751,6	2448,85	-2,8866	-78,9908
1690dc35-c2bd-4ef6-b0a4-bc9ca1661883	2022-08-02T20:56:00.0000000Z	eu1-0ac758f5f7e491f6	28,4	751,6	2449,18	-2,8866	-78,9908
2bba9139-6615-4ef2-8662-af0c66e817e0	2022-08-02T20:56:30.0000000Z	eu1-0ac758f5f7e491f6	26,3	751,6	2449,54	-2,8866	-78,9908
6cd48aee-04fb-49cd-ad3f-8769ee460fbf	2022-08-02T20:57:02.0000000Z	eu1-0ac758f5f7e491f6	25,6	751,6	2449,54	-2,8866	-78,9908
64150a93-4fe2-438e-8c66-3a12a62e6838	2022-08-02T20:58:02.0000000Z	eu1-0ac758f5f7e491f6	23,9	751,6	2449,65	-2,8866	-78,9908
360bf28-2f84-4718-abea-a513536ebba9	2022-08-02T20:58:34.0000000Z	eu1-0ac758f5f7e491f6	24,1	751,6	2449,3	-2,8866	-78,9908
cf6ae1d1-98e9-4661-a85c-3cb7ee11493f	2022-08-02T20:59:06.0000000Z	eu1-0ac758f5f7e491f6	24,2	751,5	2449,97	-2,8866	-78,9908
11052a24-bee1-466a-a0fa-9ae39e8c0774	2022-08-02T20:59:36.0000000Z	eu1-0ac758f5f7e491f6	24,2	751,5	2450,05	-2,8866	-78,9908
805a1382-4f54-4cfa-a091-6e5188e80fcd	2022-08-02T21:00:06.0000000Z	eu1-0ac758f5f7e491f6	23,9	751,5	2450,4	-2,8866	-78,9908
6c9ce1899-630f-43ae-b248-61e0d031f494	2022-08-02T21:00:38.0000000Z	eu1-0ac758f5f7e491f6	23,8	751,5	2449,9	-2,8866	-78,9908
1bebc022-0278-42b7-9661-02c82de509eb	2022-08-02T21:01:10.0000000Z	eu1-0ac758f5f7e491f6	24,1	751,5	2449,68	-2,8866	-78,9908
10c4187b-3e5d-4942-a9dc-e2a7b05900a6	2022-08-02T21:01:40.0000000Z	eu1-0ac758f5f7e491f6	24,3	751,5	2450,09	-2,8866	-78,9908
b0027fd-d0f0e-4f6e-99f4-f3b1cc66e1ed	2022-08-02T21:02:12.0000000Z	eu1-0ac758f5f7e491f6	24,5	751,5	2450,19	-2,8866	-78,9908
d3b94bb-ae7e-404d-42c8-d1932f2c063d	2022-08-02T21:02:44.0000000Z	eu1-0ac758f5f7e491f6	24,6	751,5	2450,6	-2,8866	-78,9908
eb248605-9d1e-44a1-93a8-361d58a30d7	2022-08-02T21:03:16.0000000Z	eu1-0ac758f5f7e491f6	24,7	751,5	2450,22	-2,8866	-78,9908
f6e10743-ed5f-482a-99c2-1cc82976d4b2	2022-08-02T21:03:46.0000000Z	eu1-0ac758f5f7e491f6	24,8	751,5	2450,23	-2,8866	-78,9908
8f52198f-ae41-401f-903d-55db89e7a226	2022-08-02T21:04:18.0000000Z	eu1-0ac758f5f7e491f6	24,8	751,5	2450,01	-2,8866	-78,9908
f5574bf8-d906-4310-b6b6-6ea272042c29	2022-08-02T21:05:22.0000000Z	eu1-0ac758f5f7e491f6	25	751,5	2450,39	-2,8866	-78,9908
f3d9af34-99a8-4903-b117-67a5faa9695d	2022-08-02T21:05:54.0000000Z	eu1-0ac758f5f7e491f6	25	751,5	2450,31	-2,8866	-78,9908
cc3424e1-e9b7-4b98-9b5f-0c7c02804d8	2022-08-02T21:06:24.0000000Z	eu1-0ac758f5f7e491f6	25	751,5	2450,08	-2,8866	-78,9908
7c0f74be-46b8-4bb7-8205-a09ec7f55d7	2022-08-02T21:06:56.0000000Z	eu1-0ac758f5f7e491f6	25	751,5	2450,28	-2,8866	-78,9908

Figura 6-14.: Tabla de datos en PowerBI desde CosmosDB.

14. La fig. 6-15 corresponde al cambio de temperatura en centígrados con relación al tiempo. Se observa que la temperatura cambia en medida que pasa el tiempo en minutos.

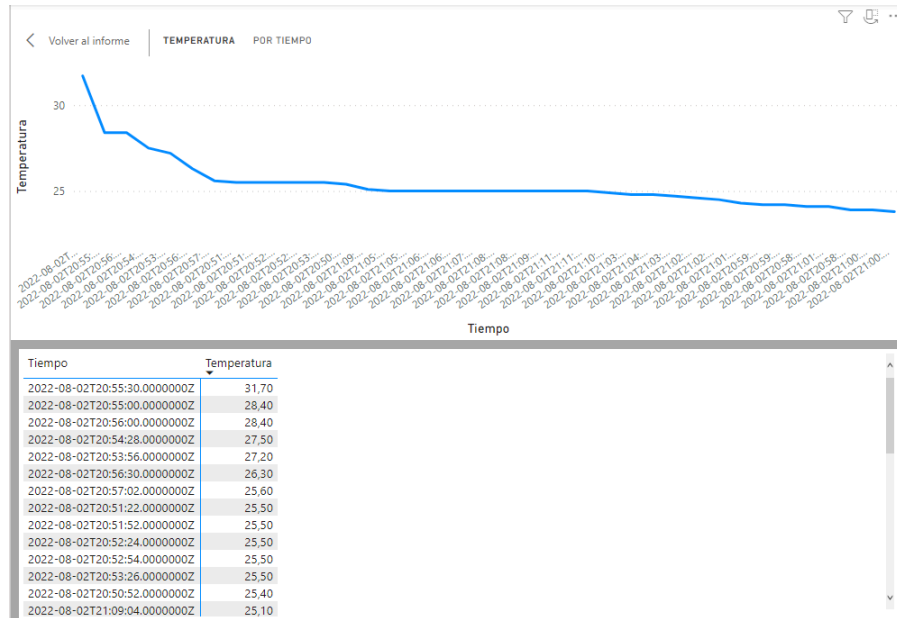


Figura 6-15.: Gráfica Temperatura por Tiempo.

15. La siguiente fig. 6-16 corresponde al cambio de temperatura con relación al tiempo en visualización de Pastel. Se observa que la temperatura cambia en medida que pasa el



tiempo en minutos con su respectivo porcentaje que va cambiando.

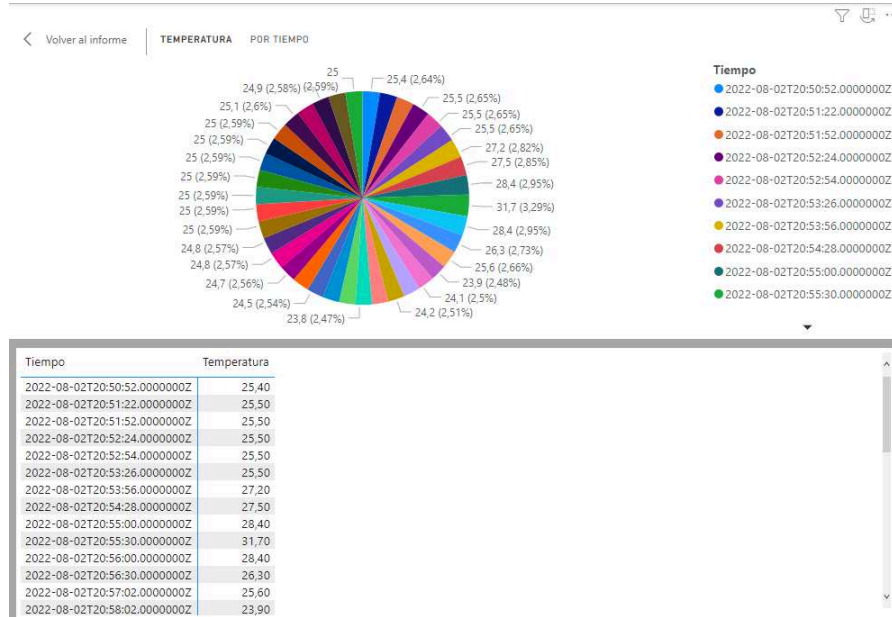


Figura 6-16.: Gráfica de Pastel Temperatura por Tiempo.

16. La fig. 6-17 corresponde al cambio de temperatura en centígrados con relación a la altitud en metros. Se observa que la temperatura cambia según la altitud en la que se encuentra el dispositivo.

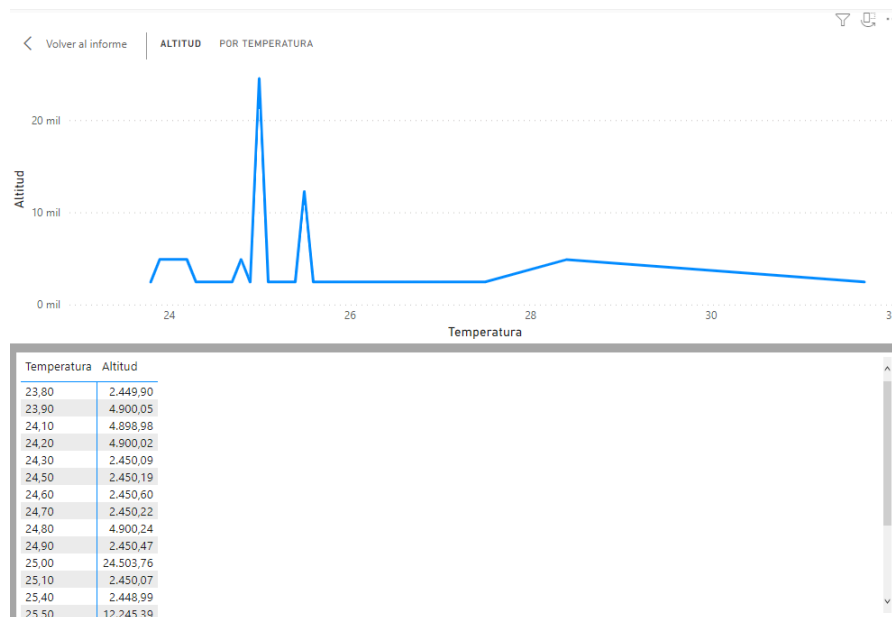
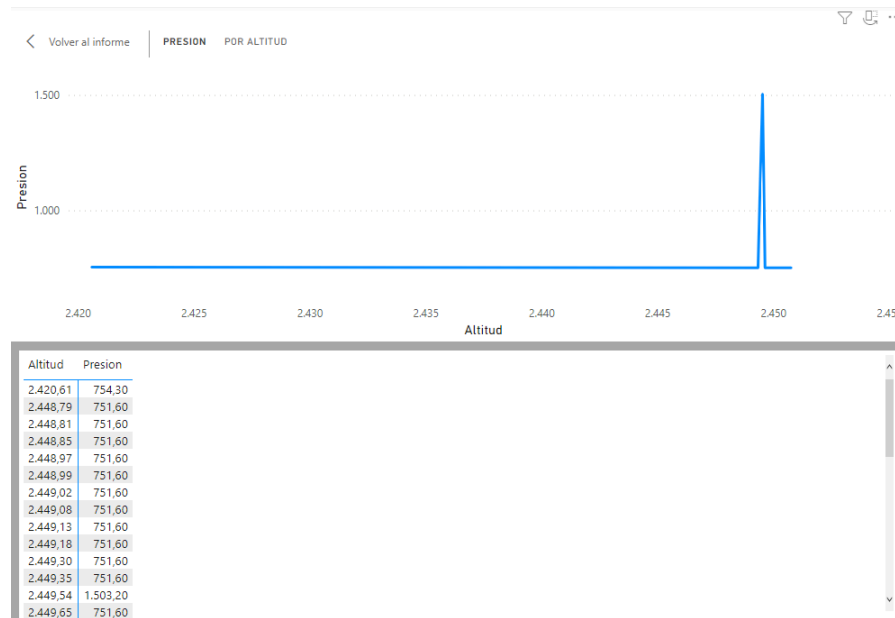


Figura 6-17.: Gráfica de la Temperatura por Altitud.

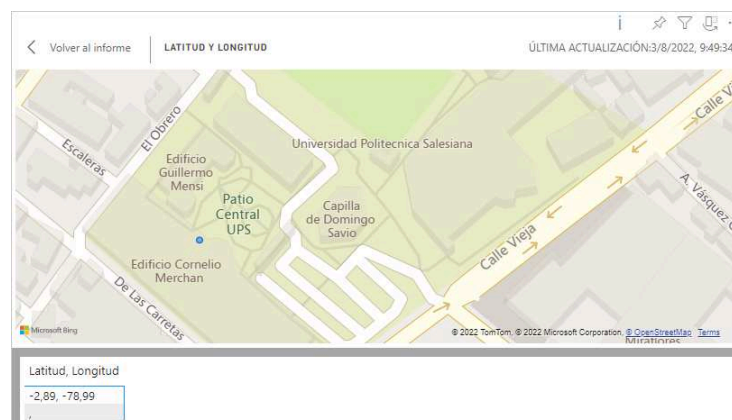


17. La fig. **6-18** corresponde al cambio de Presión Atmosférica en hectopascal con relación a la altitud en metros. La presión atmosférica disminuye al aumentar la altura y aumenta si se baja la altitud.



**Figura 6-18.:** Gráfica Presión por Altitud.

18. La fig. **6-19** corresponde a la ubicación de nuestro dispositivo HelTec ESP32 ayudándonos con los datos de la latitud y longitud. Se observa un punto azul que corresponde a la ubicación de nuestro dispositivo que se encuentra en la Universidad Politécnica Salesiana.



**Figura 6-19.:** Ubicación geográfica ESP32.

## 6.2. Cayenne LPP

Es un aplicación de TTN que permite visualizar, crear e interpretar los datos en gráficos, recolecta la información generada por los dispositivos IoT en tiempo real. Su configuración se detalla a continuación.

1. Desde la página web de TTN en el apartado de **Applications** nos dirigimos a **Integrations**, agregaremos un nuevo Webhook observe la fig. 6-20.

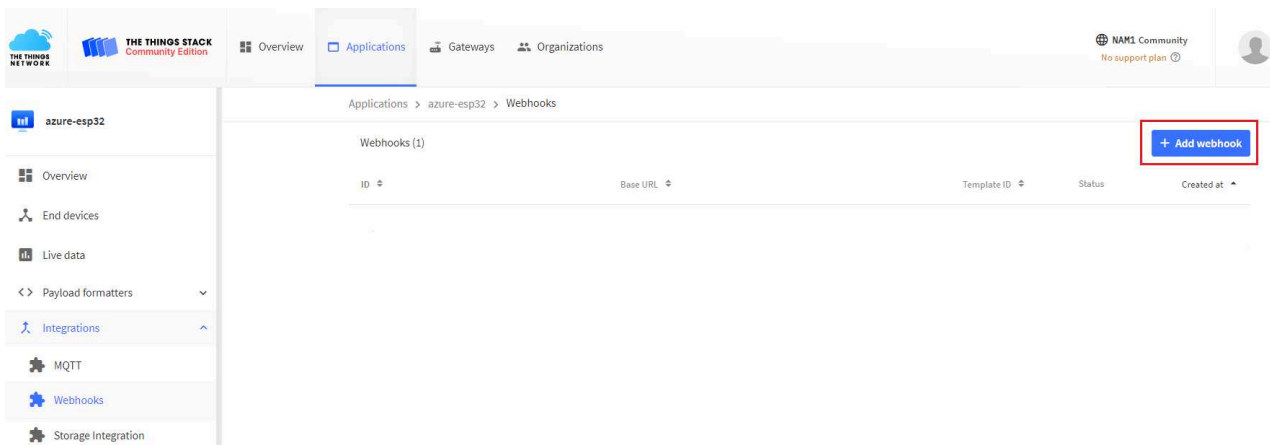


Figura 6-20.: Creación de un Webhook para Cayenne.

2. Aparece la siguiente lista de la cual se selecciona la de Cayenne fig. 6-21.

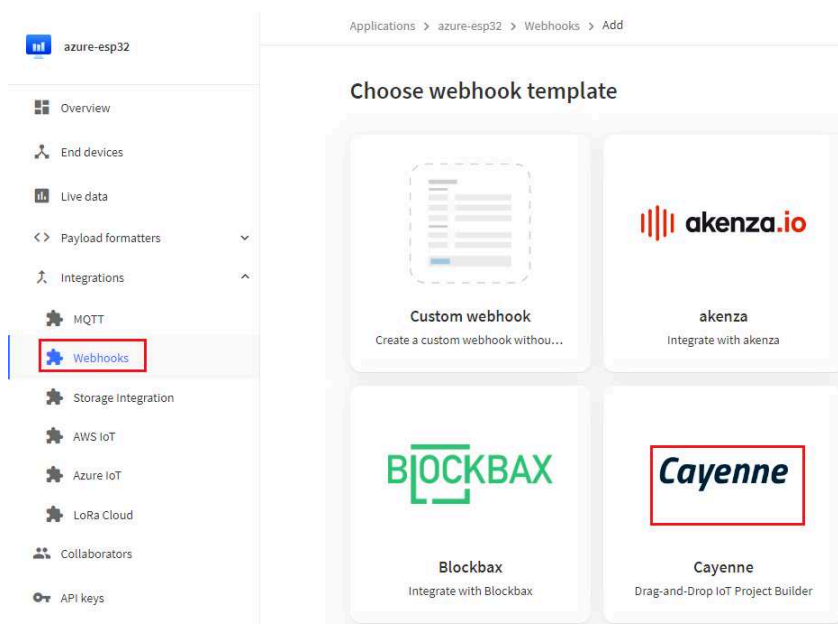
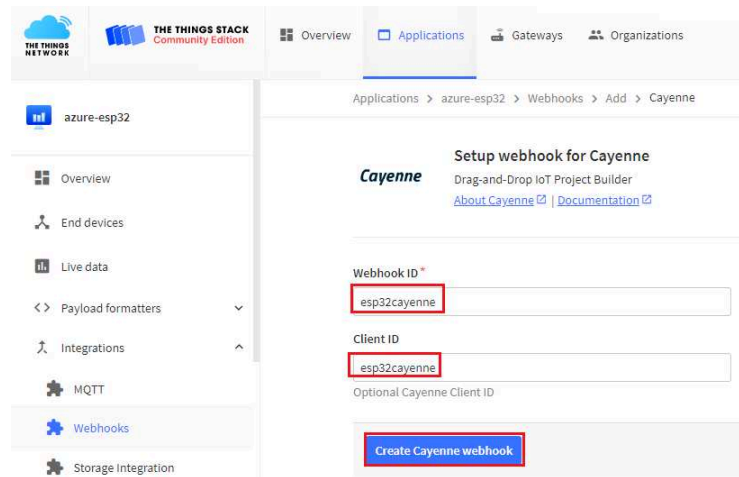


Figura 6-21.: Seleccionamos el template de Cayenne.

3. Configuramos los siguientes campos fig. 6-22:

**Webhook ID:** Se crea un nombre del Webhook.

**Client ID:** Se le asigna un nombre al Client ID debe ser distinto al primero.

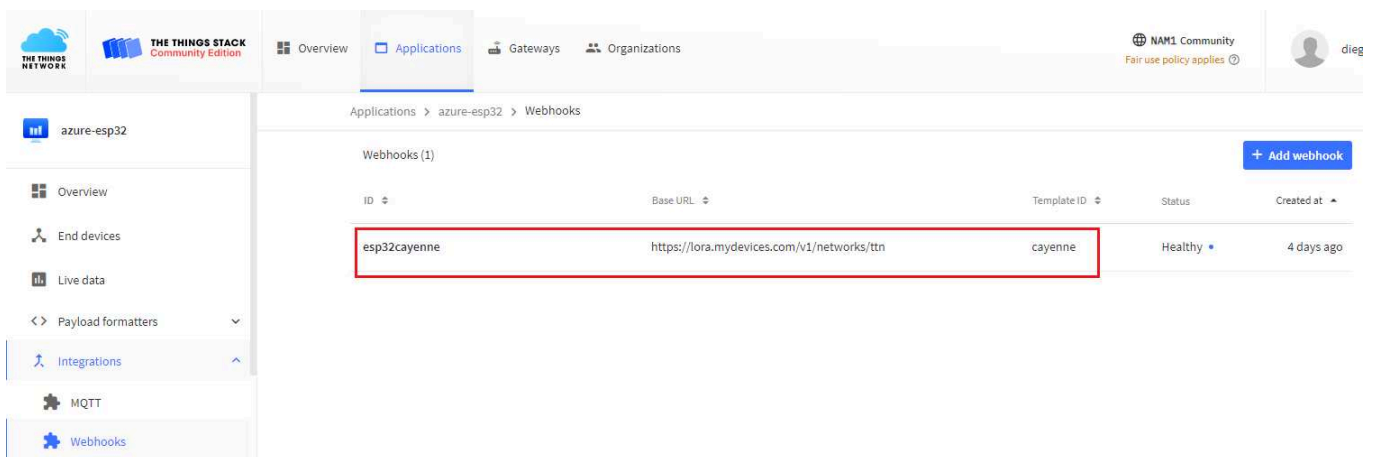


The screenshot shows the 'Setup webhook for Cayenne' form in the Cayenne web interface. The form is titled 'Cayenne Setup webhook for Cayenne' and includes the following fields:

- Webhook ID:** A text input field containing the value 'esp32cayenne'.
- Client ID:** A text input field containing the value 'esp32cayenne'.
- Optional Cayenne Client ID:** An empty text input field.
- Create Cayenne webhook:** A blue button to submit the form.

Figura 6-22.: Formulario para la creación de Cayenne.

4. Se visualiza la correcta creación del Webhooks Cayenne como se observa en la fig. 6-23.

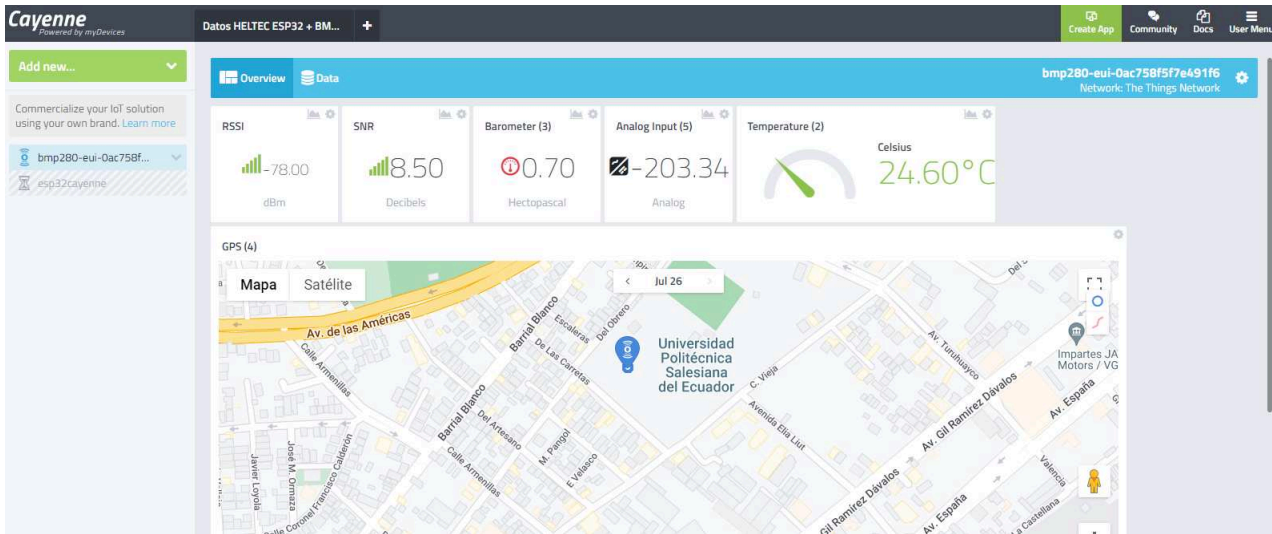


The screenshot shows the 'Webhooks (1)' table in the Cayenne web interface. The table has the following columns: ID, Base URL, Template ID, Status, and Created at. The table contains one entry with the following data:

ID	Base URL	Template ID	Status	Created at
esp32cayenne	https://lora.mydevices.com/v1/networks/ttn	cayenne	Healthy	4 days ago

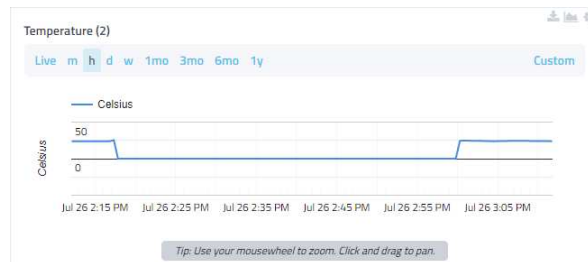
Figura 6-23.: Creacion correcta de Cayenne.

5. Nos dirigimos a la aplicación web Cayenne Dashboard y nos aparece el dispositivo creado dentro de Cayenne se visualiza los datos como son la temperatura, presión, la altitud entre otros fig. 6-24.



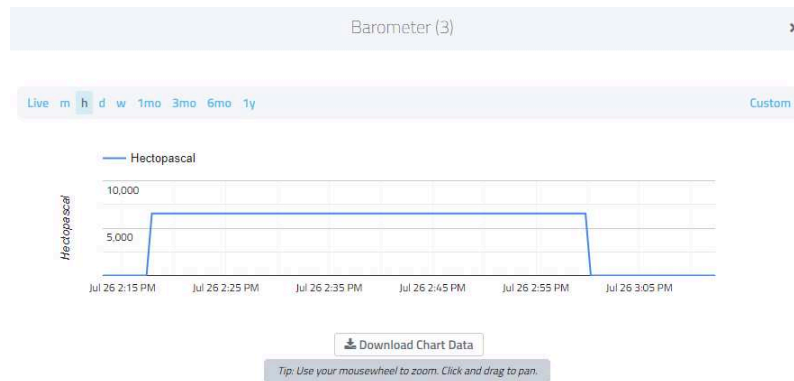
**Figura 6-24.:** Visualización por gráficos de las mediciones en Cayenne.

6. El siguiente gráfico corresponde a la toma de datos en tiempo real sobre la Temperatura fig. 6-25 que recolecta desde el sensor BMP280.



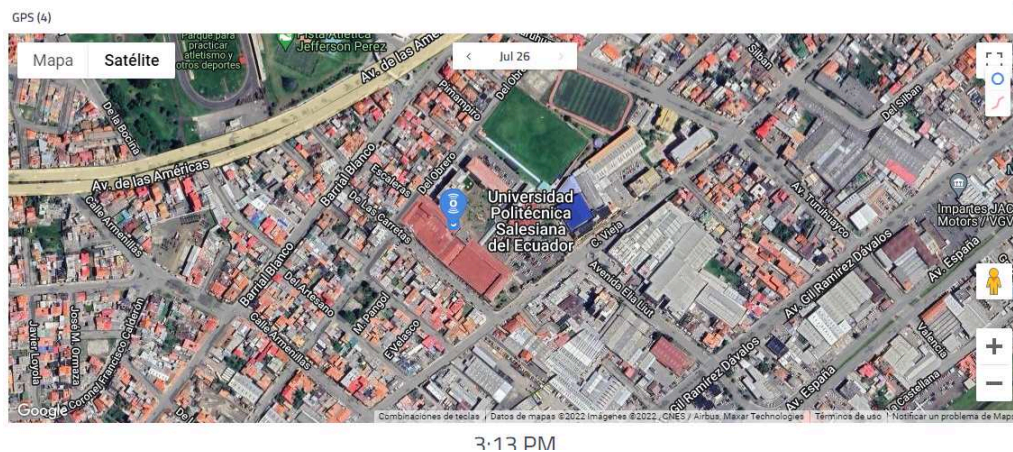
**Figura 6-25.:** Gráfico de la temperatura en Cayenne.

7. El siguiente gráfico corresponde a la toma de datos en tiempo real sobre la Presión fig. 6-26 que recolecta desde el sensor BMP280.



**Figura 6-26.:** Gráfico de la presión en Cayenne.

8. El siguiente gráfico corresponde a la toma de datos en tiempo real sobre la Altitud fig. 6-27 que recolecta desde el sensor BMP280.



**Figura 6-27.:** Localización del dispositivo IoT en Cayenne.

9. Por último, se visualiza los datos en una tabla, estos ingresan en tiempo real como se ve en la fig. A-4, se puede observar los cambios en los datos capturados por el sensor BMP280.

The screenshot displays the Cayenne IoT dashboard interface. On the left, there is a sidebar with a search bar and a list of devices, including 'bmp280-eui-0ac758f...' and 'esp32cayenne'. The main area shows a 'Data' tab for the selected device. The data is presented in a table with the following columns: Timestamp, Device Name, Channel, Sensor Name, Sensor ID, Data Type, Unit, and Values. The table contains 20 rows of data, showing real-time sensor readings from 2022-07-26 3:16:57 to 2022-07-26 3:15:54. The sensor readings include RSSI, SNR, Temperature (2), Barometer (3), GPS (4), Analog Input (5), and various units like dbm, db, c, hpa, m, and null.

Timestamp	Device Name	Channel	Sensor Name	Sensor ID	Data Type	Unit	Values
2022-07-26 3:16:57	bmp280-eui-0...	100	RSSI	1f429720-0796-11ed-8df2-dd5...	rsssi	dbm	-85
2022-07-26 3:16:57	bmp280-eui-0...	101	SNR	1f6b54d0-0796-11ed-bbc1-5d0...	snr	db	8.5
2022-07-26 3:16:57	bmp280-eui-0...	2	Temperature (2)	1f7985a0-0796-11ed-bbc1-5d0...	temp	c	28.1
2022-07-26 3:16:57	bmp280-eui-0...	3	Barometer (3)	1f90b720-0796-11ed-bbc1-5d0...	bp	hpa	0.7
2022-07-26 3:16:57	bmp280-eui-0...	4	GPS (4)	1fae5140-0796-11ed-bbc1-5d0...	gps	m	-2.8866,-78.9908,2418.29
2022-07-26 3:16:57	bmp280-eui-0...	5	Analog Input (5)	1fc6bb40-0796-11ed-bbc1-5d0...	null	null	-203.15
2022-07-26 3:16:25	bmp280-eui-0...	100	RSSI	1f429720-0796-11ed-8df2-dd5...	rsssi	dbm	-85
2022-07-26 3:16:25	bmp280-eui-0...	101	SNR	1f6b54d0-0796-11ed-bbc1-5d0...	snr	db	10
2022-07-26 3:16:25	bmp280-eui-0...	2	Temperature (2)	1f7985a0-0796-11ed-bbc1-5d0...	temp	c	31.5
2022-07-26 3:16:25	bmp280-eui-0...	3	Barometer (3)	1f90b720-0796-11ed-bbc1-5d0...	bp	hpa	0.7
2022-07-26 3:16:25	bmp280-eui-0...	4	GPS (4)	1fae5140-0796-11ed-bbc1-5d0...	gps	m	-2.8866,-78.9908,2418.25
2022-07-26 3:16:25	bmp280-eui-0...	5	Analog Input (5)	1fc6bb40-0796-11ed-bbc1-5d0...	null	null	-203.19
2022-07-26 3:15:54	bmp280-eui-0...	100	RSSI	1f429720-0796-11ed-8df2-dd5...	rsssi	dbm	-84
2022-07-26 3:15:54	bmp280-eui-0...	101	SNR	1f6b54d0-0796-11ed-bbc1-5d0...	snr	db	8.25
2022-07-26 3:15:54	bmp280-eui-0...	2	Temperature (2)	1f7985a0-0796-11ed-bbc1-5d0...	temp	c	24.3
2022-07-26 3:15:54	bmp280-eui-0...	3	Barometer (3)	1f90b720-0796-11ed-bbc1-5d0...	bp	hpa	0.7
2022-07-26 3:15:54	bmp280-eui-0...	4	GPS (4)	1fae5140-0796-11ed-bbc1-5d0...	gps	m	-2.8866,-78.9908,2418.24
2022-07-26 3:15:54	bmp280-eui-0...	5	Analog Input (5)	1fc6bb40-0796-11ed-bbc1-5d0...	null	null	-203.2

Figura 6-28.: Tabla de datos enviados por el sensor en tiempo real en Cayenne.

# 7. CRONOGRAMA

Actividades	Duración	Comienzo	Fin
<b>OE. 1</b>			
ACT. 1. Estudio de los fundamentos de las Redes LoRaWAN.	28 h	lun 11/04/22	mie 13/04/22
ACT. 2. Estudio y análisis de las herramientas de TTN.	28 h	jue 14/04/22	lun 18/04/22
ACT. 3. Estudio de las bases fundamentales de dispositivos IoT.	28 h	mar 19/04/22	mie 20/04/22
ACT. 4. Estudio de las características del Gateway.	28 h	jue 21/04/22	lun 25/04/22
ACT. 5. Estudio de las características del Servidor Azure IoT Hub.	28 h	mar 26/4/22	jue 28/04/22
<b>OE. 2</b>			
ACT. 1. Analizar las características de TTN con las redes LoRaWAN.	32 h	mie 04/05/22	vie 06/05/22
ACT. 2. Analizar las configuraciones de las herramientas de TTN.	32 h	jue 12/05/22	lun 16/05/22
ACT. 3. Analizar el despliegue de la red LoRaWAN con TTN.	32 h	mar 17/05/22	mar 24/05/22
<b>OE. 3</b>			
ACT. 1. Diseño de la red LoRaWAN con TTN.	40 h	lun 30/05/22	jue 02/06/22
ACT. 2. Implementación y configuración del Gateway con TTN.	40 h	vie 03/06/22	mie 08/06/22
ACT. 3. Crear la conexión del gateway con los nodos.	40 h	jue 09/06/22	mar 14/06/22
ACT. 4. Programación del dispositivo LoRa Heltec ESP32 y Sensor BMP280.	80 h	mie 15/06/22	jue 23/06/22
<b>OE. 4</b>			
ACT. 1. Implementar el servidor de aplicaciones Azure IoT Hub en TTN.	80 h	vie 24/06/22	lun 04/07/22
ACT. 2. Gestionar la comunicación de Azure IoT Hub y obtener los datos desde TTN.	50 h	mar 05/07/22	mar 12/07/22
ACT. 3. Visualizar el comportamiento de los nodos.	40 h	mie 13/07/22	mie 20/07/22
<b>OE. 5</b>			
ACT. 1. Recolección de datos generados de las pruebas con el Servicio Stream Analytics de Azure IoT Hub.	50 h	jue 21/07/22	vie 29/07/22
ACT. 2. Implementar Servicio Cosmos DB de Azure IoT Hub para filtrar de datos.	50 h	lun 01/08/22	jue 04/8/22
ACT. 3. Implementar Servicio Storage Blob de Azure IoT Hub para presentación de datos.	50 h	vie 05/08/22	mie 10/08/22
ACT. 4. Implementar Servicio Power BI, Cayenne, PowerBI Robots de Azure IoT Hub para presentar resultados y reportes de datos.	50 h	jue 11/08/22	mie 24/08/22
<b>TOTAL</b>	<b>800 horas</b>	lun 11/04/22	mie 31/08/22

Tabla 7-1.: Cronograma de actividades

## 8. PRESUPUESTO

Denominación	Cantidad(Unidades)	Costo Unitario - dólares	Total - dólares
<b>1.Tecnológico</b>			
Computador Portátil	2	1000	2000
Celular Inteligente	2	300	600
Equipo Gateway	1	200	200
Dispositivo LoRa HelTec ESP32	1	35	35
Sensores inalámbricos	2	6	12
Protoboard	1	5	5
Cables para protoboard	10	0,15	1,50
Servidor en Nube Azure	1	200	360
<b>2.Servicios</b>			
Servicio de transporte	600	0,30	180
Datos Móviles	30Gb	5	40
Taxis	2	5	10
Alimentación	200	2,50	500
<b>TOTAL</b>			<b>\$3.943,50</b>

Tabla 8-1.: Presupuesto para el proyecto.



## 9. CONCLUSIONES

En el transcurso de esta investigación se pudo desarrollar el estado del arte de las redes LoRaWAN con el servidor TTN, pudiendo conocer a fondo dichas tecnologías que servirá de marco referencial para futuras investigaciones. También se analizaron las diferentes características de TTN en base al despliegue de las redes LoRaWAN.

En el diseño y desarrollo de este proyecto técnico y bajo varias pruebas realizadas, se analizó más a fondo la gestión e integración del GW con sus aplicaciones creadas y conectadas en el servidor TTN, considerando que dio inconvenientes de compatibilidad con los medidores de agua, se buscó una solución para que exista la conexión con el GW y con el TTN. Para que exista esta comunicación hubo que adquirir y programar nuestro propio dispositivo como ya se lo indico anteriormente. En esta misma línea, también se integró sin ningún inconveniente el servicio de Azure IoT Hub en TTN para poder gestionar la información que captura los nodos finales y a su vez también no hubo conflictos de aceptación en el servidor de aplicaciones de Azure IoT Hub.

Con la implementación de diferentes servicios con los que cuenta Microsoft Azure IoT Hub, los datos enviados desde el dispositivo HelTec ESP32 con el sensor de Presión BMP280 fueron almacenados en una base de datos y de esta manera poder visualizar el comportamiento de los nodos, su funcionamiento, fallas, mensajes, entre otros. Muchos de los servicios adicionales que fueron implementados en Azure fueron de gran ayuda a la hora de recolectar los datos, entre ellos están Stream Analytics, Storage Account Blob, Cosmos DB, y Power BI, con el uso de estos servicios fue mucho más fácil recopilar y analizar los resultados requeridos en este proyecto de IoT. La captura de los datos con el servicio de Cosmos DB fueron satisfactorios, mediante varias pruebas realizadas, con la toma de datos desde el nodo pasando por el servidor de red TTN hasta el servidor de aplicaciones Azure IoT Hub, con una obtención de datos variables, con lo que se permitió generar resultados claros y precisos. También se hace uso de una plataforma que nos permite ver los datos en tiempo real, como es el caso de la herramienta de Cayenne, nos permite visualizar los datos que se envían desde el nodo hacia el servidor TTN, en tiempo real y con visualización más precisa de los resultados obtenidos.

Con la arquitectura de red establecida en el capítulo 3, la cual nos permite realizar todas las conexiones necesarias para la realización de pruebas y analizar el comportamiento de los nodos, este tipo de arquitectura nos permite poder monitorizar y gestionar dispositivos IoT

por medio de Internet, de esta manera la implementación de una LoRaWAN puede ayudar mucho para dar paso a tecnología moderna y duradera, ya que los nodos pueden incluir sus propias baterías lo que hace que esta red sea de gran ayuda para establecerse a gran escala.

## 10. RECOMENDACIONES

Para poder implementar una arquitectura de red LoRaWAN es necesario definir la estructura de la red, las herramientas necesarias que se requiere para poder gestionar los diferentes dispositivos, se deben elegir dispositivos compatibles con la plataforma de TTN, se debe considerar como es el funcionamiento de cada servicio de Azure IoT Hub.

Se recomienda que, para trabajos futuros, los servicios implementados para este proyecto se las puedan intercambiar con servicios de mejores características, tal sea el caso de poder usar una base de datos con mejores recursos y así mismo realizar la visualización de resultados en tiempo real usando el servicio de Azure Power BI de pago, esto no fue posible implementar para este proyecto por no poder contar con un apoyo económico.

Considerar en adquirir los equipos con anticipación, ya que al ser una tecnología innovadora en la Ciudad de Cuenca, no fue muy accesible la adquisición de dispositivos de IoT, por lo que se debe tener en cuenta a la hora de realizar un proyecto LoRaWAN, dada todas estas recomendaciones será más fácil y factible trabajar en cualquier proyecto o investigación de estas características.

# 11. TRABAJO FUTURO

Basándonos en las pruebas y resultados obtenidos en este proyecto, queda como interrogante aplicar los mismos pasos y configuraciones e implementar la integración con muchos más dispositivos IoT de diferentes proveedores, ya que según la información recaudada tanto TTN como Azure IoT Hub pueden gestionar grandes cantidades de dispositivos IoT, lo que puede ser de gran ayuda para elaborar proyectos más avanzados y con mayor número de dispositivos finales.

Para trabajos a futuros, la implementación del GW se le puede proyectar con un área de cobertura mucho mayor a comparación a la capacidad que se trabajó en este proyecto siendo este de poco rango de alcance, por lo que no se puede realizar pruebas a gran escala o conectar con otros equipos de mejores características.

# Bibliografía

- [1] A. Lavric and A. I. Petrariu, “Lorawan communication protocol: The new era of iot,” in *2018 International Conference on Development and Application Systems (DAS)*, pp. 74–77, IEEE, 2018.
- [2] M. Mehic, M. Duliman, N. Selimovic, and M. Voznak, “Lorawan end nodes: Security and energy efficiency analysis,” *Alexandria Engineering Journal*, vol. 61, no. 11, pp. 8997–9009, 2022.
- [3] J. D. Alvarado Menéndez and L. A. Córdova Pilaló, *Diseño de red LPWAN (low power wide area network) y desarrollo del prototipo para monitoreo de dispositivos de control de sistemas de paneles solares y la integración con GATEWAY LORAWAN de MIKROTIK*. PhD thesis, Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas ..., 2020.
- [4] J. de Carvalho Silva, J. J. Rodrigues, A. M. Alberti, P. Solic, and A. L. Aquino, “Lorawan—a low power wan protocol for internet of things: A review and opportunities,” in *2017 2nd International multidisciplinary conference on computer and energy science (SpliTech)*, pp. 1–6, IEEE, 2017.
- [5] J. C. Merino Polidura *et al.*, “Despliegue y evaluación de una red onsite lorawan basada en the things network stack versión 3,” 2019.
- [6] mydevices, “Cayenne.” <https://developers.mydevices.com/cayenne/features>, 2022.
- [7] S. Klein, “Azure stream analytics,” in *IoT Solutions in Microsoft’s Azure IoT Suite*, pp. 71–84, Springer, 2017.
- [8] B. Kettner and F. Geisler, “Processing streams of data,” in *Pro Serverless Data Handling with Microsoft Azure*, pp. 231–240, Springer, 2022.
- [9] S. Arora and R. Rani, “A streamlined approach for real-time data analytics,” in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pp. 732–736, IEEE, 2018.
- [10] B. Power, U. Excel, P. Desktop, and P. Tiles, “Microsoft power bi,” *Available here: <https://powerbi.microsoft.com/en-us>*, 2021.

- [11] J. R. Cotrim and J. H. Kleinschmidt, “Lorawan mesh networks: A review and classification of multihop communication,” *Sensors*, vol. 20, no. 15, p. 4273, 2020.
- [12] mikrotik.com, “Mikrotik.” <https://mikrotik.com/aboutus>, 2022.
- [13] M. Pereira and P. Cruvinel, “Desenvolvimento de um sistema de coleta automática de dados agrícolas baseado em rede lora e no microprocessador esp32,” in *Anais da X Escola Regional de Informática de Mato Grosso*, pp. 43–48, SBC, 2019.
- [14] sigmaelectronica, “Módulo sensor de presión atmosférica bmp280..” <https://www.sigmaelectronica.net/producto/tarjeta-bmp280/>, 2022.
- [15] P. Bertoleti, *Proyectos com ESP32 y LoRa*. Editora NCB, 2019.
- [16] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, “A comparative study of lpwan technologies for large-scale iot deployment,” *ICT express*, vol. 5, no. 1, pp. 1–7, 2019.
- [17] mikrotik, “wap lr8 kit.” [https://mikrotik.com/product/wap\\_lr8\\_kit](https://mikrotik.com/product/wap_lr8_kit), 2022.
- [18] O. Q. Muñoz, *Internet de las Cosas (IoT)*. Ibukku LLC, 2019.
- [19] Europapress, “En 2023 habrá 29.300 millones de dispositivos conectados a internet, según cisco.” <https://www.europapress.es/portaltic/internet/noticia-2023-habra-29300-millones-dispositivos-conectados-internet-cisco-20200517112937.html>, 2022.
- [20] A. Lavric and V. Popa, “A lorawan: Long range wide area networks study,” in *2017 International Conference on Electromechanical and Power Systems (SIELMEN)*, pp. 417–420, IEEE, 2017.
- [21] M. Parra, E. Guillen, F. Le Mouël, and O. Carrillo, “Sistema colaborativo de medición de parámetros ambientales basado en iot,” in *2nd Workshop CATAÍ-SmartData for Citizen Wellness*, 2019.
- [22] R. Stackowiak, “Azure iot hub,” in *Azure Internet of Things Revealed*, pp. 73–85, Springer, 2019.
- [23] A. Osorio, M. Calle, J. D. Soto, and J. E. Candelo-Becerra, “Routing in lorawan: Overview and challenges,” *IEEE Communications Magazine*, vol. 58, no. 6, pp. 72–76, 2020.
- [24] D. Davcev, K. Mitreski, S. Trajkovic, V. Nikolovski, and N. Koteli, “Iot agriculture system based on lorawan,” in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, pp. 1–4, IEEE, 2018.
- [25] L. H. Haug, “An indoor/outdoor air quality relationship analysis using internet of things,” Master’s thesis, The University of Bergen, 2019.

- [26] S. W. Cevallos García, Bryan Leonel Rubio Echeverría, “Desarrollo de una red iot con tecnología lora para gestión de invernaderos,” Master’s thesis, Quito, 2021.
- [27] G. Laura, “Test y despliegue de tecnología de comunicaciones lora para aplicaciones de internet of things,” Master’s thesis, Madrid, 2019.
- [28] A. Sourì, A. Hussien, M. Hoseyninezhad, and M. Norouzi, “A systematic review of iot communication strategies for an efficient smart environment,” *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, p. e3736, 2022.
- [29] S. Ahmetoglu, Z. Che Cob, and N. Ali, “A systematic review of internet of things adoption in organizations: Taxonomy, benefits, challenges and critical factors,” *Applied Sciences*, vol. 12, no. 9, p. 4117, 2022.
- [30] D. P. Rodriguez Alvarado and E. J. Sacoto-Cabrera, “Implementation and analysis of the results of the application of the methodology for hybrid multi-cloud replication systems,” in *The International Conference on Advances in Emerging Trends and Technologies*, pp. 273–286, Springer, 2021.
- [31] A. Sanchis-Cano, J. Romero, E. J. Sacoto-Cabrera, and L. Guijarro, “Economic feasibility of wireless sensor network-based service provision in a duopoly setting with a monopolist operator,” *Sensors*, vol. 17, no. 12, p. 2727, 2017.
- [32] G. A. León-Paredes, O. G. Bravo-Quezada, E. J. Sacoto-Cabrera, O. F. Pizarro-Gordillo, P. E. Vintimilla-Tapia, J. F. Bravo-Torres, and W. P. Cabrera-Chica, “Virtual reality and data analysis based platform for urban mobility awareness as a tool for road education,” in *2020 IEEE ANDESCON*, pp. 1–6, 2020.
- [33] G. A. León-Paredes, O. G. Bravo-Quezada, E. J. Sacoto-Cabrera, W. F. Calle-Siavichay, L. L. Jiménez-González, and J. Aguirre-Benalcazar, “Virtual reality platform for sustainable road education among users of urban mobility in cuenca, ecuador,” *Virtual Reality*, vol. 13, no. 6, 2022.
- [34] S. M. González, “Entendiendo el internet de las cosas,” *Investiga. TEC*, no. 24, pp. página–22, 2015.
- [35] J. Aranda, E. J. Sacoto Cabrera, D. Haro Mendoza, and F. Astudillo Salinas, “5g networks: A review from the perspectives of architecture, business models, cybersecurity, and research developments,” *Novasinerгия*, vol. 4, 2021.
- [36] S. Aguilar Zavaleta, “Diseño de una solución basada en el internet de las cosas (iot) empleando lorawan para el monitoreo de cultivos agrícolas en Perú,” 2020.
- [37] H. Larian, A. Larian, M. Sharifi, and H. Movahednejad, “Towards web of things middleware: A systematic review,” *arXiv preprint arXiv:2201.08456*, 2022.

- [38] E. Sacoto Cabrera, *Análisis basado en teoría de juegos de modelos de negocio de operadores móviles virtuales en redes 4G y 5G*. PhD thesis, Universitat Politècnica de València, 2021.
- [39] E. J. Sacoto-Cabrera, L. Guijarro, J. R. Vidal, and V. Pla, “Economic feasibility of virtual operators in 5g via network slicing,” *Future Generation Computer Systems*, vol. 109, pp. 172–187, 2020.
- [40] E. J. Sacoto Cabrera, L. Guijarro, and P. Maillé, “Game theoretical analysis of a multi-mno mvno business model in 5g networks,” *Electronics*, vol. 9, no. 6, p. 933, 2020.
- [41] V. Vimos and E. J. S. Cabrera, “Results of the implementation of a sensor network based on arduino devices and multiplatform applications using the standard opc ua,” *IEEE Latin America Transactions*, vol. 16, no. 9, pp. 2496–2502, 2018.
- [42] E. J. Sacoto-Cabrera, A. Sanchis-Cano, L. Guijarro, J. R. Vidal, and V. Pla, “Strategic interaction between operators in the context of spectrum sharing for 5g networks,” *Wireless Communications and Mobile Computing*, vol. 2018.
- [43] M. G. Ortiz Sosa, “Desarrollo de una red de sensores inalámbricos utilizando tecnología lora para el monitoreo de un sistema,” B.S. thesis, 2020.
- [44] M. Mroue, A. Nasser, B. Parrein, A. Mansour, C. Zaki, and E. M. Cruz, “Esco: Eligibility score-based strategy for sensors selection in cr-iiot: Application to lorawan,” *Internet of Things*, vol. 13, p. 100362, 2021.
- [45] E. J. Sacoto-Cabrera, G. León-Paredes, and W. Verdugo-Romero, “Lorawan: application of nonlinear optimization to base stations location,” in *Communication, Smart Technologies and Innovation for Society*, pp. 515–524, Springer, 2022.
- [46] D. Bankov, E. Khorov, and A. Lyakhov, “On the limits of lorawan channel access,” in *2016 International conference on engineering and telecommunication (EnT)*, pp. 10–14, IEEE, 2016.
- [47] M. A. Ertürk, M. A. Aydın, M. T. Büyükakkaşlar, and H. Evirgen, “A survey on lorawan architecture, protocol and technologies,” *Future Internet*, vol. 11, no. 10, p. 216, 2019.
- [48] D. Rodríguez Díaz, “Red privada lorawan para el ámbito de aplicaciones en la industria 4.0,” B.S. thesis, Universitat Politècnica de Catalunya, 2021.
- [49] F. X. Mejía Tamayo, “Propuesta de solución internet of things (iiot) con tecnología lorawan, para un sistema de parqueadero inteligente (spi). caso de estudio conjunto habitacional parque real (chpr),” Master’s thesis, PUCE-Quito, 2021.



- [50] B. Asadollahseraj, *Design of Low Power Gateways for LoRaWAN Applications In Remote Areas*. PhD thesis, Politecnico di Torino, 2018.
- [51] M. Aernouts, R. Berkvens, K. Van Vlaenderen, and M. Weyn, “Sigfox and lorawan datasets for fingerprint localization in large urban and rural areas,” *Data*, vol. 3, no. 2, p. 13, 2018.
- [52] V. H. Lopez Chalacan, “Performance evaluation of long range (lora) wireless rf technology for the internet of things (iot) using dragino lora at 915 mhz,” 2020.
- [53] J. R. Guay Paz, “Introduction to azure cosmos db,” in *Microsoft Azure Cosmos DB Revealed*, pp. 1–23, Springer, 2018.
- [54] R. Stackowiak, “Analyzing and visualizing data in azure,” in *Azure Internet of Things Revealed*, pp. 87–118, Springer, 2019.
- [55] D. Agarwal and S. K. Prasad, “Azurebench: Benchmarking the storage services of the azure cloud platform,” in *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, pp. 1048–1057, IEEE, 2012.
- [56] R. K. Kodali, S. Yerroju, and S. Sahu, “Smart farm monitoring using lora enabled iot,” in *2018 second international conference on green computing and internet of things (ICGCIoT)*, pp. 391–394, IEEE, 2018.
- [57] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, “Concept and hardware considerations for product-service system achievement in internet of things,” in *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, pp. 1–6, IEEE, 2019.
- [58] J. E. Quimis Suárez, *Diseño de un prototipo experimental de medición de señales de ambientes de contaminación acústica*. PhD thesis, Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas ..., 2021.
- [59] J. M. Sulca Coral, “Adaptación de la biblioteca lmic en arduino para soportar la especificación lorawan versión 1.1.,” B.S. thesis, Quito, 2021, 2021.
- [60] B. Sensortec, “Bmp280,” *Bosch Sensortec*: <https://www.boschsensortec.com/products/environmental-sensors/pressure-sensors/pressuresensors-bmp280-1.html>, Accessed, vol. 21, 2020.

# A. Anexo: Código Arduino

En el siguiente anexo se detallan los códigos implementados en el IDE de Arduino, esta desarrollado en el lenguaje de programación es C++ también se incluyen los códigos para sacar el ID del dispositivo HelTec ESP32, así como el código para saber la dirección de bus de datos en la que trabaja el Sensor BMP280 y el código de decodificación de datos Cayenne en la plataforma de TTN para dispositivos IoT.

## A.1. Código implementado para la conexión del dispositivo HelTec ESP32 V2 + Sensor BMP280 en TTN

Código que se debe cargar en el dispositivo HelTec ESP32 para poder establecer la comunicación y el envío de datos entre el nodo y la plataforma de TTN.

## A.1 Código implementado para la conexión del dispositivo HelTec ESP32 V2 + Sensor BMP280 en TTN

135

```
1 #include <Wire.h>
2 #include <ESP32_LoRaWAN.h>
3 #include <Arduino.h>
4 #include <CayenneLPP.h>
5 #include <Adafruit_Sensor.h>
6 #include <Adafruit_BMP280.h>
7
8 #define SEALEVELPRESSURE_HPA (1006.0)
9 Adafruit_BMP280 bme; // I2Cheltec.h Heltec;
10 float temp, pa, alt;
11
12 CayenneLPP lpp(51); // create a buffer of 51 bytes to store the payload
13 unsigned long delayTime;
14 /*license for Heltec ESP32 LoRaWan, query your ChipID relevant license: http://resource.heltec.cn/search */
15 uint32_t license[4] = { 0x03E061B3,0xC165BCB2,0XECE1AAC4,0x11075A78 };
16 /* OTAA parametros*/
17 uint8_t DevEui[] = { 0x5A, 0xD2, 0xA0, 0x00, 0xD0, 0x70, 0x7B, 0x9C };
18 uint8_t AppEui[] = { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 };
19 uint8_t AppKey[] = { 0x97, 0x55, 0x94, 0xC3, 0xBF, 0xAD, 0x9E, 0x20, 0xDB, 0x15, 0x86, 0x04, 0xFE, 0x71, 0xAE, 0x3B };
20 /* ABP para* Need for Compile */
21 uint8_t NwkSKey[] = { 0x15, 0xB1, 0xD0, 0xEF, 0xA4, 0x63, 0xDF, 0xBE, 0x3D, 0x11, 0x18, 0x1E, 0x1E, 0xC7, 0xDA, 0x11 };
22 uint8_t AppSKey[] = { 0xD7, 0x2C, 0x78, 0x75, 0x8C, 0xDC, 0xCA, 0xBF, 0x55, 0xEE, 0x4A, 0x77, 0x8D, 0x16, 0xEF, 0x11 };
23 uint32_t DevAddr = ( uint32_t )0x007e6a11;
24
25 /*LoRaWan channelmask, default channels 0-7*/
26 uint16_t userChannelsMask[6] = {
27   0x00FF, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000
28 };
29 /*LoRaWan Class, Class A and Class C are supported*/
30 DeviceClass_t loraWanClass = CLASS_A;
31 /*the application data transmission duty cycle. value in [ms].*/
32 uint32_t appTxDutyCycle = 30000;
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68 uint8_t debugLevel = LoRaWAN_DEBUG_LEVEL;
69 /*LoRaWan region, select in arduino IDE tools*/
70 LoRaMacRegion_t loraWanRegion = ACTIVE_REGION;
71 // Add your initialization code here
72 void setup() {
73
74   Serial.begin(115200);
75   unsigned status;
76   status = bme.begin(0x76);
77   if (!status) {
78     Serial.println('Could not find a valid BME280 sensor, check wiring, address, sensor ID!');
79   }
80
81   while (!Serial);
82
83   temp = bme.readTemperature();
84   pa = bme.readPressure()/ 100.0F;
85   alt = bme.readAltitude();
86   SPI.begin(SCK, MISO, MOSI, SS);
87   Mcu.init(SS, RST_LoRa, DIO0, DIOL, license);
88
89   deviceState = DEVICE_STATE_INIT;
90
91 }
92 // The loop function is called in an endless loop
93 void loop() {
94   switch (deviceState) {
95     case DEVICE_STATE_INIT: {
96 #if (LoRaWAN_DEVEUI_AUTO)
97     LoRaWAN.generateDeveuiByChipID();
98 #endif
99
100    static void prepareTxFrame(uint8_t port) {
101      Serial.print("Time:");
102      Serial.println(millis());
103      Serial.print("Temperature = ");
104      Serial.print(temp);
105      Serial.println(" °C");
106
107      Serial.print("Pressure = ");
108      Serial.println(pa);
109      Serial.println("hPa");
110      Serial.print("Altitude = ");
111      Serial.print(alt);
112      Serial.println("m");
113      Serial.println();
114    }
115
116    //Se usa los métodos de la biblioteca cayenne para presentar los datos del Sensor BMP280
117
118    lpp.reset(); // clear the buffer
119    lpp.addTemperature(2, temp);
120    lpp.addBarometricPressure(3, pa / 1000);
121    lpp.addGPS(4, -2.8866898925165785, -78.99087846262775, alt);
122    lpp.addAnalogInput(5, alt);
123    appDataSize = lpp.getSize();
124    uint8_t lppBuffer[128];
125    memcpy(lppBuffer, lpp.getBuffer(), 128);
126    memcpy(appData, lppBuffer, lpp.getSize());
127
128 }
129 }
```

Figura A-1.: Código en Arduino para cargar al dispositivo.

### A.1.1. Código para ver el ID del dispositivo HelTec ESP32

Para poder sacar la licencia del dispositivo HelTec ESP32 es necesario saber el ID del dispositivo para lo cual a continuación se muestra el código implementado para poder generarlo.

```
GetChipID $
1 uint32_t chipId = 0;
2
3 void setup() {
4   Serial.begin(115200);
5 }
6
7 void loop() {
8   for(int i=0; i<17; i=i+8) {
9     chipId |= ((ESP.getEfuseMac() >> (40 - i)) & 0xff) << i;
10  }
11
12  Serial.printf("ESP32 Chip model = %s Rev %d\n", ESP.getChipModel(), ESP.getChipRevision());
13  Serial.printf("This chip has %d cores\n", ESP.getChipCores());
14  Serial.print("Chip ID: "); Serial.println(chipId);
15
16  delay(3000);
17
18 }
```

Figura A-2.: Código para obtener el ChipID del dispositivo.

### A.1.2. Código de la biblioteca de Cayenne para TTN

Se muestra el código que contiene la biblioteca de Cayenne la misma que es compatible para poder trabajar con dispositivos de IoT y poder decodificar los datos en la plataforma de TTN.

```
DecodeTTN$
1 #include <CayenneLPP.h>
2 void setup() {
3   DynamicJsonDocument jsonBuffer(1024);
4   CayenneLPP lpp(160);
5
6   JsonObject root = jsonBuffer.to<JsonObject>();
7   Serial.begin(115200);
8   Serial.println();
9
10  lpp.reset();
11  lpp.addDigitalInput(1, 0);
12  lpp.addDigitalOutput(2, 1);
13  lpp.addAnalogInput(3, 1.23f);
14  lpp.addAnalogOutput(4, 3.45f);
15  lpp.addLuminosity(5, 20304);
16  lpp.addPresence(6, 1);
17  lpp.addTemperature(7, 26.5f);
18  lpp.addRelativeHumidity(8, 86.6f);
19  lpp.addAccelerometer(9, 1.234f, -1.234f, 0.567f);
20  lpp.addBarometricPressure(10, 1023.4f);
21  lpp.addGyrometer(1, -12.34f, 45.56f, 89.01f);
22  lpp.addGPS(1, -12.34f, 45.56f, 9.01f);
23
24  lpp.addUnixTime(1, 135005160);
25
26  lpp.addGenericSensor(1, 4294967295);
27  lpp.addVoltage(1, 3.35);
28  lpp.addCurrent(1, 0.321);
29  lpp.addFrequency(1, 50);
30  lpp.addPercentage(1, 100);
31  lpp.addAltitude(1, 50);
32  lpp.addPower(1, 50000);
```

Figura A-3.: Código de la biblioteca Cayenne.

## A.2. Código para ver la dirección de bus del dispositivo

Para poder implementar cualquier tipo de sensor en la programación del dispositivo HelTec ESP32 es necesario conocer cual es la dirección de bus de datos en la que trabaja el sensor, con el código a continuación se puede saber cual es la dirección correcta y implementarlo en la programación de esta manera se evita contratiempos y fallas de conexión.

```
1 #include <Wire.h>
2 void setup() {
3   Wire.begin();
4   Serial.begin(115200);
5   Serial.println("I2C Scanner");
6 }
7 Metodo que nos permite escanear todas las direcciones en las que trabajan los
8 diferentes tipos de sensores.
9 void loop() {
10  byte error, address;
11  int nDevices;
12  Serial.println("Scanning...");
13  nDevices = 0; for (address = 1; address < 127; address++ ) {
14    Wire.beginTransmission(address);
15    error = Wire.endTransmission();
16    if (error == 0) {
17      Serial.print("I2C device found at address 0x");
18      if (address < 16) {
19        Serial.print("0");
20      }
21      Serial.println(address, HEX);
22      nDevices++;
23    }
24    else if (error == 4) {
25      Serial.print("Unknow error at address 0x");
26      if (address < 16) {
27        Serial.print("0");
28      }
29      Serial.println(address, HEX);
30    }
31  }
```

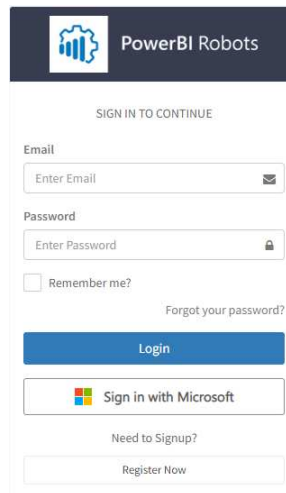
**Figura A-4.:** Código para obtener el BUS del dispositivo.

## B. Anexo: Generar Reportes

En el siguiente anexo vamos a mostrar los pasos de como generar reportes usando Power BI Robots mediante algunas configuraciones podemos crear un PDF para mostrar el dashboard generado de los datos adquiridos de los dispositivos IoT.

### B.1. Pasos para usar PowerBI Robots

1. Primero debemos crearnos una cuenta para poder usar el servicio de PowerBI Robots para lo cual nos dirigimos a la siguiente URL; <https://www.powerbitiles.com/PBIRobots/Account> podemos usar una cuenta existente de Microsoft tal como se observa en la fig.**B-1**.



**Figura B-1.:** Logging PowerBI Robots.

2. Una vez que ya se haya creado una cuenta o usado una cuenta existente ya se puede ingresar a Power BI Robots fig.**B-2**, dentro de esta pagina debemos descargar el archivo **PowerBI Robots Agent** luego nos dirigimos a **Settings** para poder copiar el Id y así crear un reporte.

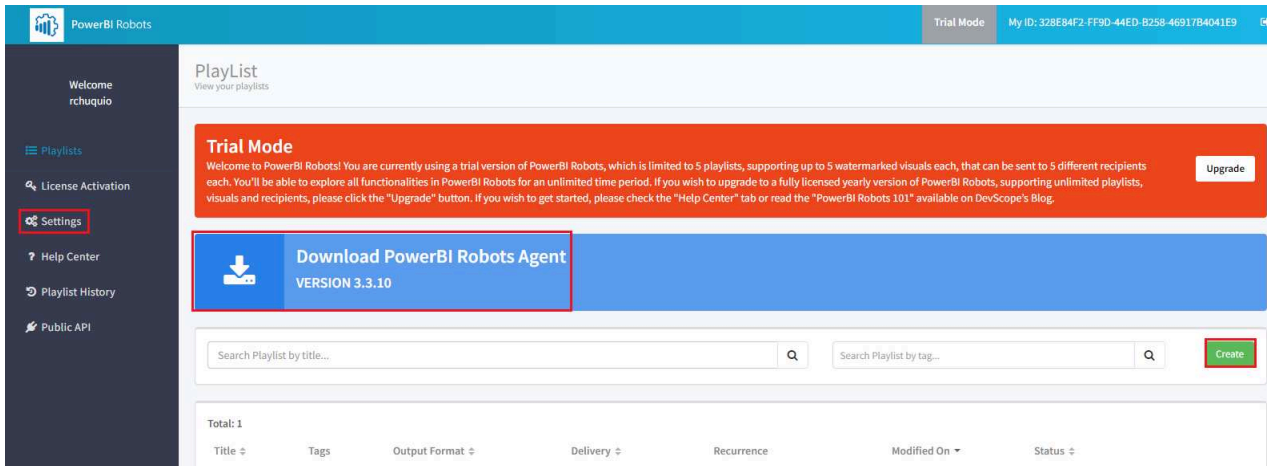


Figura B-2.: Pagina de inicio PowerBI Robots.

3. En esta parte se debe copiar **My Id** ya que este código lo vamos a usar en el PowerBI Robots Agent fig.B-3.

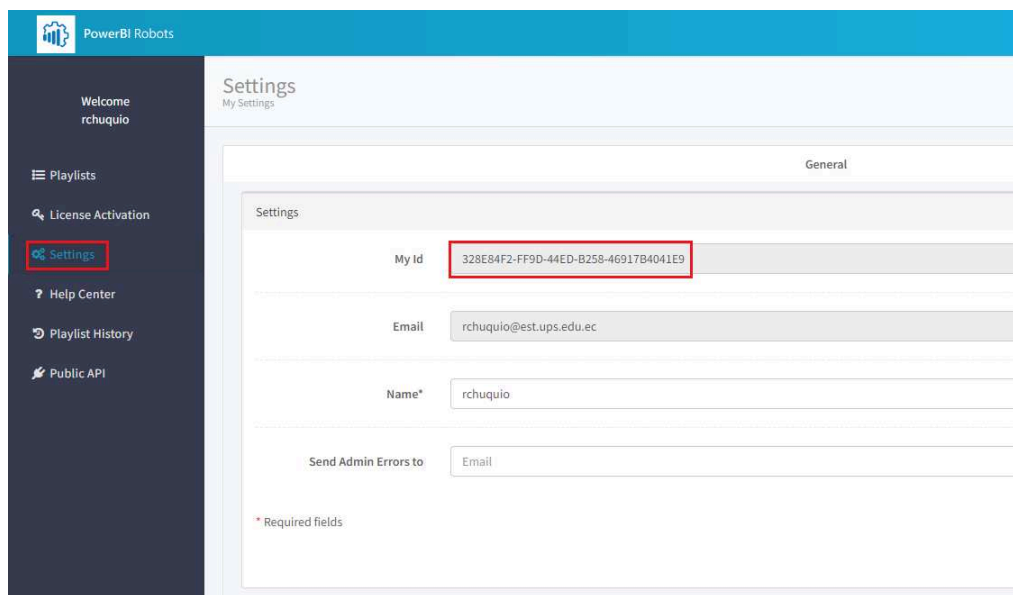


Figura B-3.: Pagina para ver My Id.

4. Debemos ejecutar el archivo descargado PowerBI Robots Agent y nos parece la siguiente fig.B-4, en la cual se debe pegar el **My Id** y configurar el acceso a PowerBI.



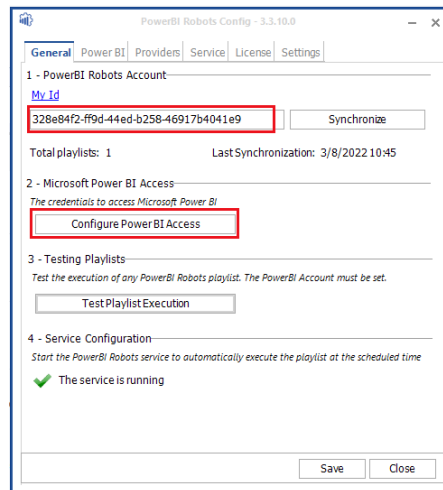


Figura B-4.: Ventana de PowerBI Robots Agent.

5. El acceso a PowerBI se lo puede realizar usando la cuenta existente en donde se tiene los trabajos creados en PowerBI solo es necesario iniciar sesión en su cuenta de Power BI y automáticamente se reconoce la conexión tal como se visualiza en la fig.B-5.

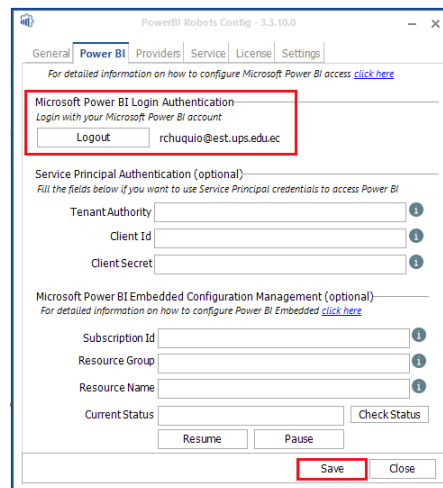


Figura B-5.: Conexión con PowerBI.

6. Una vez realizado la conexión en el punto anterior solo debemos inicializar el servicio de PowerBI Robots para lo cual nos dirigimos a **Service** tal como se muestra en fig.B-6, y luego le damos a **Star**.

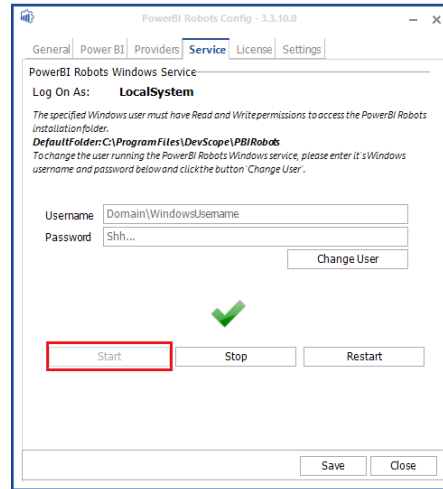


Figura B-6.: Iniciar servicio PowerBI Robots.

7. Se debe crear un nuevo Reporte en la pagina principal web de PowerBI Robots para lo cual debemos llenar el siguiente formulario tal como se observa en la fig.B-7.

Figura B-7.: Formulario para crear Reportes.

8. Se debe especificar una dirección de correo para poder recibir los reportes como se ilustra en la fig.B-8.

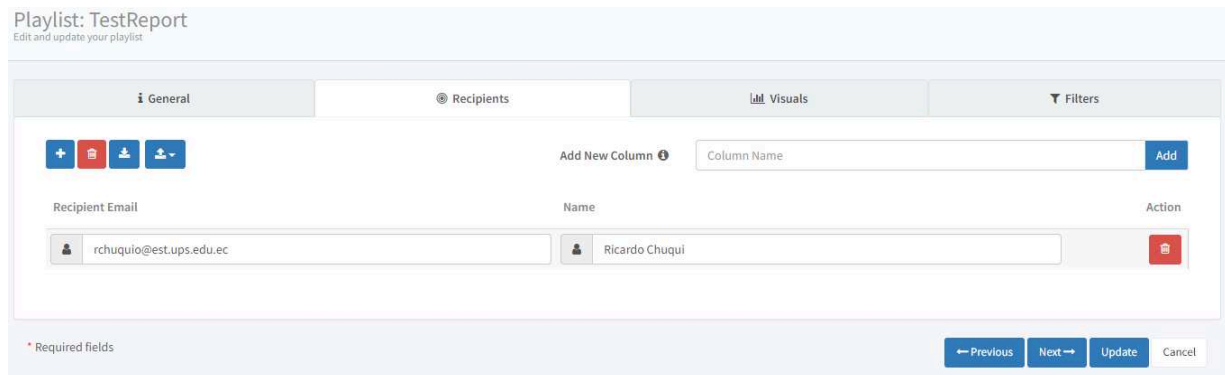


Figura B-8.: Correo de Recepcion.

9. En este punto debemos seleccionar la opción **Add PowerBI Visual** lo que nos llevara a iniciar sesión en el servicio de Power BI solo debemos seleccionar el área de trabajo con el trabajo que desea generar el reporte elegir los campos que necesita y automáticamente se cargan los datos tal como se observa en la fig.B-9.

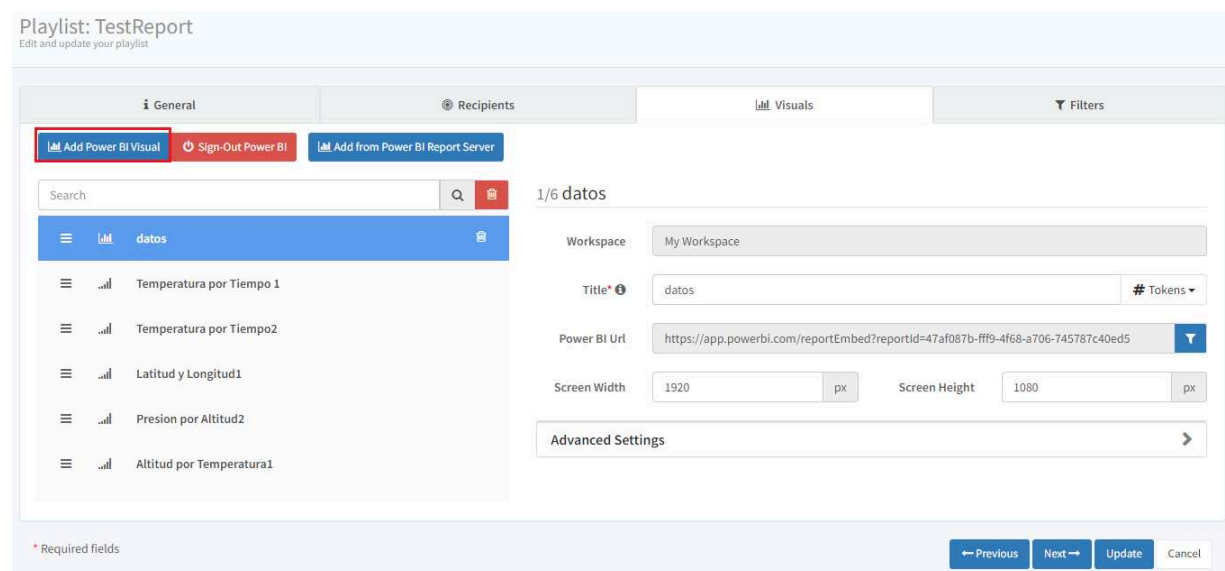


Figura B-9.: Visualización datos de PowerBI.

10. Una vez realizado toda la configuración solo debemos mandar a crear el nuevo archivo de Reportes fig.B-10.

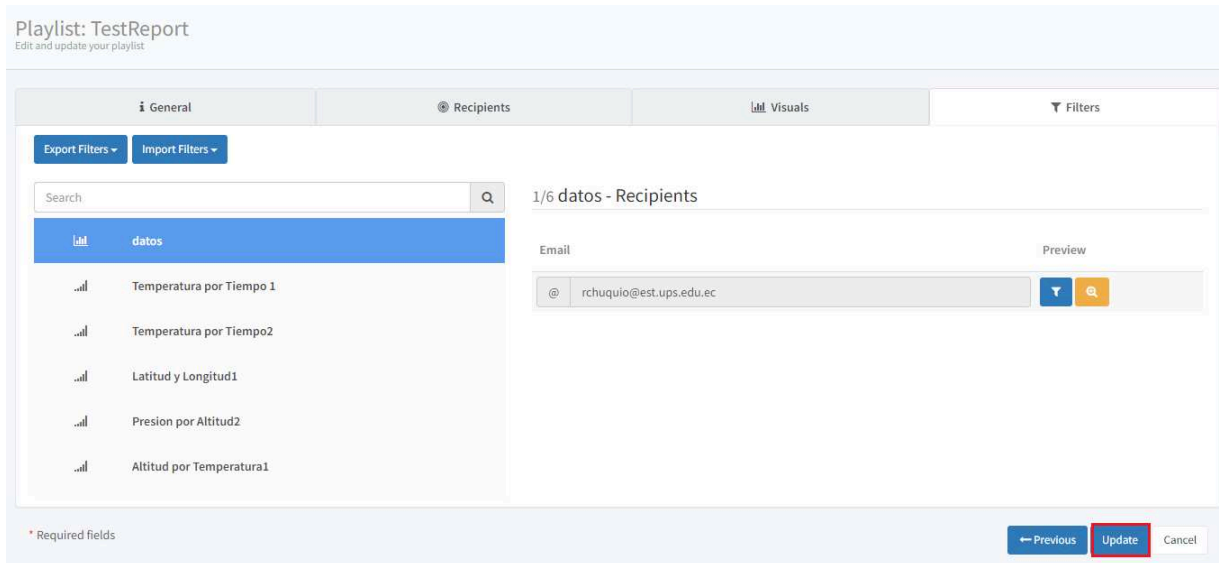


Figura B-10.: Crear Reporte en PowerBI Robots.

- Para poder generar el Reporte se debe de sincronizar el PowerBI Robots Agent tal como se muestra en fig.B-11, luego nos dirigimos a **TestPlaylistExecution** se nos debe de visualizar el Reporte creado anteriormente y la ruta en donde se va a guardar el reporte.

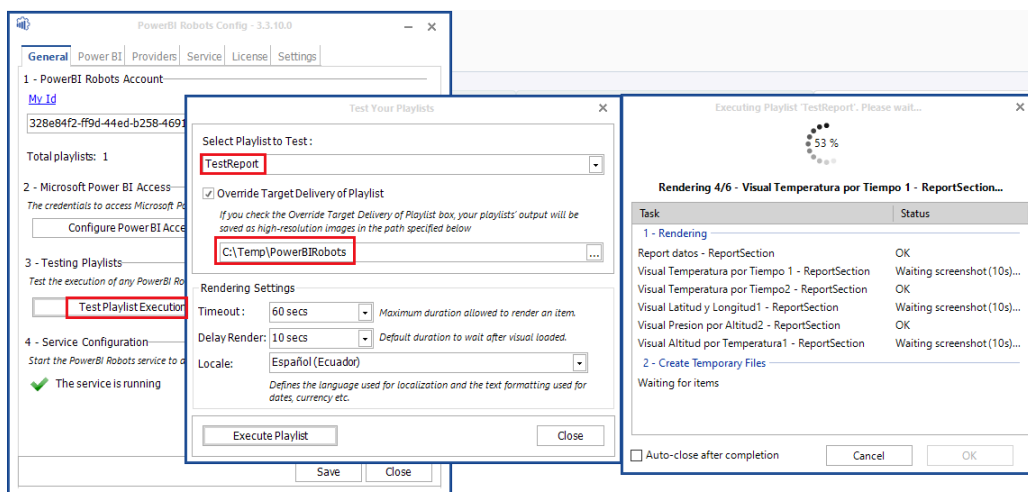


Figura B-11.: Ejecución de TestPlaylistExecution.

- Por ultimo nos dirigimos a la ruta en donde se debe de observar diferentes carpetas creadas con los reportes generados con PowerBI Robots tal como se ilustra en la fig.B-12.

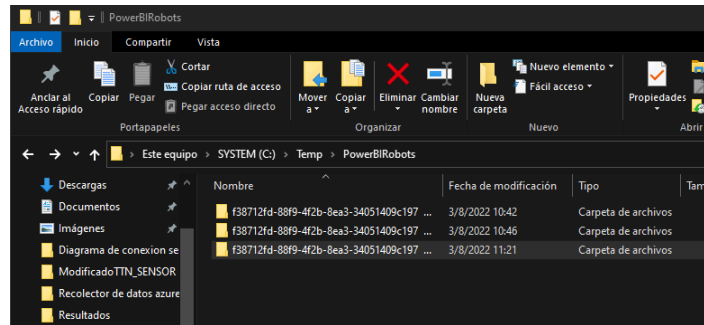
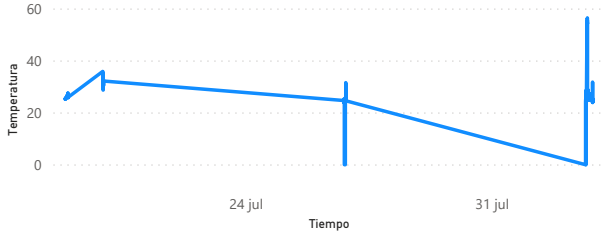


Figura B-12.: Carpetas de Reportes.

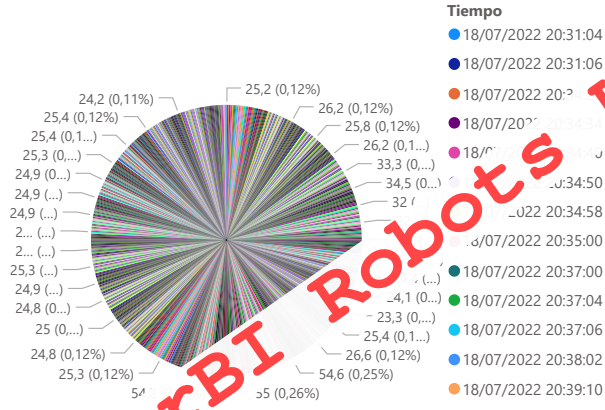
13. Como resultado del reporte generado con PowerBI Robots tenemos los siguientes resultados que se muestran a continuación.



Temperatura por Tiempo



Temperatura por Tiempo

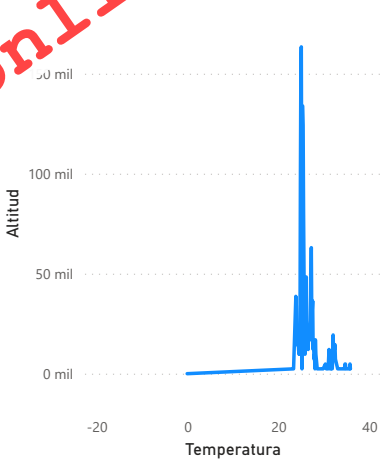


- 18/07/2022 20:31:04
- 18/07/2022 20:31:06
- 18/07/2022 20:31:08
- 18/07/2022 20:31:10
- 18/07/2022 20:31:12
- 18/07/2022 20:31:14
- 18/07/2022 20:31:16
- 18/07/2022 20:31:18
- 18/07/2022 20:31:20
- 18/07/2022 20:31:22
- 18/07/2022 20:31:24
- 18/07/2022 20:31:26
- 18/07/2022 20:31:28
- 18/07/2022 20:31:30
- 18/07/2022 20:31:32
- 18/07/2022 20:31:34
- 18/07/2022 20:31:36
- 18/07/2022 20:31:38
- 18/07/2022 20:31:40
- 18/07/2022 20:31:42
- 18/07/2022 20:31:44
- 18/07/2022 20:31:46
- 18/07/2022 20:31:48
- 18/07/2022 20:31:50
- 18/07/2022 20:31:52
- 18/07/2022 20:31:54
- 18/07/2022 20:31:56
- 18/07/2022 20:31:58
- 18/07/2022 20:32:00
- 18/07/2022 20:32:02
- 18/07/2022 20:32:04
- 18/07/2022 20:32:06
- 18/07/2022 20:32:08
- 18/07/2022 20:32:10
- 18/07/2022 20:32:12
- 18/07/2022 20:32:14
- 18/07/2022 20:32:16
- 18/07/2022 20:32:18
- 18/07/2022 20:32:20
- 18/07/2022 20:32:22
- 18/07/2022 20:32:24
- 18/07/2022 20:32:26
- 18/07/2022 20:32:28
- 18/07/2022 20:32:30
- 18/07/2022 20:32:32
- 18/07/2022 20:32:34
- 18/07/2022 20:32:36
- 18/07/2022 20:32:38
- 18/07/2022 20:32:40
- 18/07/2022 20:32:42
- 18/07/2022 20:32:44
- 18/07/2022 20:32:46
- 18/07/2022 20:32:48
- 18/07/2022 20:32:50
- 18/07/2022 20:32:52
- 18/07/2022 20:32:54
- 18/07/2022 20:32:56
- 18/07/2022 20:32:58
- 18/07/2022 20:33:00

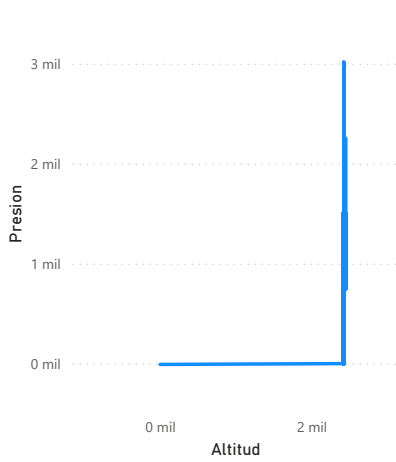
Latitud y Longitud



Altitud y Temperatura

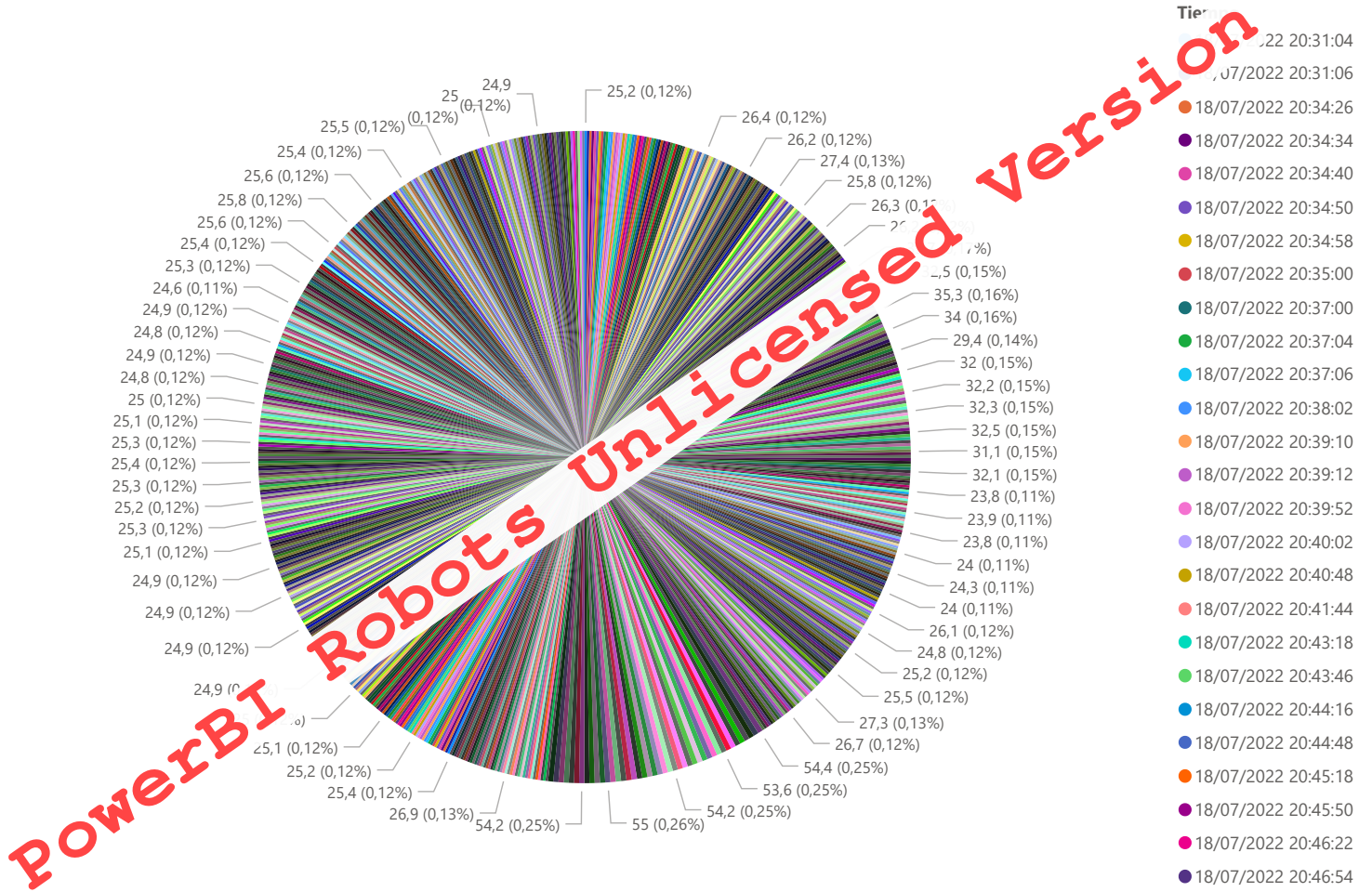


Presion por Altitud

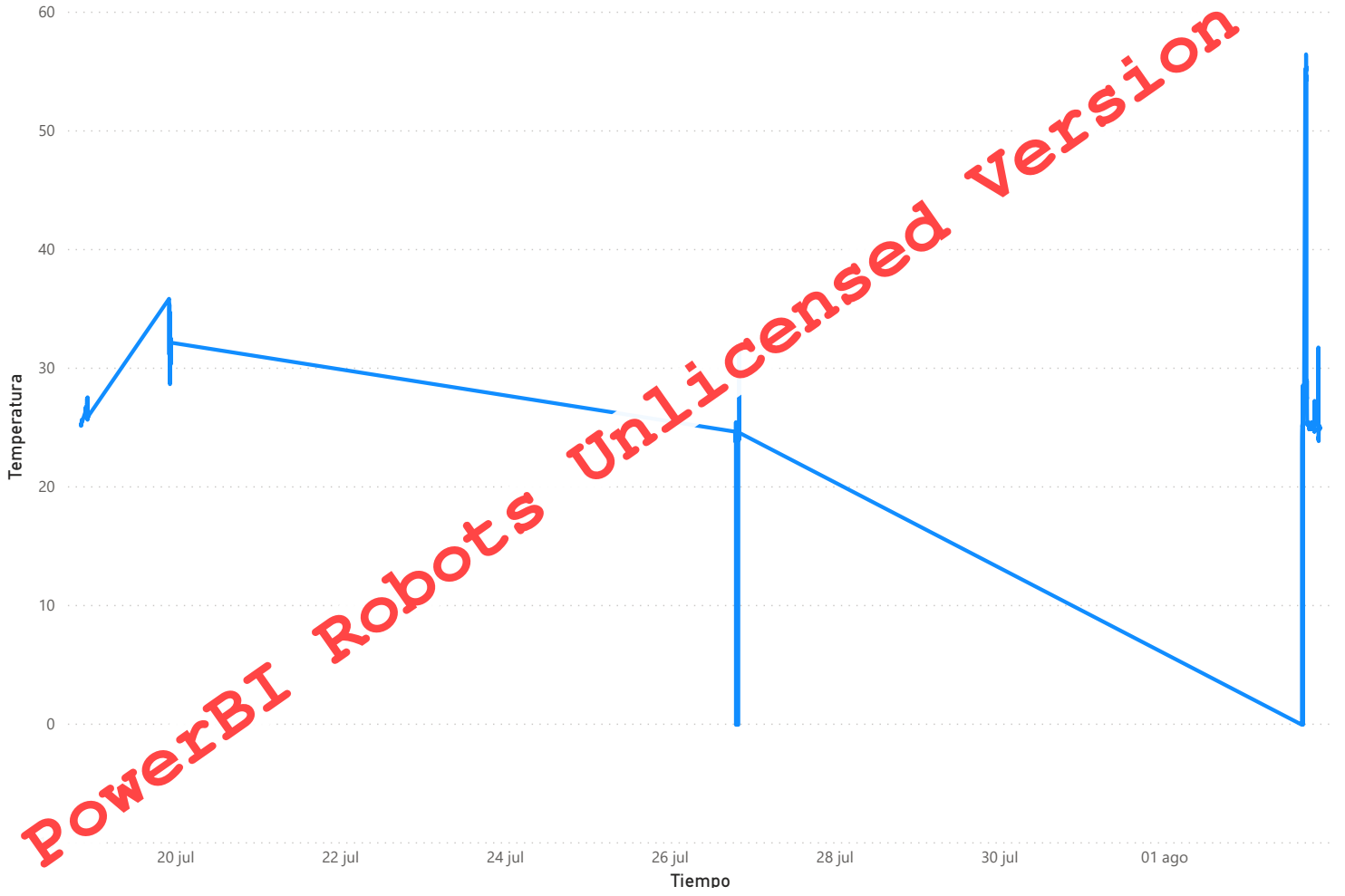


PowerBI Robots Unlicensed Version

Temperatura por Tiempo

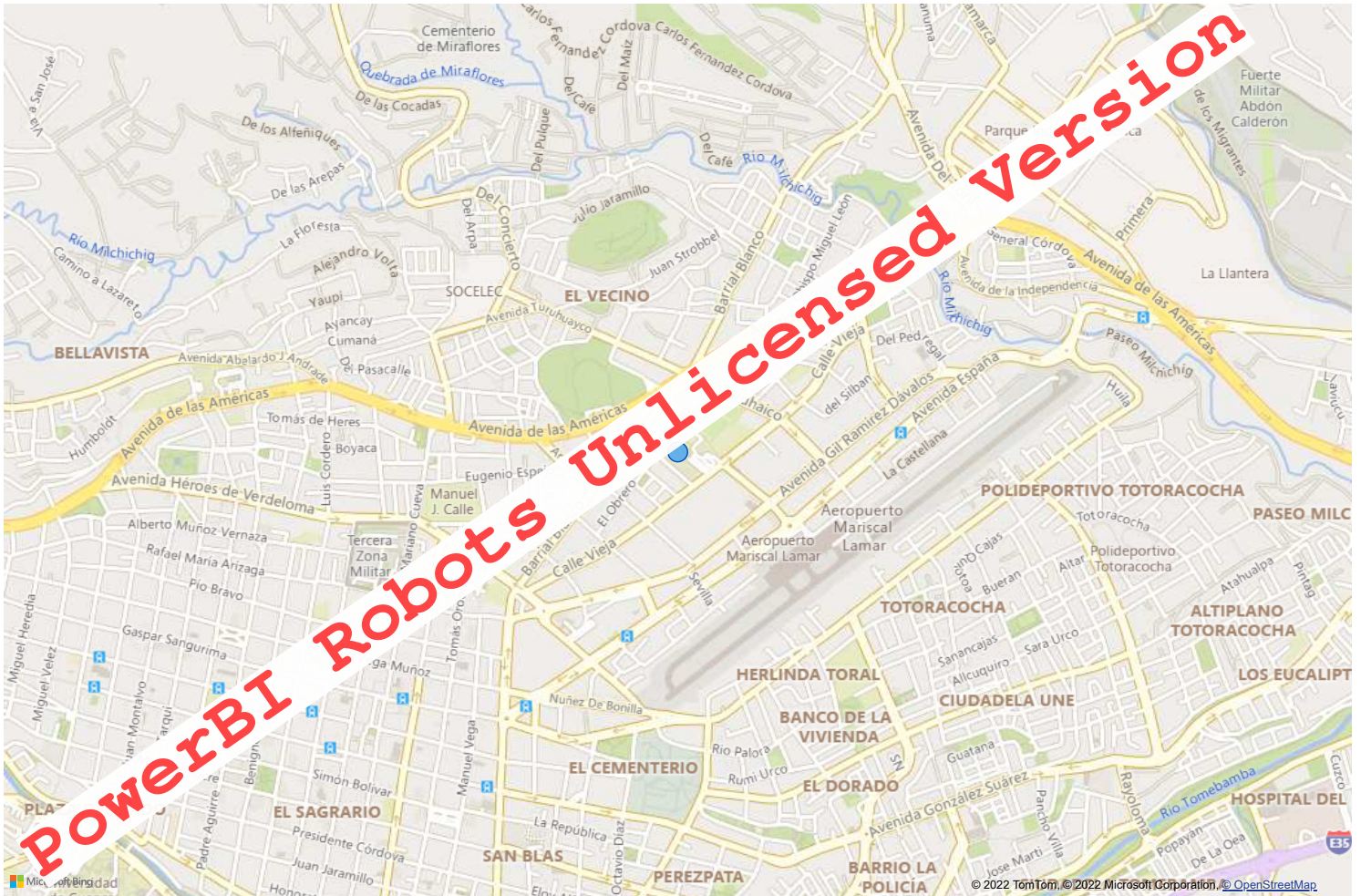


Temperatura por Tiempo

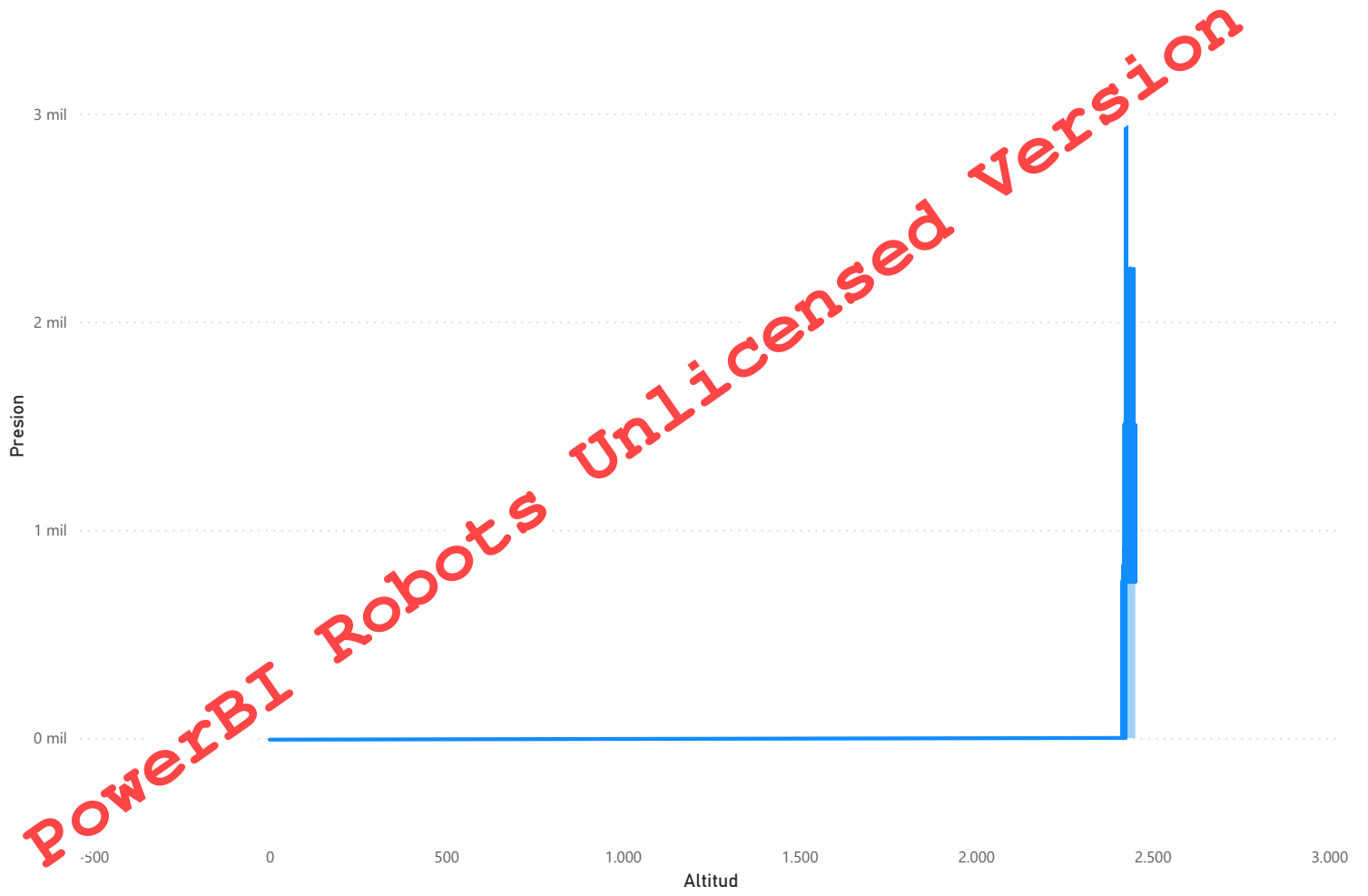




Latitud y Longitud



Presion por Altitud



Altitud por Temperatura

