

INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO

---

Ecosistema de IoT para integración de proyectos de aula utilizando herramientas de código abierto

---

*Autor:*  
Nicolás Martínez Marín

*Director:*  
Gabriel Eduardo Ávila Buitrago

Trabajo presentado como requisito para optar por el título de Ingeniero en  
Telecomunicaciones

MODALIDAD. TRABAJO DE GRADO

Escuela Tecnologías de la Información y las comunicaciones

Bogotá D.C

Colombia

11 de Julio de 2023

## *Declaración de Autoría*

Yo, Nicolás Martínez Marín, declaro que el documento de grado titulado <<Ecosistema de IoT para integración de proyectos de aula utilizando herramientas de código abierto>> y todo el trabajo en este es de mi autoría. Asimismo confirmo que he seguido los lineamientos del reglamento académico y disciplinario de la Institución Universitaria Politécnico Grancolombiano. En especial el artículo 86. Deberes de los estudiantes y el artículo 89. Faltas contra el orden académico.

Firma: Nicolás Martínez

Fecha: 11 de Julio de 2023

# Resumen

Este documento incluye las actividades realizadas para la selección y despliegue de una plataforma de Internet de las Cosas (IoT, por sus siglas en inglés) en el Politécnico Grancolombiano. Se presenta la metodología propuesta para el desarrollo del proyecto, así como un marco conceptual que permita comprender algunos de los términos y tecnologías trabajados en el documento, particularmente respecto a lo correspondiente a IoT. Posteriormente se describen las actividades de diseño y despliegue de la arquitectura para apoyar proyectos de aula que usen tecnologías de IoT. Finalmente, se incluye una descripción de las actividades realizadas con estudiantes que usaron la plataforma seleccionada desplegada en las instalaciones del Politécnico Grancolombiano.

## Palabras Clave

Open Source, Internet de las cosas, Software, Hardware, Redes, Internet, Interconexión, Comunicación, Automatización, Tecnología, Protocolos, Dispositivos, Conectividad, Interoperabilidad, Sensores.

## Abstract

This document encompasses the activities conducted for the selection and implementation of an Internet of Things (IoT) platform at Politécnico Grancolombiano. It outlines the proposed methodology for project development and provides a conceptual framework to facilitate comprehension of the terms and technologies explored within the document, specifically pertaining to IoT. Furthermore, it elucidates the design and deployment activities involved in establishing an architecture to support classroom projects utilizing IoT technologies. Lastly, it includes a description of the activities undertaken with students who utilized the chosen platform deployed on the premises of Politécnico Grancolombiano.

## Keywords

Open Souce, Internet of things, Software, Hardware, Networks, Internet, Interconection, Communication, Automation, Technology, Protocols, Devices, Connectivity, Interoperability, Sensors.

---

Datos generales del proyecto de grado

---

Departamento Académico: Facultad de Ingeniería, Diseño e Innovación. Escuela TIC

Nombre completo estudiante: Nicolás Martínez Marín

Código estudiante: 1410010183

Cédula No: 1020822123

Correo electrónico: [nimartinezma@poligran.edu.co](mailto:nimartinezma@poligran.edu.co)

Fecha inicio: 29/03/2023

Fecha finalización: 11/07/2023

Docente: Gabriel Eduardo Ávila Buitrago

Título del proyecto: Ecosistema de IoT para integración de proyectos de aula utilizando herramientas de código abierto

Área de conocimiento: Telecomunicaciones

---

<b>RESUMEN.....</b>	<b>3</b>
<b>CAPÍTULO 1.....</b>	<b>9</b>
1. ANTECEDENTES.....	9
1.1. <i>Justificación</i> .....	9
1.2. <i>Alcance</i> .....	9
1.3. <i>Objetivo general</i> .....	10
1.3.1. <i>Objetivos específicos</i> .....	10
<b>CAPITULO 2.....</b>	<b>11</b>
2. MARCO METODOLÓGICO.....	11
2.1. <i>Fase de análisis</i> .....	11
2.2. <i>Fase de diseño</i> .....	11
2.3. <i>Fase de despliegue</i> .....	11
2.4. <i>Fase de verificación</i> .....	12
<b>CAPITULO 3.....</b>	<b>13</b>
3. MARCO CONCEPTUAL.....	13
3.1. <i>IoT</i> .....	13
3.2. <i>Funcionamiento del IoT</i> .....	14
3.2.1. <i>Recopilación de datos:</i> .....	14
3.2.2. <i>Comunicación</i> .....	14
3.2.3. <i>Procesamiento y análisis</i> .....	14
3.2.4. <i>Toma de decisiones y acciones</i> .....	14
3.3. <i>Protocolos de comunicación</i> .....	15
3.3.1. <i>HTTP (Hypertext Transfer Protocol)</i> .....	15
3.3.2. <i>Web Socket</i> .....	16
3.3.3. <i>CoAP (Constrained Application Protocol)</i> .....	16
3.3.4. <i>MQTT (Message Queuing Telemetry Transport)</i> .....	17
<b>CAPÍTULO 4.....</b>	<b>18</b>
4. DESARROLLO DEL PROYECTO.....	18
4.1. <i>Selección de tecnologías</i> .....	18
4.1.1. <i>Revisión de proyectos</i> .....	18
4.1.2. <i>Análisis de plataformas IoT</i> .....	19
4.1.2.1. <i>Arduino cloud</i> .....	20
4.1.2.2. <i>Hive mq</i> .....	20
4.1.2.3. <i>Home Assistant</i> .....	21
4.1.2.4. <i>Kaa</i> .....	21
4.1.2.5. <i>Node-red</i> .....	21
4.1.2.6. <i>OpenHAB</i> .....	22
4.1.2.7. <i>ThingSpeak</i> .....	22
4.1.2.8. <i>Thingsboard</i> .....	22
4.1.3. <i>Selección plataforma IoT</i> .....	22
4.1.4. <i>Selección herramientas de hardware IoT</i> .....	23
4.2. <i>Diseño del ecosistema IoT</i> .....	23
4.3. <i>Fase de despliegue</i> .....	28
4.4. <i>Fase de verificación</i> .....	29
4.4.1. <i>Semana TIC</i> .....	29
4.4.2. <i>Proyectos de aula Sistemas Digitales</i> .....	30
4.4.3. <i>Resultados Encuesta conocimientos previos</i> .....	33
<b>CAPÍTULO 5.....</b>	<b>34</b>
5. <i>Conclusiones y trabajo futuro</i> .....	34
5.1. <i>Conclusiones</i> .....	34
5.2. <i>Trabajo futuro</i> .....	34

5.3.	BIBLIOGRAFÍA .....	35
<b>CAPÍTULO 6.....</b>		<b>37</b>
6.	ANEXOS.....	37
6.1.1.	<i>Guía instalación Thingsboard .....</i>	37
6.1.1.1.	<i>Instalación Docker sobre Ubuntu .....</i>	37
6.1.1.2.	<i>Instalación en equipo local .....</i>	37
6.1.1.3.	<i>Instalación en Google Cloud.....</i>	39
6.1.1.4.	<i>Instalación en raspberry Pi 4.....</i>	41
6.2.	<i>Experimentos locales .....</i>	44
6.3.	<i>Información proporcionada.....</i>	49
6.4.	<i>Entrevista grupal estudiantes.....</i>	51
6.5.	<i>Formulario conocimientos previos IoT.....</i>	52
6.6.	<i>Formulario encuesta de satisfacción .....</i>	53
6.7.	<i>Acceso para docentes.....</i>	54
6.8.	<i>Manual funcionalidades Thingsboard .....</i>	57

Ilustración 1.Figura comunicación HTTP .....	15
Ilustración 2.Comunicación WebSocket .....	16
Ilustración 3. Comunicación COAP .....	16
Ilustración 4. Servidor MQTT .....	17
Ilustración 5. Clasificación Plataformas IoT .....	23
Ilustración 6. Comunicación esperada de las tecnologías escogidas.....	24
Ilustración 7. Arquitectura Thingsboard(Imagen tomada de <a href="https://thingsboard.io">https://thingsboard.io</a> ) .....	25
Ilustración 8.Despliegues planteados para el desarrollo del proyecto.....	27
Ilustración 9.Programación de charla Thingsboard tomado de la programación semana TIC .....	29
Ilustración 10.Participación de la semana TIC .....	29
Ilustración 11.Panel de Control Thingsboard mostrado en semana TIC .....	30
Ilustración 12.Tablero proyecto Sumo .....	31
Ilustración 13. Tablero proyecto PetFeeder.....	32
Ilustración 14.Proyecto estación del clima .....	32
Ilustración 15. Verificación de instalación de Docker en Ubuntu.....	37
Ilustración 16.Instalación exitosa en Ubuntu .....	38
Ilustración 17. Estimación Ubuntu en Google Cloud.....	39
Ilustración 18.Reglas de firewall en GCP .....	40
Ilustración 19. Despliegue en GCP.....	41
Ilustración 20. Verificación instalación OpenJDK 11 .....	41
Ilustración 21. Servicio activo Thingsboard en raspberry Pi 4.....	42
Ilustración 22. Reglas de puertos para el despliegue en la Universidad.....	42
Ilustración 23. Despliegue en raspberry Pi 4.....	43
Ilustración 24.Dispositivos seleccionados en Thingsboard.....	44
Ilustración 25. Token de acceso para el dispositivo .....	44
Ilustración 26.Conexión a tópico MQTT .....	46
Ilustración 27.Atributos registrados en Thingsboard .....	47
Ilustración 28. Creación de tablero Thingsboard.....	47
Ilustración 29. Tablero de temperatura simulada .....	48
Ilustración 30.Panel de administrador Thingsboard .....	54
Ilustración 31.Creación de propietario Thingsboard .....	55
Ilustración 32. Agregar administradores de un propietario .....	56
Ilustración 33. Creación de usuario que administra un propietario .....	56
Ilustración 34.Panel de un propietario .....	57
Ilustración 35. Relación de entidades Manual.....	58
Ilustración 36.Manual, creación de activos .....	59
Ilustración 37.Manual, asociar activos .....	59
Ilustración 38.Manual, crear dispositivos.....	59
Ilustración 39.Manual, asociar dispositivos .....	60
Ilustración 40.Manual, crear atributos de servidor.....	60
Ilustración 41.Manual, crear tablero.....	60
Ilustración 42.Manual, crear Alias activos .....	61
Ilustración 43.Manual, crear mapa por Alias .....	61
Ilustración 44.Manual, creación lista de bloques campus principal .....	62
Ilustración 45.Manual, creación sub-estado por bloques del campus .....	62
Ilustración 46.Manual, crear acción para sub-estado de dispositivos.....	63

Ilustración 47.Manual, crear Alias para dispositivos de los bloques.....	63
Ilustración 48.Manual, creación de dispositivos asociados a un bloque .....	64
Ilustración 49.Manual, Alias para el dispositivo final.....	64
Ilustración 50.Manual, acción para navegar a un panel del detalle de los sensores .....	65
Ilustración 51.Manual, graficas específicas de los dispositivos embebidos.....	65
Video 1.Muestra para el uso de Thingsboard en GCP .....	49



# Capítulo 1

## 1. Antecedentes

### 1.1. Justificación

En la Escuela de Tecnologías de la Información y las Comunicaciones (TIC) particularmente para los programas de Ingeniería de Sistemas, Ingeniería de Software e Ingeniería en Telecomunicaciones se desarrollan proyectos relacionados con Internet de las Cosas (IoT, por su sigla en inglés), en asignaturas como Sistemas Digitales y Ensambladores, Señales y Sistemas, Arquitectura de Computadores e Internet de las Cosas. El alcance de estos proyectos se ve limitado por la falta de uso de estándares de IoT en las asignaturas y la no existencia de un ecosistema adecuado que permita agilizar los desarrollos. Por esta razón, se dificulta llegar a etapas avanzadas de visualización y análisis de datos, debido a los reprocesos comunes en etapas iniciales.

Mediante este proyecto de grado se pretende diseñar e implementar una arquitectura que permita interactuar con dispositivos IoT para la transmisión, acceso y visualización de datos en diferentes espacios académicos, enfocado especialmente en los proyectos de aula y prácticas de laboratorio.

En la convocatoria de proyectos de semilleros de 2023-1 se propuso el proyecto de investigación llamado: “Uso de herramientas open source de IoT como Estrategia didáctica para la Escuela TIC”. Este trabajo de grado hace parte de una primera fase con el fin de suplir aspectos técnicos necesarios para la implementación y evaluación de las estrategias didácticas.

### 1.2. Alcance

Se pretende proveer una solución de Internet de las Cosas que permita la conexión de diferentes dispositivos embebidos, la transmisión de datos, su almacenamiento y visualización, para proyectos en la Institución Universitaria Politécnico Grancolombiano, con el fin de desarrollar proyectos con un mayor alcance, de una manera ágil y manteniendo un histórico de lo trabajado en los diferentes semestres.

La solución desplegada permitirá interconectar proyectos realizados en asignaturas seleccionadas que se estén cursando en el primer semestre del año 2023.

### 1.3. Objetivo general

Proveer un ecosistema de Internet de las Cosas que permita conectar diversos dispositivos embebidos utilizando protocolos de comunicación y plataformas de código abierto, cumpliendo con estándares de la industria.

#### 1.3.1. Objetivos específicos

- Definir las tecnologías de IoT a utilizar de acuerdo con una búsqueda de información enfocada en proyectos similares.
- Diseñar un ecosistema para el desarrollo de proyectos de IoT basado en estándares, tecnologías y referentes seleccionados.
- Desplegar el conjunto de herramientas seleccionadas bajo el ecosistema diseñado que permita una adecuada conectividad de dispositivos embebidos.
- Verificar el funcionamiento de las herramientas desplegada mediante casos de uso típicos de IoT.

# Capítulo 2

## 2. Marco metodológico

Para el desarrollo de este proyecto se plantea una metodología en cascada para poder lograr los objetivos planteados, la metodología es escogida dada la naturaleza del proyecto donde se pueden identificar fases muy marcadas que se deben ejecutar de forma secuencial, a continuación se van a explicar las fases planeadas y la forma en la que se va a ejecutar cada una.

### 2.1. Fase de análisis

Para definir las tecnologías a utilizar primero se hace una búsqueda de información para establecer los conceptos claves de IoT, adicionalmente tener claro qué se considera como buenas prácticas o estándares dentro de esta industria.

Después se hace una revisión de algunos trabajos previos que involucran implementaciones para cubrir casos de uso en el campo de IoT utilizando herramientas de código abierto, posteriormente se hace una selección de plataformas de IoT para revisar sus características y funcionalidades con el propósito de compararlas y seleccionar las herramientas que más se adaptan a las necesidades del proyecto.

### 2.2. Fase de diseño

En esta fase se busca mostrar el enfoque planteado para hacer uso de las tecnologías seleccionadas, se muestra como con la selección hecha se podría adoptar un ecosistema IoT dentro de la institución, sirviendo de apoyo para el desarrollo de proyectos de aula de las asignaturas con temáticas asociadas.

### 2.3. Fase de despliegue

Una vez definido el diseño se procede a hacer despliegue de la solución planteada para poder conectar múltiples dispositivos embebidos que permiten cubrir casos de uso utilizando IoT.

Para lograr esto se hacen experimentos que validan el correcto funcionamiento de las herramientas elegidas y se procede a darle acceso a un grupo de estudiantes seleccionados para que utilicen las herramientas dentro del desarrollo de sus proyectos de aula.

## 2.4. Fase de verificación

Como este proyecto está enmarcado dentro de la iniciativa “Uso de herramientas open source de IoT como Estrategia didáctica para la Escuela Tic”, se plantea evaluar la acogida del mismo por la comunidad estudiantil, para esto se busca un espacio en la Semana TIC organizada por la Facultad de Ingeniería, Diseño e Innovación – FIDI.

Con esta participación, se le da más visibilidad al proyecto y se plantean encuestas que tienen como objetivo obtener retroalimentación sobre el mismo.

Adicionalmente se plantea una entrevista grupal al curso que tuvo acceso al despliegue realizado, con esta entrevista se busca evaluar la efectividad de la solución con la percepción de los estudiantes y el resultado obtenido de los proyectos de aula.

# Capítulo 3

## 3. Marco conceptual

El presente marco conceptual busca dejar claro qué es el internet de las cosas , cómo es su funcionamiento, resaltar su importancia y además explicar algunos protocolos de comunicación que se suelen utilizar cuando se trabaja con IoT.

### 3.1. IoT

El Internet de las cosas es un concepto que se refiere a la interconexión de objetos físicos a través de internet, permitiendo la comunicación y el intercambio de datos entre ellos sin la intervención humana. Estos objetos, también conocidos como dispositivos o sensores, están equipados con tecnología y capacidad de conexión a internet, lo que les permite recopilar, transmitir y recibir información, consultando estándares sobre la buena implementación de un sistema IoT se puede ver que el uso resalta en múltiples áreas, se puede ver un ejemplo a continuación ("IEEE Standard for an Architectural Framework for the Internet of Things (IoT)," in IEEE Std 2413-2019 , vol., no., pp.1-269, 10 March 2020, doi: 10.1109/IEEESTD.2020.9032420., 2020).

Al conectar objetos físicos a Internet, se puede automatizar tareas y procesos, lo que conduce a una mejora en la eficiencia y productividad. Por ejemplo, puede utilizarse para monitorear y controlar equipos industriales, lo que ayuda a reducir los tiempos de inactividad y aumentar la producción.

Puede mejorar la seguridad, por ejemplo, puede utilizarse para rastrear la ubicación de vehículos y equipos en tiempo real, ayudando así a prevenir accidentes y robos.

También destaca la importancia de utilizar esta tecnología para abordar algunos de los desafíos más importantes del mundo, como el cambio climático, la pobreza y la desigualdad. Por ejemplo, puede utilizarse para monitorear y rastrear el medio ambiente, lo que ayuda a reducir nuestro impacto en el planeta. También puede utilizarse para proporcionar acceso a educación y atención médica en países en desarrollo, lo que contribuye a reducir la pobreza y la desigualdad.

Es una tecnología que ha ganado mucha importancia en los últimos años y cada vez se empieza adoptar más trabajar con este tipo de sistemas, esto hace que sea importante adquirir conocimiento sobre el uso, funcionamiento y posibles implementaciones que se pueden dar sobre esta tecnología.

## 3.2. Funcionamiento del IoT

El funcionamiento básico del IoT implica los siguientes pasos:

### 3.2.1. Recopilación de datos:

Los dispositivos IoT recopilan datos a través de sensores integrados, como sensores de temperatura, acelerómetros, cámaras, entre otros. Estos sensores capturan información sobre el entorno o el objeto al que están conectados.

### 3.2.2. Comunicación

Los datos recopilados son transmitidos a través de redes de comunicación, utilizando diferentes protocolos, esto depende de los requisitos de conectividad, los dispositivos y la distancia de transmisión.

### 3.2.3. Procesamiento y análisis

Los datos recibidos se procesan y analizan en tiempo real o se envían a servidores remotos para su procesamiento. Aquí se aplican algoritmos y técnicas de análisis de datos para extraer información relevante.

### 3.2.4. Toma de decisiones y acciones

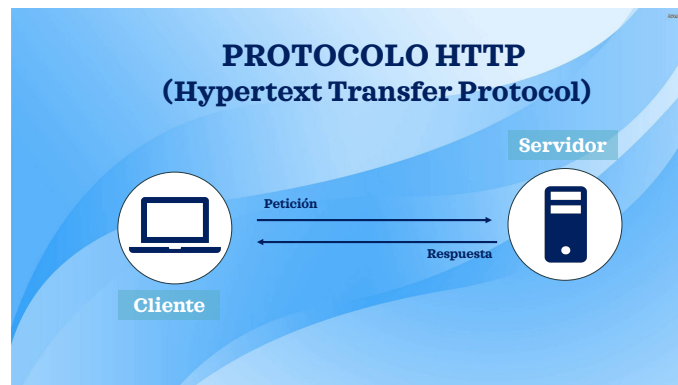
Con base en el análisis de datos, se pueden tomar decisiones automáticas o proporcionar información a los usuarios para que tomen acciones. Esto puede implicar activar o desactivar dispositivos, ajustar parámetros, enviar alertas, entre otros.

### 3.3. Protocolos de comunicación

Los protocolos son fundamentales para la comunicación efectiva entre los dispositivos IoT. Algunos protocolos utilizados en tecnologías de IoT son:

#### 3.3.1. HTTP (Hypertext Transfer Protocol)

Es un protocolo de comunicación ampliamente utilizado y se puede considerar el más popular para la comunicación web. Permite la transferencia de datos entre un cliente y un servidor de forma segura y confiable.



*Ilustración 1. Figura comunicación HTTP*

En el contexto de IoT, HTTP se utiliza para enviar y recibir datos entre dispositivos IoT y servidores basados en la nube, así como para controlar dispositivos de forma remota. HTTP es un protocolo versátil, confiable y seguro que se puede adaptar a diferentes dispositivos embebidos y ofrece ventajas como su amplia disponibilidad de herramientas y bibliotecas. Sin embargo, puede ser ineficiente para gran volúmenes de datos y dificultar su implementación en algunos dispositivos que no cuenten con capacidades de procesamiento mínimas para utilizar este protocolo, HTTP es una buena elección para aplicaciones de IoT que requieren comunicación confiable y segura, aunque se deben considerar sus limitaciones antes de su implementación.

### 3.3.2. Web Socket

Es un protocolo de comunicación bidireccional que permite una conexión persistente y en tiempo real entre un cliente y un servidor a través de un único socket, está basado sobre TCP lo que garantiza que es un protocolo confiable y seguro. En el contexto de IoT es utilizado para aplicaciones donde se esté recibiendo y enviando información en tiempo real, no es tan ligero como otros protocolos por lo que usualmente los componentes que suelen utilizarlo son de visualización de cara a las personas, esto porque estos equipos tienen mayores capacidades de procesamiento.



Ilustración 2. Comunicación WebSocket

### 3.3.3. CoAP (Constrained Application Protocol)

Es un protocolo de aplicación diseñado para dispositivos con recursos limitados, como sensores y actuadores. CoAP permite la comunicación eficiente en redes de baja potencia y baja capacidad de almacenamiento, es una implementación más ligera del protocolo HTTP que busca un comportamiento similar al MQTT.

Este protocolo puede funcionar en casos de uso cuando se tienen limitaciones técnicas, pero es importante tener en cuenta que no es tan compatible como otros protocolos y hay menos librerías disponibles para hacer uso del mismo.



Ilustración 3. Comunicación COAP



### 3.3.4. MQTT (Message Queuing Telemetry Transport)

Es un protocolo de mensajería ligero y eficiente diseñado para entornos de IoT. MQTT sigue un modelo de publicación/suscripción y se basa en un mecanismo de envío y recepción de mensajes, lo que significa que hay dos componentes principales involucrados. (MQTT Sparkplug solution, s.f.)

- Clientes: Estos son los dispositivos que publican o se suscriben a mensajes.
- Brokers: Estos son los servidores que almacenan y reenvían los mensajes.

Es el protocolo más utilizado en sistemas de IoT gracias a lo pequeños que son sus mensajes, además facilita el uso del mismo en dispositivos con recursos limitados.

Está diseñado para la comunicación entre maquina a máquina, lo hace estar muy alineado con IoT que busca interconectar múltiples dispositivos también. (MQTT Sparkplug solution, s.f.)

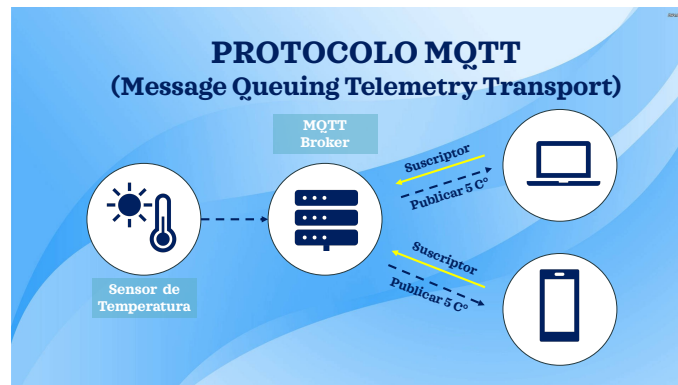


Ilustración 4. Servidor MQTT

# Capítulo 4

## 4. Desarrollo del proyecto

### 4.1. Selección de tecnologías

#### 4.1.1. Revisión de proyectos

Lo primero que se hace para la selección de las tecnologías es una revisión de proyectos donde se implementen sistemas de IoT para solucionar algún caso de uso, a continuación se muestran los trabajos consultados.

En (Tyutyundzhiev, y otros, 2023) los autores describen como utilizar el microcontrolador ESP32 y el protocolo MQTT para recolectar de forma remota información de un sensor UV, se resalta el uso de tecnologías de código abierto como ESP32, Mosquitto MQTT bróker y Grafana para facilitar y bajar los costos de construcción de sistemas que sean capaces de recolectar y visualizar datos.

En el artículo (Manchanayaka, y otros, 2021) se describe el desarrollo de un sistema de monitoreo de la contaminación del aire utilizando herramientas de código abierto, se utiliza un circuito impreso personalizado (PCB por sus siglas en inglés) junto con el microcontrolador Arduino Mega, al que se le conecta la placa NodeMCU para tener conexión Wi-Fi y sensores de gases para detectar niveles de contaminación en el aire. El Sistema transmite los datos usando GPRS y los muestra en la herramienta open-source “slpiot”, en el artículo se especifica que el sistema cumple con el estándar IEEE 1451 para integrarse fácilmente con otros sistemas y que los datos que genera sean interoperables. Los autores resaltan que al hacer uso de tecnologías de código abierto el sistema se vuelve más accesible y flexible, también resaltan el uso de estándares abiertos con el fin de facilitar la interoperabilidad del sistema.

En (Okhovat & Bauer, 2021) los autores explican cómo se pueden utilizar las herramientas de código abierto Thingsboard y Node-RED para monitorear y visualizar datos de sensores en una ciudad inteligente. El artículo destaca la escalabilidad, personalización y seguridad de estas herramientas de código abierto, afirmando que son una solución rentable y efectiva para el monitoreo de grandes cantidades de datos de sensores, las herramientas son elegidas después de hacer una revisión de múltiples plataformas IoT, los autores del artículo utilizaron Thingsboard para crear un panel de control que muestra datos de sensores de diversas fuentes, como sensores de temperatura, sensores de humedad y sensores de calidad del aire. También utilizaron Node-RED para crear flujos que envían datos de sensores a Thingsboard y activan alertas en función a ciertas reglas configuradas que se aplican a los mensajes recibidos.

Las tecnologías de código abierto desempeñan un papel fundamental en el desarrollo de sistemas IoT, como se ha destacado en los artículos mencionados. La disponibilidad de hardware y microcontroladores de código abierto y bajo costo, como el ESP32, NodeMCU y Arduino Mega, permite la creación de soluciones accesibles y flexibles. Además, el uso de protocolos de comunicación como MQTT y estándares abiertos como IEEE 1451 garantiza la interoperabilidad y la integración con otros sistemas. Las plataformas como Mosquitto MQTT broker, Grafana, Thingsboard y Node-RED ofrecen herramientas poderosas para el almacenamiento, análisis y visualización de datos de sensores en tiempo real. En conjunto, las tecnologías de código abierto impulsan el desarrollo de sistemas IoT más asequibles, adaptables y escalables, abriendo nuevas posibilidades en el monitoreo y gestión de datos en diferentes aplicaciones.

#### 4.1.2. Análisis de plataformas IoT

Para terminar esta fase se procedió a elegir la mejor plataforma. Para esto se evaluaron 8 plataformas reconocidas por permitir realizar trabajos con IoT, se hace una revisión de las características y funcionalidades de las herramientas seleccionadas para realizar una comparación y proceder a elegir la que mejor se ajuste a las necesidades del proyecto. Para hacer esta comparación se definen 6 parámetros de evaluación. Los parámetros escogidos son los siguientes:

- **Comunidad de usuarios**

Una comunidad activa de usuarios indica que la plataforma tiene un respaldo sólido en términos de soporte, documentación, actualizaciones y mejoras frecuentes. Una comunidad activa también puede proporcionar recursos valiosos, como ejemplos de proyectos, soluciones a problemas comunes y oportunidades de colaboración.

- **Flexibilidad y escalabilidad**

La flexibilidad se refiere a la capacidad de la plataforma para adaptarse a diferentes necesidades y requisitos específicos del proyecto. Esto implica que la plataforma sea capaz de manejar diferentes casos de uso y tamaños de implementación. La escalabilidad se refiere a la capacidad de la plataforma para crecer y manejar cargas de trabajo cada vez mayores a medida que el proyecto se expande.

- **Interoperabilidad**

La interoperabilidad es crucial en el contexto del IoT, ya que implica la capacidad de la plataforma para trabajar con diferentes dispositivos, protocolos y sistemas. Esto permite una integración más fácil y eficiente con el ecosistema IoT existente y futuras expansiones.

- **Facilidad de Uso**

La facilidad de uso es importante para agilizar el desarrollo de proyectos y

maximizar la productividad. Una interfaz intuitiva y una curva de aprendizaje razonable permiten a los usuarios aprovechar al máximo la plataforma sin dificultades excesivas.

- **Seguridad**

La seguridad es un aspecto crítico en IoT, ya que involucra la protección de datos y dispositivos sensibles. Una plataforma sólida debe contar con mecanismos de seguridad robustos, como autenticación, cifrado de datos y control de acceso, para garantizar la integridad y confidencialidad de la información.

- **Costo**

Aunque las herramientas de IoT en sí mismas pueden no tener un costo directo, es importante considerar los costos asociados con el despliegue y mantenimiento, así como cualquier necesidad de utilizar herramientas adicionales o servicios complementarios.

#### 4.1.2.1. [Arduino cloud](#)

Arduino cloud cuenta con una comunidad de usuarios activa debido a su popularidad en el ámbito de la electrónica y la programación, además que cuenta con su propio foro, ofrece flexibilidad para adaptarse a diferentes aplicaciones aunque su enfoque principal son el desarrollo de proyectos con dispositivos Arduino, permite integraciones con otros servicios en la nube, en general Arduino se destaca por su facilidad de uso, especialmente para personas que estén incursionando en el mundo de la electrónica y la programación, la plataforma también ofrece mecanismos de seguridad para garantizar la integridad del proyecto y de su información. Por último la plataforma solo se puede utilizar en forma de suscripción, cuenta con un plan gratuito con bastantes limitaciones y para hacer uso de todas las características sería necesario pagar otro tipo de plan. (Arduino cloud, s.f.)

#### 4.1.2.2. [Hive mq](#)

HiveMQ es una plataforma de mensajería MQTT escalable y confiable utilizada en aplicaciones de IoT y otros escenarios donde se requiere una comunicación eficiente entre dispositivos. Proporciona un servidor MQTT altamente escalable, con alta disponibilidad y seguridad integrada, lo que permite el intercambio de mensajes bidireccional entre dispositivos y aplicaciones de manera eficiente y confiable. Esta herramienta es interoperable dentro del contexto de MQTT ya que cuenta con un robusto desarrollo de clientes en diferentes tecnologías que permiten la integración con esta plataforma, pero no existe forma de utilizar esta herramienta con otros protocolos ya que no es el enfoque de la herramienta, tampoco cuenta con un mecanismo propio de visualización de datos pero tiene la capacidad de integrarse con otras herramientas que estén diseñadas para esto. Cuenta con un foro propio donde se puede encontrar actividad en preguntas y respuestas. (MQTT Sparkplug solution, s.f.)

#### 4.1.2.3. Home Assistant

Esta herramienta de código abierto permite agregar diferentes dispositivos a una casa inteligente, cuenta con una comunidad muy activa lo que refleja su popularidad y soporte continuo, la herramienta se destaca por su interfaz de usuario intuitiva y su enfoque en la facilidad de configuración y personalización, también cuenta con opciones de seguridad para proteger los datos de los dispositivos conectados, permite la instalación de la herramienta en múltiples sistemas operativos para usarla de forma local, sin embargo la herramienta está diseñada para monitorear los dispositivos de una red local, esto limita el uso y además no permite trabajar con múltiples proyectos de IoT. (Home Assistant, s.f.)

#### 4.1.2.4. Kaa

Esta herramienta no tiene una comunidad tan activa de desarrollo, para encontrar soporte se remite a la página web Stack Overflow donde se puede encontrar información de la plataforma pero el último post es del 2021, dos años atrás al momento de elaborar este documento, la plataforma ofrece flexibilidad y escalabilidad, la plataforma también es compatible con gran variedad de protocolos y dispositivos, lo que permite que sea interoperable con otros sistemas, para el uso de la plataforma se requiere tener ciertos conocimientos técnicos aunque la plataforma no es particularmente difícil de usar, ofrece características de seguridad para proteger los datos y los dispositivos conectados, las características mencionadas hacen parte de la primera versión de la plataforma (v0.x.) que era totalmente gratuita pero en la documentación informan que actualmente todas las actualizaciones se hacen sobre una nueva versión de la plataforma orientada a un producto empresarial de pago que solo permite un uso gratuito por 30 días. (Kaa documentation, s.f.)

#### 4.1.2.5. Node-red

Es una plataforma basada en flujos que permite la presentación del comportamiento del sistema o aplicación y está construida sobre el entorno de ejecución node.js. El flujo Node-Red contiene bloques específicos de código llamados nodos, que se conectan entre sí y los datos pasan a través de ellos. Esta herramienta de código abierto es ideal para aplicaciones basadas en eventos y se puede aplicar en una variedad de arquitecturas de sistemas, desde entornos locales hasta entornos en la nube como IBM Cloud, AWS y Microsoft Azure. El enfoque visual de la aplicación hace que la herramienta sea fácil de aprender y utilizar, esto también ha causado que gane popularidad y se puede ver reflejado en la comunidad activa con la que cuenta, también cuenta opciones de seguridad y cifrado pero la configuración y uso adecuado depende totalmente de la implementación, la herramienta es gratuita pero puede tener costos adicionales asociados a la selección y configuración de nodos desarrollados por la comunidad para cumplir necesidad que la plataforma no cumple por defecto, como lo puede ser la visualización y monitoreo de datos. (Node-RED, s.f.)

#### 4.1.2.6. OpenHAB

Es una plataforma de automatización del hogar para casas inteligentes altamente flexible y escalable dentro de este contexto. Cuenta con una comunidad de usuarios activa y comprometida que brinda soporte, documentación y mejoras frecuentes. Su enfoque en la interoperabilidad permite la integración fluida con una amplia gama de dispositivos y servicios a través de diversos protocolos. Aunque requiere cierto nivel de conocimientos técnicos para su personalización y configuración, openHAB ofrece una interfaz intuitiva, características de seguridad como cifrado y control de acceso, es de código abierto y gratuito pero puede tener costos adicionales relacionados al trabajo necesario en la personalización de funcionalidades cuando se quiera implementar de casos de uso que no estén orientados a la automatización de un hogar ya que es el enfoque de la herramienta. (openHab, s.f.)

#### 4.1.2.7. ThingSpeak

ThingSpeak es una plataforma que ofrece una comunidad de usuarios activa y una buena flexibilidad y escalabilidad para adaptarse a diferentes casos de uso en el monitoreo y análisis de datos en tiempo real. Es compatible con varios protocolos y permite la integración con una variedad de dispositivos y servicios. Además, se destaca por su interfaz intuitiva y facilidad de uso para configurar y visualizar datos. Ofrece funcionalidades de protección de datos. El uso de la herramienta solo se puede hacer en la nube y ofrece una opción gratuita con limitaciones, para obtener mayor funcionalidad ofrece suscripciones de pago. (ThingSpeak, 2023)

#### 4.1.2.8. Thingsboard

Thingsboard es una plataforma IoT open-source que ofrece una amplia variedad de funcionalidades para el desarrollo de proyectos IoT. Entre estas capacidades se incluyen la provisión y gestión de sensores y dispositivos reales y virtuales, la recopilación y análisis de datos, la visualización de datos en tiempo real con widgets en paneles de control, la generación de alarmas basadas en controles de flujo, el control de dispositivos e integraciones con sistemas externos. Es compatible con múltiples dispositivos y protocolos lo que permite buena interoperabilidad, ofrece características de seguridad que pueden ser fácilmente configurables. Adicionalmente cuenta con una amplia documentación de todas las funcionalidades con múltiples ejemplos que facilitan el uso de la herramienta, tiene una comunidad bastante activa en el proyecto de Github sacando periódicamente nuevas funcionalidades y dando soporte sobre los incidentes encontrados, se puede desplegar en múltiples sistemas operativos de forma local y también en servicios de la nube. Además de la versión gratuita ofrece opciones de planes comerciales con funcionalidades adicionales y soporte. (ThingsBoard docs, s.f.)

### 4.1.3. Selección plataforma IoT

Con los parámetros de evaluación elegidos se realiza un puntuación siguiendo el criterio propio, donde a cada plataforma se le asigno un puntaje de 1 a 5, 1 significa que no cumple con el parámetro y 5 significa que cumple totalmente.

Plataformas	Comunidad de usuarios	Flexibilidad y escalabilidad	Interoperabilidad	Facilidad de uso	Seguridad	Costo	$\bar{x}$
Arduino cloud	4	4	3	4	4	2	3.5
Hivemq	3	4	3	3	4	3	3.33
Home Assistant	5	2	5	4	4	4	4
Kaa	2	4	4	3	3	1	2.83
Node-Red	4	4	4	4	4	3	3.83
OpenHAB	4	2	5	3	4	3	3.5
ThingSpeak	4	4	4	4	4	2	3.66
Thingsboard	5	4	4	4	4	4	4.16

Ilustración 5. Clasificación Plataformas IoT

Del análisis de las plataformas se concluye que Thingsboard es la opción que mejor se ajusta a las necesidades del proyecto permitiendo trabajar con múltiples protocolos, casos de uso donde se gestionen los dispositivos, permitiendo almacenar, monitorear y visualizar la información enviada.

#### 4.1.4. Selección herramientas de hardware IoT

De acuerdo a los proyectos que se usaron de referencia (Manchanayaka, y otros, 2021) (Tyutyundzhiev, y otros, 2023) (Okhovat & Bauer, 2021), y a los recursos disponibles para este tipo de proyectos, para los componentes de hardware se decidió trabajar con tarjetas de desarrollo de bajo costo como los son NodeMCU y ESP32 debido a su popularidad para hacer implementaciones de IoT.

Además, con Arduino IDE, software elegido para programar las placas ya que es de los programas más populares y por ende se tiende a encontrar más documentación, librerías, ejemplos y soporte.

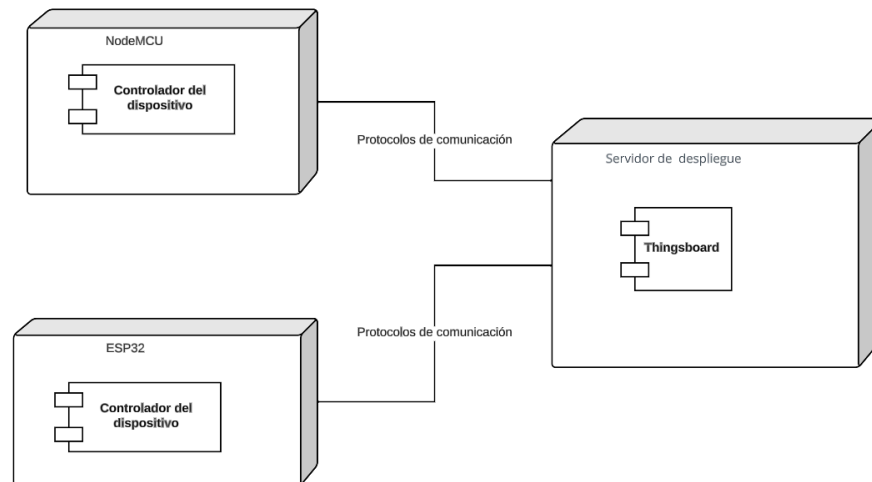
Adicionalmente se decidió como herramienta de despliegue dentro de la institución trabajar con un computador de bajo costo de referencia raspberry pi 4.

#### 4.2. Diseño del ecosistema IoT

Por las tecnologías escogidas el diseño del ecosistema es muy sencillo, consiste en comunicar directamente los sistemas embebidos con la plataforma Thingsboard como se puede ver en la Ilustración 6, la comunicación se lleva a cabo por controladores que se cargan sobre los dispositivos, estos controladores pueden estar elaborados en herramientas de desarrollo como Arduino IDE y utilizar múltiples protocolos de comunicación ya que la plataforma Thingsboard tiene esa interoperabilidad.

A los componentes de hardware pueden estar conectados diversos sensores que recopilen información del entorno en el que se encuentren, se hace la aclaración que se pueden utilizar otras placas de desarrollo a las mostradas en la Ilustración 6, sin embargo, para este proyecto se tiene planteado utilizar estas herramientas.

Dado que la plataforma IoT escogida cuenta de manera compacta con los mecanismos para implementar un ecosistema IoT con buenas prácticas, en esta sección se van a diseñar mecanismos en los cuales se puede poner la herramienta a disposición de los estudiantes.



*Ilustración 6. Comunicación esperada de las tecnologías escogidas.*

Para entender mejor la plataforma IoT seleccionada, a continuación se detalla la arquitectura en la que está construida y las funcionalidades que ofrece .

La arquitectura está compuesta por 4 componentes principales que se pueden ver en la Ilustración 7, están descritos a continuación (Thingsboard arquitectura, s.f.):

- **ThingsBoard Transport** es responsable de recibir los mensajes de los sensores y dispositivos en la red utilizando protocolos de comunicación, analizar los mensajes y enviarlos a las colas de mensajes.
- **ThingsBoard Core** se encarga de manejar las llamadas a la API REST, las suscripciones de WebSocket a la telemetría de las entidades y los cambios de atributos, almacenar información sobre las sesiones de los dispositivos y supervisar el estado de conectividad de los dispositivos.
- **ThingsBoard Rule Engine** El motor de reglas se considera el componente principal del sistema y se suscribe a los mensajes entrantes para procesarlos.
- **ThingsbBoard UI** es el último componente, consiste en la interfaz web que está escrita en Express.js.

Como se mencionó al momento de seleccionar la herramienta, es bastante flexible y permite hacer el despliegue de forma local ya sea directamente en el sistema operativo o en contenedores Docker, adicionalmente permite el despliegue en servicios de nube como



AWS, Azure, Google Cloud entre otros. ThingsBoard ofrece dos tipos de arquitecturas: monolítica y con microservicios, para este proyecto se va a usar la opción monolítica porque requiere un menor costo en términos del mantenimiento sobre la herramienta y el uso de equipos físicos, esta arquitectura es la que se puede ver en la Ilustración 7.

Según la cantidad y el tipo de datos, ofrece diferentes opciones de almacenamiento, como SQL (PostgreSQL, HSQLDB), NoSQL (Cassandra) y un enfoque híbrido que almacena todos los datos relacionados con ThingsBoard en PostgreSQL y datos relacionados a la telemetría en Cassandra.

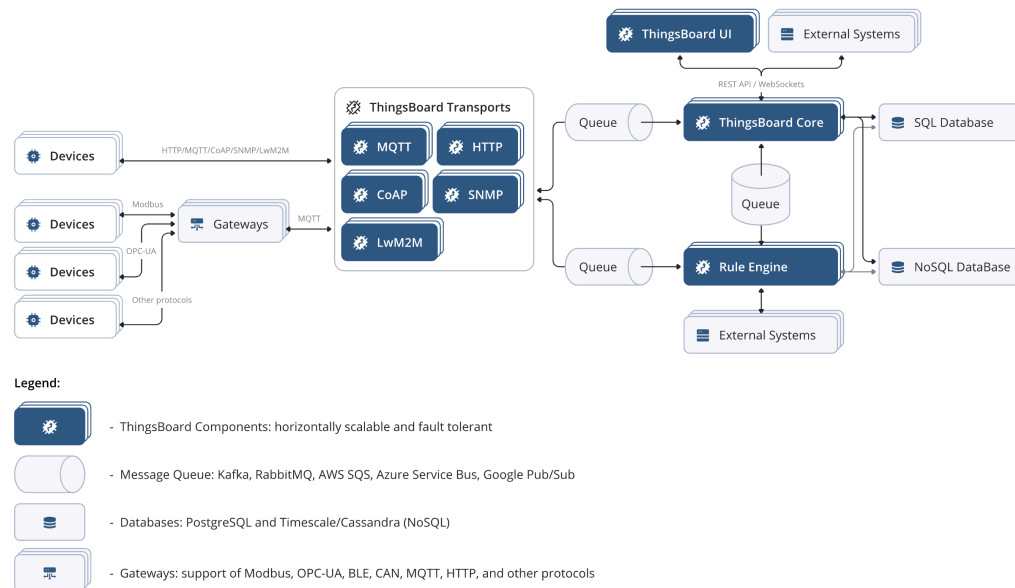


Ilustración 7. Arquitectura Thingsboard (Imagen tomada de <https://thingsboard.io>)

La plataforma ofrece dos ediciones locales, la edición Community que es código abierto y la versión Professional que ofrece algunas características extendidas. Como ya se ha mencionado a lo largo del proyecto el enfoque es utilizar herramientas de código abierto, por esto en este proyecto se utiliza la versión Community que cuenta con las siguientes funcionalidades (Thingsboard funcionalidades, 2023):

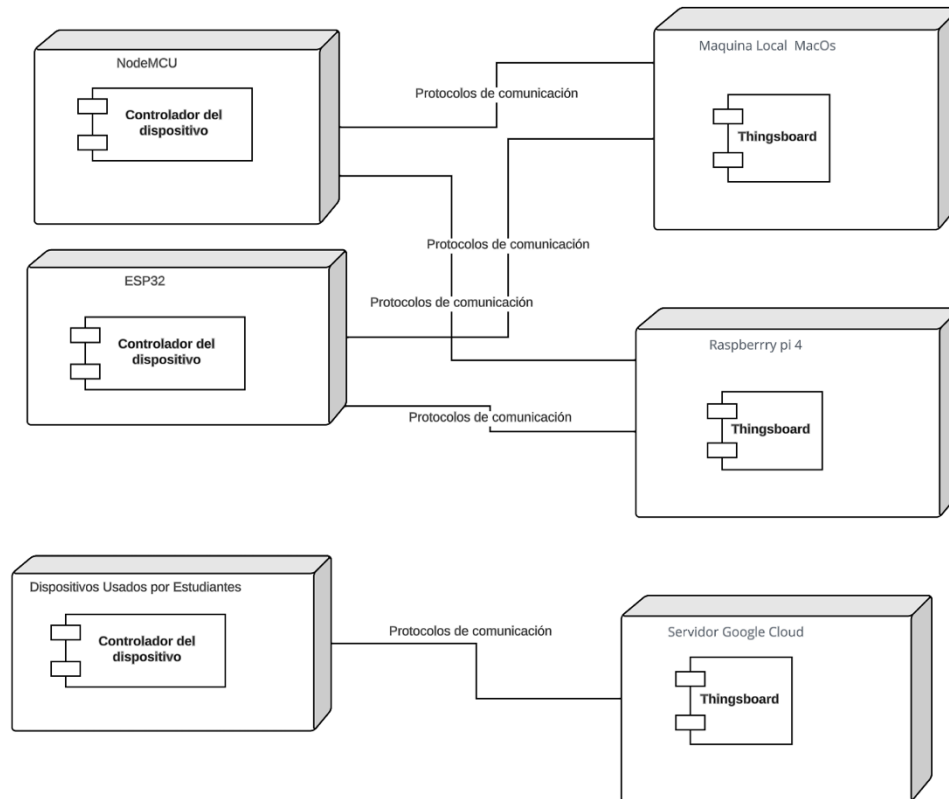
- Visualización de datos en tiempo real. Los datos se pueden mostrar en forma de gráficos, tablas e incluso en widgets personalizados.
- Atributos. Esta es la información asociada a los sensores o dispositivos como el nombre, la ubicación, batería entre otros.
- Telemetría. Esta información son las mediciones que provienen de los sensores, por ejemplo, datos de temperatura o humedad.
- Entidades y relaciones. Dentro de las entidades admitidas se pueden encontrar dispositivos, propietarios, usuarios y assets, se puede definir la relación entre dos entidades como, por ejemplo, un asset que represente un bus y múltiples dispositivos en este.

- El motor de reglas es un framework para crear eventos basados en flujos de trabajo.
- Las llamadas de procedimiento remoto (RPC por sus siglas en inglés) es una funcionalidad que permite enviar comandos y controlar los dispositivos.
- Seguimiento y auditoría a las acciones del usuario y su almacenamiento.

Se plantea hacer el despliegue dentro las instalaciones del campus principal de la Institución Universitaria Politécnico Gran Colombiano utilizando un Raspberry pi 4, este despliegue se planea hacer con el apoyo de los docentes pertenecientes a la Escuela TIC con el fin de darle acceso a los estudiantes, sin embargo, como se buscaría hacer sobre la red existente de la universidad, sería necesario gestionar permisos y accesos por parte del área de tecnología, llevar acabo estos procesos puede que tomen más tiempo del que se tiene destinado para este proyecto, por tal motivo también se plantea el despliegue sobre la plataforma en la nube Google Cloud, esto se haría sobre un plan gratuito que se puede adquirir en la plataforma con la duración de tres meses, el tiempo sería suficiente para que los estudiantes puedan apoyarse de la herramienta para sus proyectos de aula y sea posible desarrollar este proyecto. Adicionalmente para hacer un primer uso de la herramienta se plantea el despliegue en la maquina personal para hacer experimentos en un ambiente local.

El uso ideal de la plataforma sería dentro de las instalaciones de la institución haciendo uso de certificados SSL que la plataforma permite configurar, además que el acceso se hiciera usando el dominio propio de la universidad y de esta forma se convirtiera en una herramienta disponible para toda la comunidad donde se puedan crear múltiples proyectos IoT, con diversos protocolos de comunicación de una forma segura, sin embargo como el propósito de este proyecto es verificar la posible adopción de la plataforma en la población estudiantil, los despliegues planteados se enfocan en la funcionalidad de la plataforma y no en configuraciones de seguridad.

Teniendo en cuenta lo anterior se harán despliegues en tres sistemas diferentes, dos con herramientas locales y uno en la nube, teniendo en cuenta la comunicación planteada en la Ilustración 6, los despliegues para el desarrollo del proyecto serían los que se pueden ver en la Ilustración 8.



*Ilustración 8. Despliegues planteados para el desarrollo del proyecto*

En este proyecto los despliegues se van a hacer con PostgreSQL para los casos de la máquina local y la Raspberry pi 4, se va usar el tipo de cola en memoria por las características de los equipos y con esta configuración se podría manejar hasta 1 000 entradas de información por segundo, en la máquina del servicio en la nube se va utilizar la opción de Kafka para para el tipo de cola y esto permitiría manejar hasta 15 000 entradas de información por segundo, adicionalmente si es necesaria una escala mayor, se podría usar Cassandra para soportar más de 20 000 entradas de información por segundo (Thingsboard escenarios despliegue, 2023).

Por último se planea generar un documento destinado a los docentes de la Escuela TIC que contenga la información para el acceso al despliegue dentro la institución y la propuesta para el uso de la herramienta con los estudiantes a nivel administrativo, el documento en cuestión se incluye como anexo pero las credenciales se comparten de forma externa por cuestiones de privacidad.

### 4.3. Fase de despliegue

Para esta fase lo que se busca es facilitar el uso de la plataforma IoT Thingsboard a un curso de Sistemas Digitales y Ensambladores que cuenta con 10 estudiantes, el propósito de esto es verificar que la herramienta sea capaz de gestionar diferentes casos de uso relacionados con IoT que van a ser planteados en los proyectos de aula de la asignatura.

Lo primero que se hace es realizar el despliegue en la máquina de uso personal para verificar que se pueda hacer uso de la herramienta utilizando la documentación existente, el proceso realizado se puede ver en el anexo 6.1.1.2, luego de tener el ambiente correctamente configurado para hacer uso de Thingsboard de forma local, se procede a hacer uso de la misma utilizando las placas de desarrollo seleccionadas en la fase 4.1.4, para este uso se simulan datos de sensores de temperatura y el propósito es poder visualizar la información enviada utilizando el protocolo MQTT, la prueba se completa de forma exitosa y el proceso se puede ver en el anexo 6.2

Una vez se han realizado pruebas exitosas de forma local, se busca la manera de desplegar la herramienta dentro de la red de la institución, en apoyo con los docentes de la Escuela TIC se hace la consulta al área de tecnología para verificar si es posible tener acceso a una IP pública para desplegar la herramienta dentro del dominio de la universidad, en vista que el proyecto no puede depender de los tiempos de respuesta del área de tecnología se procede a hacer el despliegue de la herramienta en el servicio de la nube Google Cloud. El proceso que se hizo dentro del equipo local se hizo en un ambiente virtualizado así que se busca instalar el mismo ambiente virtualizado dentro del servidor de la nube para replicar el proceso de instalación ya realizado. Luego de hacer este proceso y poder obtener una IP pública proporcionada por el servicio en la nube, se habilitan las reglas de tráfico para que se pueda utilizar la plataforma Thingsboard alojada en la nube desde cualquier sitio de Internet, el proceso se puede ver en el anexo 6.1.1.3.

Con el despliegue realizado en una IP pública se procede a darle acceso a los estudiantes del curso de Sistemas Digitales y Ensambladores, se comparte la dirección de acceso, la información de credenciales, un video que muestra cómo se utiliza un cliente HTTP para enviar información a un dispositivo en la plataforma y un archivo de Arduino que contenía un programa para realizar envío de datos al servidor por medio del protocolo HTTP. Con la información anteriormente mencionada se espera que los estudiantes puedan utilizar la plataforma Thingsboard como apoyo en el desarrollo de sus proyectos de aula, la información compartida se puede ver en el anexo 6.3.

Después de haber puesto la plataforma a disposición de los estudiantes, con apoyo de los docentes de la Escuela TIC se logra hacer un despliegue dentro de la institución utilizando una IP pública que ya existía, para ingresar al despliegue se debe utilizar el dominio **academiacisco.poligran.edu.co** y el puerto **2214**, al tener este despliegue disponible se plantea utilizarlo para la presentación de la Semana TIC que busca darle visibilidad al proyecto, el proceso realizado se puede ver en el anexo 6.1.1.4.

#### 4.4. Fase de verificación

##### 4.4.1. Semana TIC

La participación quedo programada para el día 17 de mayo en el city campus, esto se puede ver en la programación oficial del Poli.

<b>Miércoles 17 de mayo</b>	18:45 p. m.	Aplicaciones en salud de los modelos pre-entrenados para el procesamiento de lenguaje natural Elías Buitrago	<b>Bogotá:</b> City Campus Salón 403 y <b>Medellín:</b> Sede Los Colores- Mezzanine	<a href="https://bit.ly/3CX8q2B">https://bit.ly/3CX8q2B</a>
	19:45 p. m.	Hablemos de Seguridad en AWS Odina Jackos	<b>Bogotá:</b> City Campus Salón 405 y <b>Medellín:</b> Sede Los Colores- Mezzanine	<a href="https://bit.ly/3CX8q2B">https://bit.ly/3CX8q2B</a>
	19:45 p. m.	Taller: <i>ThingsBoard</i> como herramientas para implementaciones de IoT Nicolás Martínez	<b>Bogotá:</b> City Campus Salón 403 y 404	<a href="https://bit.ly/3CX8q2B">https://bit.ly/3CX8q2B</a>
	20:45 p. m.	Ciberamenazas para Gente Común Luis Castellano	<b>Bogotá:</b> City Campus Salón 403 y <b>Medellín:</b> Sede Los Colores- Mezzanine	<a href="https://bit.ly/3CX8q2B">https://bit.ly/3CX8q2B</a>

Ilustración 9. Programación de charla Thingsboard tomado de la programación semana TIC

Para la charla se tiene una asistencia de alrededor de 20 estudiantes y 5 docentes, en la primera parte de la charla se realiza una encuesta de conocimientos previos que se puede encontrar en el anexo 6.5, en la primera parte de la sesión se hizo una breve introducción a IoT, se mostraron algunos protocolos de comunicación como se puede ver en la Ilustración 10 y se compartió un caso de uso de IoT.



Ilustración 10. Participación de la semana TIC

Para la segunda parte de la charla se presenta el proyecto y se muestra que a raíz de este se ha seleccionado la plataforma Thingsboard como propuesta de uso interno de la universidad, apoyando el desarrollo de proyectos de aula en asignaturas donde se traten temáticas relacionadas a IoT, el despliegue que se había realizado generó conflictos con la red utilizada en el city campus que no se habían identificado hasta el día de la presentación, por lo que es necesario utilizar datos compartidos de un teléfono móvil para mostrar en funcionamiento el despliegue realizado, durante la sesión se muestra el panel de control como se puede ver en la Ilustración 11.

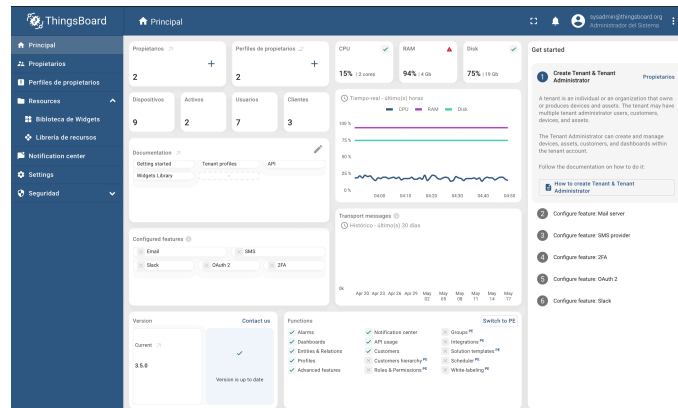


Ilustración 11. Panel de Control Thingsboard mostrado en semana TIC

Después de esto la red compartida empezó a fallar por lo que no se pudo seguir mostrando el despliegue hecho, a causa de esto tampoco se pudo finalizar la demostración en vivo con la herramienta.

#### 4.4.2. Proyectos de aula Sistemas Digitales

Después de haber realizado el despliegue en la nube de Google Cloud y haberle dado acceso a los estudiantes del curso a la plataforma, se programa una sesión de retroalimentación para escuchar a los estudiantes sobre las percepciones de la herramienta y el proyecto, al grupo también se les pide llenar la encuesta de conocimientos previos que se puede encontrar en el anexo 6.5.

Luego de esto la sesión se orienta a una entrevista grupal donde se busca conocer el uso que se le dio a la herramienta, las preguntas planteadas se pueden ver en el anexo 6.4. en términos generales, de esa entrevista se pudo concluir que los estudiantes ven la herramienta bastante útil para utilizarla en el ámbito académico y así afianzar los conocimientos de IoT, también hubo un conceso en que les hubiera gustado contar con una guía dedicada a explicar las funcionalidades de Thingsboard y ejemplos de su uso.

Adicionalmente a la entrevista se hace una validación de los proyectos de aula entregados por los estudiantes con el fin de evaluar la utilidad que tuvo la plataforma en su desarrollo. En total durante el curso se trabajaron en 3 proyectos diferentes: Implementación de máquina de estados SUMO, Dispensadores automatizados de alimentos para mascotas y

una estación meteorológica. A continuación, se presenta una breve descripción y resultado de cada uno de estos:

- Implementación máquina de estados SUMO: En este proyecto se visualiza que los estudiantes diseñaron y programaron un robot que tuviera la capacidad de vencer a su oponente en una competencia de sumo utilizando diferentes combinaciones entre habilidades de detección, de control y de ataque. El robot es programado y controlado utilizando la plataforma Arduino y posteriormente a través de la placa ESP32 Wroom se conecta a Thingsboard para permitir la visualización en tiempo real del comportamiento de los sensores de movimiento que contiene el robot para detectar la posición y los movimientos del oponente. (Estupiñán López, López Rojas, Ramírez, & Duarte Cepeda, 2023)

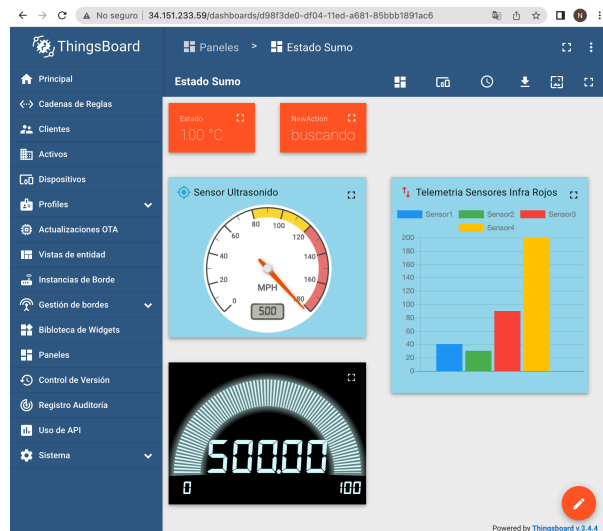


Ilustración 12. Tablero proyecto Sumo

- Dispensadores automatizados de alimentos: El objetivo de este proyecto fue diseñar un dispensador de alimentos para mascotas controlado a través de un dispositivo móvil. Como resultado se logró la definición de la cantidad exacta de comida que debería ingerir y la cantidad de comida que el dispensador dará diariamente basada en el peso y la edad de la mascota, todo a través un modelo de conexión Wifi controlado a través del protocolo MQTT. (Becerra Ruiz, Tamara Rodriguez, Redondo Hernandez, & Ramirez, 2023)

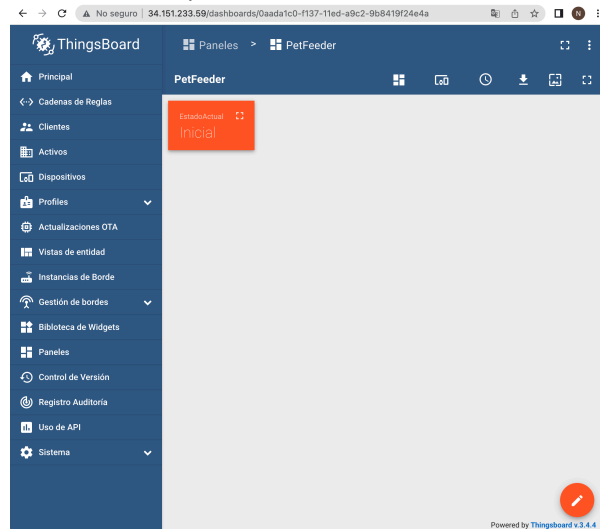


Ilustración 13. Tablero proyecto PetFeeder

- Estación meteorológica: Con este proyecto se buscó la utilización de diferentes sensores para medir y recolectar datos meteorológicos, tales como temperatura, humedad, presión atmosférica y velocidad del viento para hacer predicciones meteorológicas y monitorear el clima en tiempo real. Todos los sensores están conectados a un Arduino UNO, una placa ESP8266MOD Wifi para establecer comunicación utilizando el protocolo HTTP para conectar con la plataforma Thingsboard y poder visualizar los datos recolectados. (Ruiz, Londoño, & Sánchez, 2023)

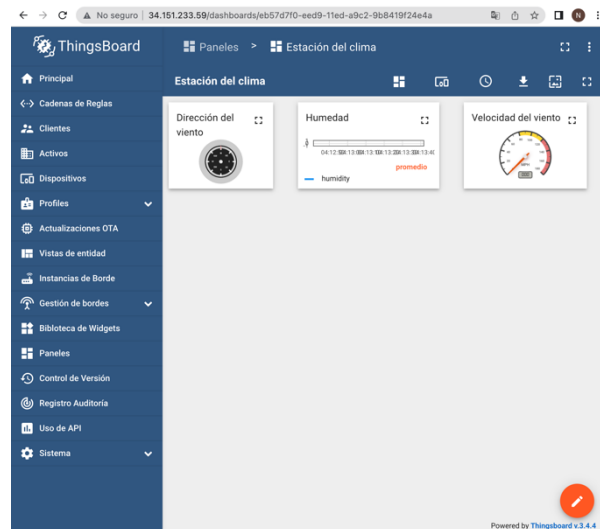


Ilustración 14. Proyecto estación del clima



Con los resultados de los proyectos de aula se puede ver que los estudiantes utilizaron la herramienta como apoyo para su desarrollo, además se puede ver que la plataforma es flexible y se puede trabajar diversos casos de uso, esto es una ventaja para el espacio académico.

#### 4.4.3. Resultados Encuesta conocimientos previos

Tras analizar los resultados de la encuesta de conocimiento previo se resaltan los siguientes aspectos:

- De un total de 18 encuestados solo 4 estudiantes no tenían previo conocimiento de lo que es IoT, lo cual representa tan solo el 22% de la población.
- El 67% de los encuestados no habían tenido experiencia previa en proyectos de IoT lo cual suma valor a la iniciativa ya que con ella se busca fomentar el uso de esta herramienta en ambientes educativos de desarrollo.
- El 100% de los encuestados confirmaron que les gustaría tener acceso a herramientas de IoT en la universidad y también consideran que contar con laboratorios de IoT les ayudaría a agilizar el proceso de desarrollo de proyectos relacionados con tecnologías de conectividad y dispositivos inteligentes.
- Entre los mayores desafíos percibidos en cuanto a la implementación de proyectos IoT se encuentran: el conocimiento de los protocolos de comunicación y la conectividad entre dispositivos.
- La herramienta de IoT de mayor utilización en la población de estudio es Arduino y tan solo 1 persona encuestada conoce del protocolo MQTT.

Basados en los resultados, se concluye que la comunidad ve interés y utilidad en la herramienta ya que a través de Thingsboard se puede acompañar el desarrollo de conocimiento en distintos frentes como lo son el uso de protocolos, la interoperabilidad de dispositivos y la implementación de proyectos relacionados a IoT en espacios educativos.

# Capítulo 5

## 5. Conclusiones y trabajo futuro

### 5.1. Conclusiones

- Las herramientas de código abierto son muy importantes para trabajar con proyectos de IoT, contribuyen a que su desarrollo sea más accesible y flexible.
- Dentro de las sesiones de verificación se puede concluir que la comunidad de estudio conoce qué es IoT pero no se profundiza en proyectos, además la comunidad ve interés y utilidad en la solución planteada, este interés le da más relevancia, porque facilitaría que esos estudiantes se vinculen con proyectos que utilicen esta tecnología.
- El despliegue realizado para desarrollar proyectos de IoT contando con herramientas propias es capaz de apoyar a los estudiantes en el desarrollo de proyectos de aula con diferentes casos de uso.
- Para hacer más fácil el uso de la herramienta se establece que es necesario generar una guía dirigida a los estudiantes, orientada a explicar las funcionalidades de la plataforma, dicha guía se anexa al presente documento.

### 5.2. Trabajo futuro

Para un futuro se puede adoptar la plataforma Thingsboard para apoyar los proyectos de IoT realizados en la universidad, para esto se necesitaría el apoyo del área de tecnología para poder realizar los despliegues en equipos proporcionados por la universidad, utilizando los dominios propios e implementando las configuraciones de seguridad. Adicionalmente el apoyo del cuerpo docente para incentivar el desarrollo de proyectos.

### 5.3. Bibliografía

- MQTT Sparkplug solution.* (s.f.). Recuperado el 16 de marzo de 2023, de <https://www.hivemq.com/solutions/technology/mqtt-sparkplug/>
- Tyutyundzhiev, N., Angelov, C., Arsov, T., Lovchinov, K., Nichev, H., Mutafov, A., & Alexieva, G. (2023). *Remote datalogging of solar UV irradiation using open-source ESP32 platform and MQTT protocol.* Journal of Physics: Conference Series.
- Okhovat, E., & Bauer, M. (2021). *Monitoring the Smart City Sensor Data Using Thingsboard and Node-Red.* Proceedings - 2021 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Internet of People, and Smart City Innovations, SmartWorld/ScalCom/UIC/ATC/IoP/SCI 2021.
- Manchanayaka, M. A., Wijesekara, J. P., Yang, C. Y., Yang, C. Y., Premachandra, C., Firdhous, M. F., & Sudantha, B. H. (2021). *Open, IoT powered Environmental Air Pollution Monitoring Framework for Traffic Management.* Proceedings of 6th International Conference on Information Technology Research: Digital Resilience and Reinvention, ICITR 2021.
- Kaa documentation .* (s.f.). Recuperado el 2023, de <https://kaaproject.github.io/kaa/docs/v0.10.0/Programming-guide/Key-platform-features/>
- Arduino cloud.* (s.f.). Recuperado el 2023, de <https://cloud.arduino.cc/plans>
- openHab.* (s.f.). Recuperado el 2023, de <https://www.openhab.org/docs/>
- Node-RED.* (s.f.). Recuperado el 2023, de <https://nodered.org/docs/>
- ThingSpeak.* (2023). Obtenido de [https://thingspeak.com/pages/learn\\_more](https://thingspeak.com/pages/learn_more)
- Home Assistant.* (s.f.). Recuperado el 2023, de <https://www.home-assistant.io/docs/>
- ThingsBoard docs.* (s.f.). Recuperado el abril de 2023, de <https://thingsboard.io/docs/getting-started-guides/what-is-thingsboard/>
- Thingsboard arquitectura.* (s.f.). Recuperado el 2023, de <https://thingsboard.io/docs/reference/>
- Thingsboard funcionalidades.* (2023). Obtenido de <https://thingsboard.io/docs/>
- Docker ubuntu.* (2023). Obtenido de <https://docs.docker.com/engine/install/ubuntu/>
- Thingsboard escenarios despliegue.* (2023). Obtenido de <https://thingsboard.io/docs/reference/iot-platform-deployment-scenarios/#performance-requirements>
- Thingsboard instalación Docker.* (2023). Obtenido de <https://thingsboard.io/docs/user-guide/install/docker/?ubuntuThingsboardQueue=kafka>
- Thingsboard ubuntu server.* (2023). Obtenido de <https://thingsboard.io/docs/user-guide/install/ubuntu/>
- "IEEE Standard for an Architectural Framework for the Internet of Things (IoT)," in IEEE Std 2413-2019 , vol., no., pp.1-269, 10 March 2020, doi: 10.1109/IEEESTD.2020.9032420. (2020). *IEEE.*
- sparkfun drivers CH341.* (2023). Obtenido de <https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers/all#mac-osx>
- Thingsboard dispositivos.* (2023). Obtenido de <https://thingsboard.io/docs/devices-library>
- Estupiñán López, J. C., López Rojas, K. N., Ramírez, D. A., & Duarte Cepeda, F. (2023). *Implementación máquina de estados SUMO.* Bogotá: Proyecto de aula Sistemas Digitales y Ensambladores no publicado.

- Becerra Ruiz, N. V., Tamara Rodriguez, J. C., Redondo Hernandez, E. R., & Ramirez, D. A. (2023). *DISPENSADOR AUTOMATIZADO DE ALIMENTOS PARA MASCOTAS*. Bogotá: Proyecto de aula sistemas digitales y ensambladores no publicado.
- Ruiz, S. S., Londoño, J. B., & Sánchez, L. Y. (2023). *Estación Meteorológica*. Bogotá: Proyecto de aula sistemas digitales y ensambladores no publicado.

# Capítulo 6

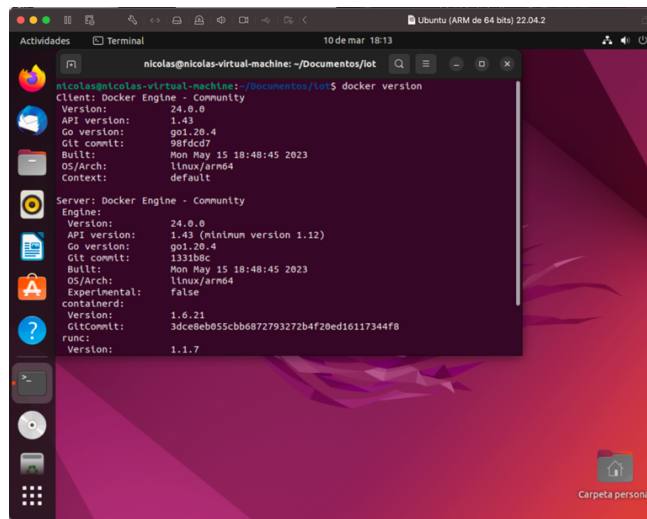
## 6. Anexos

### 6.1.1. Guía instalación Thingsboard

#### 6.1.1.1. Instalación Docker sobre Ubuntu

Dentro del sistema operativo Ubuntu lo primero que se debe hacer es la instalación de Docker, después de esto debe agregar el usuario del sistema al grupo de docker, la forma de hacer esto se puede ver consultar en la documentación oficial de Docker (Docker ubuntu, 2023).

Después de haber ejecutado los comandos descritos en la documentación, ya se debería tener Docker instalado y también el usuario del sistema tendría que estar agregado el grupo de acceso docker, para verificar esto se puede correr el comando “docker version” sin uso de la sentencia “sudo” y el resultado debería ser como el que se muestra en la Ilustración 15 .



```
nicolas@nicolas-virtual-machine: ~/Documentos/lot
Client: Docker Engine - Community
 Version: 24.0.0
 API version: 1.43
 Go version: go1.20.4
 Git commit: 98fcdcd7
 Built: Mon May 15 18:48:45 2023
 OS/Arch: linux/arm64
 Context: default

Server: Docker Engine - Community
 Engine:
  Version: 24.0.0
  API version: 1.43 (minimum version 1.12)
  Go version: go1.20.4
  Git commit: 1331bb8c
  Built: Mon May 15 18:48:45 2023
  OS/Arch: linux/arm64
  Experimental: false
 containerd:
  Version: 1.6.21
  Git commit: 3dce8eb055cbb6872793272b4f20ed16117344f8
 runc:
  Version: 1.1.7
```

Ilustración 15. Verificación de instalación de Docker en Ubuntu

#### 6.1.1.2. Instalación en equipo local

Para utilizar la herramienta en el equipo local es necesario hacer un proceso de virtualización, se utiliza una utiliza una imagen de Linux Ubuntu 22.04.2 LTS (Jammy Jellyfish).

El sistema operativo sobre el que se trabaja es macOS Ventura versión 13.4 y para poder iniciar el proceso de instalación, primero se descarga el programa de virtualización VMware Fusion y después se crea una máquina virtual con la imagen de Ubuntu antes mencionada. Las características usadas para la máquina virtual son 2 núcleos de procesador y 4096 MB de memoria.

Después de esto es necesario hacer la instalación de Docker como se puede ver en la sección 6.1.1.1. Una vez se tiene Docker configurado, el ambiente está listo para hacer la instalación de Thingsboard con los siguientes pasos descritos en la documentación oficial (Thingsboard instalación Docker, 2023):

- Crear el archivo “docker-compose.yml” el cual va a tener la configuración del contenedor, la opción de instalación elegida utiliza la persistencia en PostgreSQL y el servicio de encolamiento se hace en memoria, las configuraciones de los diferentes contenedores disponibles se pueden encontrar en (Thingsboard instalación Docker, 2023).
- Crear los directorios para el almacenamiento de la data y los logs, también es importante dar acceso al grupo de usuario utilizado por el contenedor, para esto se pueden utilizar los comandos “mkdir -p ~/ .directorio && sudo chown -R 799:799 ~/ .directorio”
- Por último, se procede a ejecutar el contenedor para que haga la instalación de la plataforma con el comando “docker compose up -d”

Una vez hecho esto se debería poder acceder a la plataforma accediendo a la IP del host, en este caso es el localhost, y al puerto expuesto para HTTP que en este caso es el 8080, al ingresar a la URL debería ser redirigido a una pantalla como la que se muestra en Ilustración 16.

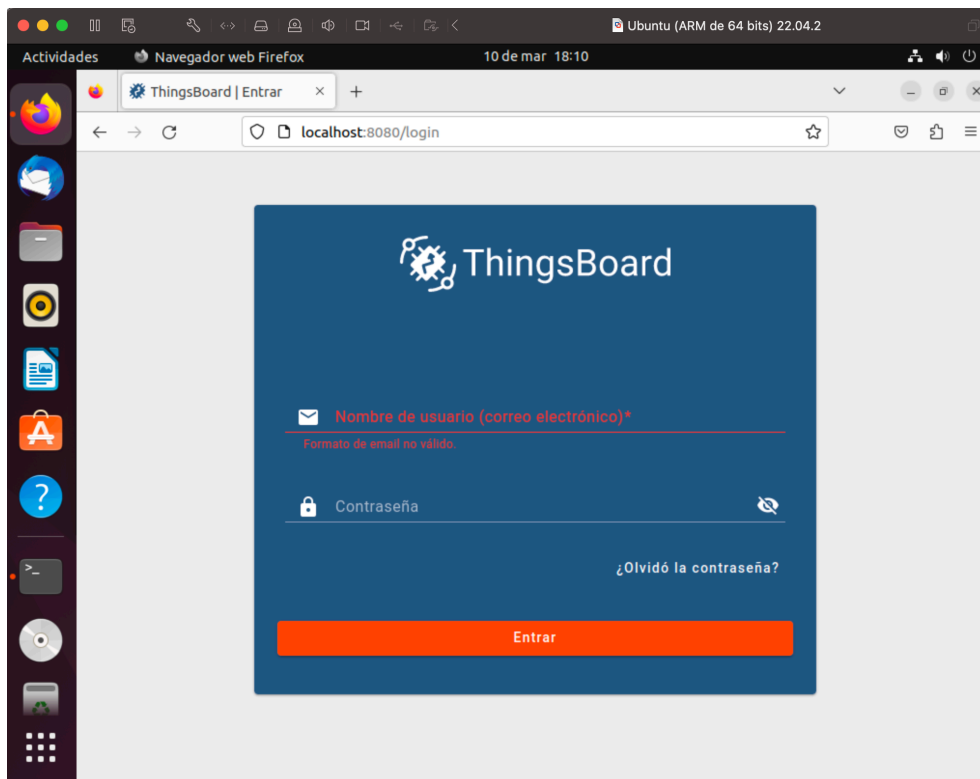


Ilustración 16. Instalación exitosa en Ubuntu

### 6.1.1.3. Instalación en Google Cloud

Para la instalación en Google Cloud se hace uso del servicio Compute Engine, esto con el propósito de contar con una instancia que cuente con la virtualización del sistema operativo “Linux Ubuntu 22.04.2 LTS” (Jammy Jellyfish), una IP pública estática para que se pueda acceder desde internet y que se mantenga la misma dirección mientras la máquina está encendida.

La estimación por mes está dentro de la capacidad del periodo de prueba con los créditos gratuitos que son 300USD, y por una duración de 3 meses, la estimación se puede ver en la Ilustración 17.

The screenshot shows the 'Estimate' interface for a Google Cloud Compute Engine instance. It details the configuration and associated costs for one month.

Compute Engine	
1 x	
Region: Sao Paulo	
730 total hours per month	
Provisioning model: Regular	
Instance type: n2-standard-2 Sustained Use Discount applied (20%)	USD 90.05
Operating System / Software: Free	
Static public IP: 1 (730 hours)	USD 2.92
<b>Estimated Component Cost: USD 92.97 per 1 month</b>	
Persistent Disk (Accompanying)	
1 x boot disk	
Product accompanying: Compute Engine	
Zonal balanced PD: 20 GiB	USD 3.00
<b>USD 3.00</b>	
<b>Total Estimated Cost: USD 95.97 per 1 month</b>	
Estimate Currency	
USD - US Dollar	

Ilustración 17. Estimación Ubuntu en Google Cloud

Una vez creada la maquina se accede por SSH y se procede a hacer la instalación de Docker como se puede ver en la sección 6.1.1.1. Con la instalación de Docker se puede proceder al despliegue de Thingsboard con los siguientes pasos (Thingsboard instalación Docker, 2023):

- Crear el archivo “docker-compose.yml” en el cual se especifica la configuración del contenedor que va a inicializar la plataforma, para esta instalación se utiliza el sistema de persistencia PostgreSQL y para el sistema de encolamiento se utiliza el servicio de Kafka, las configuraciones necesarias para crear este contenedor se pueden encontrar en (Thingsboard instalación Docker, 2023).
- Crear los directorios para el almacenamiento de la data y los logs, también es importante dar acceso al grupo de usuario utilizado por el contenedor, para esto se pueden utilizar los comandos “mkdir -p ~/ .directorio && sudo chown -R 799:799 ~/ .directorio”
- Se realiza la ejecución del contenedor para que haga la instalación de la plataforma con el comando “docker compose up -d”
- Por último, se habilitan los puertos utilizados por la aplicación en las reglas de firewall.

#### Filtros de fuente

Rangos de IP	0.0.0.0/0
--------------	-----------

#### Protocolos y puertos

tcp:8080

tcp:80

tcp:1883

tcp:7070

*Ilustración 18.Reglas de firewall en GCP*



Con esta configuración hecha ya se puede acceder a la instancia en la nube con la IP que se haya generado desde la plataforma de Google. En este caso la dirección IP es 34.151.215.47 y al acceder se puede ver la aplicación corriendo como en la ilustración Ilustración 19.

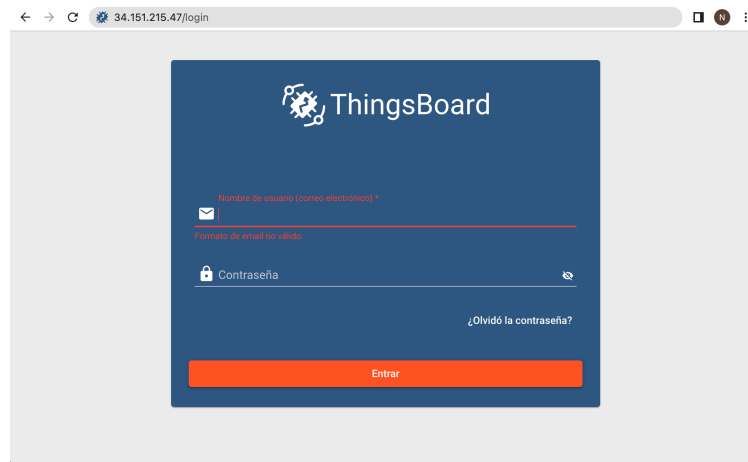


Ilustración 19. Despliegue en GCP

#### 6.1.1.4. Instalación en raspberry Pi 4

Dentro de las opciones disponibles para la instalación de la plataforma en el dispositivo raspberry Pi, se puede usar la opción de instalación directamente sobre el sistema operativo.

- La plataforma corre en Java 11 por lo que hay que instalar el OpenJDK11, para esto se puede utilizar primero el comando “sudo apt update” con el fin de actualizar el sistema y posteriormente instalar con “sudo apt install openjdk-11-jdk”.
- Se puede verificar la instalación con el comando “java -version” y el resultado esperado es.

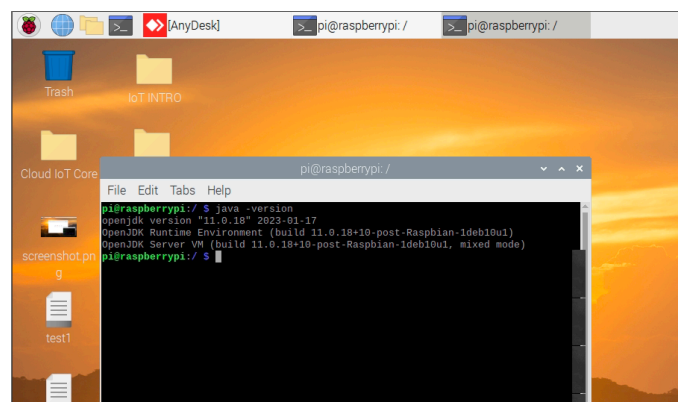


Ilustración 20. Verificación instalación OpenJDK 11

- Se hace la descarga de Thingsboard se configura la base datos y se agrega como servicio, esta configuración se puede encontrar en (Thingsboard ubuntu server, 2023). como servicio, la descarga se

- Por último se agregan los servicios al archivo “rc.local ” ubicado en el directorio “/etc” para que la herramienta se inicialice cada vez que se prenda el dispositivo

```

- sudo service postgresql start
-
- # esperar que inicie la bd antes de thingsboard
- sleep 10
-
- sudo service thingsboard start

```

- Una vez finalizada la instalación se puede verificar el funcionamiento del servicio con el comando “systemctl status thingsboard” y la respuesta esperada es como la de la Ilustración 21.

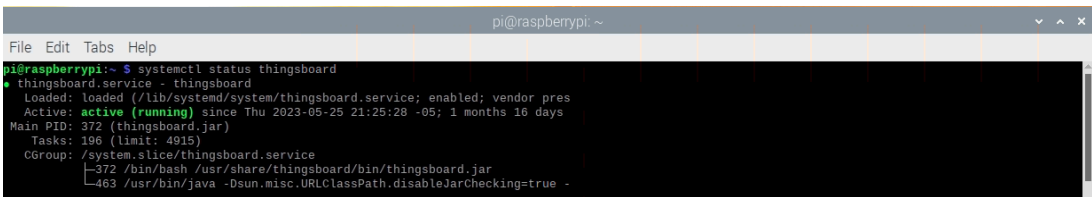


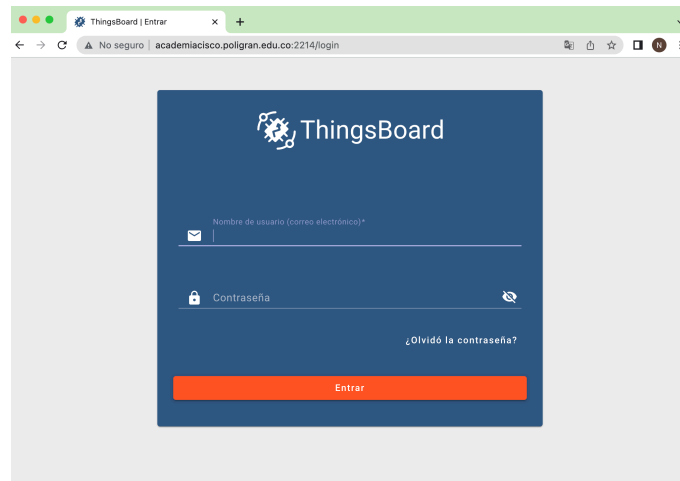
Ilustración 21. Servicio activo Thingsboard en raspberry Pi 4

- El despliegue se hizo dentro de la universidad y para esto fue necesario hacer un mapeo de puertos para poder acceder a los servicios de la plataforma, el mapeo realizado se puede ver en la Ilustración 22.

Nombre de aplicación	Puerto externo	Puerto interno	Protocolo	N.º IP del dispositivo
MQTT	2212	1883	Ambos	192.168.1.51
REALSERVER	2213	7070	Ambos	192.168.1.51
HTTP	2214	8080	Ambos	192.168.1.51
COAP	2215	5683	Ambos	192.168.1.51
MQTT/PI	2216	22	Ambos	192.168.1.51
COAP	2217	5688	Ambos	192.168.1.51

Ilustración 22. Reglas de puertos para el despliegue en la Universidad

- Finalmente, con la configuración hecha se puede acceder a la plataforma desde el dominio público de la universidad **academicisco.poligran.edu.co** para acceder a los servicios de Thingsboard como se puede ver en la Ilustración 23.



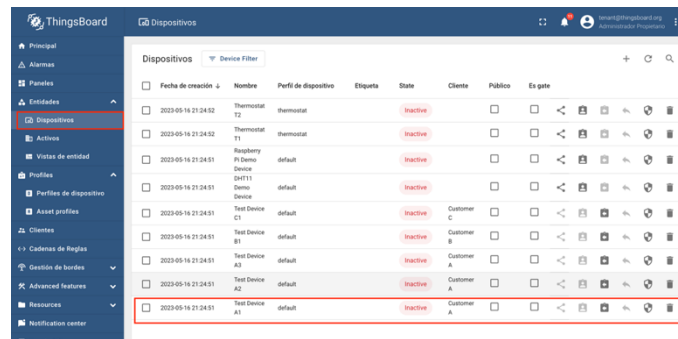
*Ilustración 23. Despliegue en raspberry Pi 4*

## 6.2. Experimentos locales

Después de tener la instalación de la plataforma hecha, se puede acceder a la misma con las credenciales por defecto de la instalación, las credenciales son las siguientes: (ThingsBoard docs, s.f.)

System Administrator: sysadmin@thingsboard.org / sysadmin  
Tenant Administrator: tenant@thingsboard.org / tenant  
Customer User: customer@thingsboard.org / customer

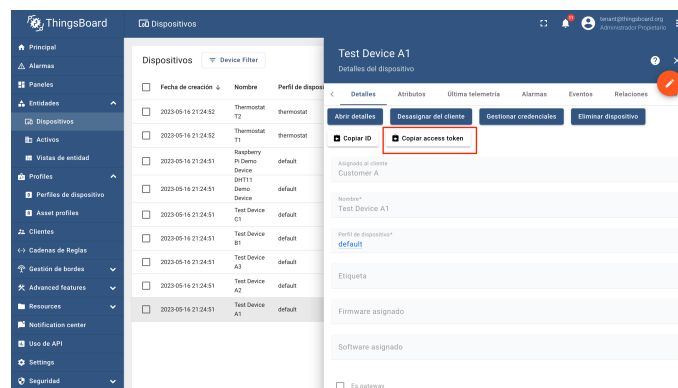
Para el experimento se utiliza el usuario tenant que tiene la capacidad de crear nuevos dispositivos y asociar tableros a los mismos. La plataforma cuenta con dispositivos por defecto, se seleccionan uno para trabajar, se pueden ver en la sección de dispositivos tal como se muestra en la Ilustración 24.



Fecha de creación	Nombre	Perfil de dispositivo	Etiqueta	State	Cliente	Público	Es gate
2023-05-16 21:24:52	Thermostat T2	thermostat		Inactive			
2023-05-16 21:24:52	Thermostat T1	thermostat		Inactive			
2023-05-16 21:24:51	Raspberry Pi Demo Device DM11	default		Inactive			
2023-05-16 21:24:51	Demo Device C1	default		Inactive			
2023-05-16 21:24:51	Test Device B1	default		Inactive	Customer B		
2023-05-16 21:24:51	Test Device A3	default		Inactive	Customer A		
2023-05-16 21:24:51	Test Device A2	default		Inactive	Customer A		
2023-05-16 21:24:51	Test Device A1	default		Inactive	Customer A		

Ilustración 24. Dispositivos seleccionados en Thingsboard.

Una vez seleccionados los dispositivos a los que se les va enviar la información, es importante guardar los tokens de acceso que van a ser necesario al momento de ejecutar los controladores en los dispositivos de hardware, esto se puede ver en la Ilustración 25, para este caso el token es A1\_TEST\_TOKEN.



Fecha de creación	Nombre	Perfil de dispositivo
2023-05-16 21:24:52	Thermostat T2	thermostat
2023-05-16 21:24:52	Thermostat T1	thermostat
2023-05-16 21:24:51	Raspberry Pi Demo Device DM11	default
2023-05-16 21:24:51	Demo Device C1	default
2023-05-16 21:24:51	Test Device B1	default
2023-05-16 21:24:51	Test Device A3	default
2023-05-16 21:24:51	Test Device A2	default
2023-05-16 21:24:51	Test Device A1	default

Test Device A1
Asignado al cliente: Customer A
Nombre: Test Device A1
Perfil de dispositivo: default
Etiqueta:
Firmware asignado:
Software asignado:
Es gateway:

Ilustración 25. Token de acceso para el dispositivo

Para el uso de las placas lo primero que hay que hacer es configurar los drivers para reconocer las placas NodeMCU y ESP 32, esto se debe a que el computador con el que se

trabaja no contiene puertos USB y se utiliza un hub usb tipo c para permitir la conexión por un puerto USB, la instalación del driver esta descrita en (sparkfun drivers CH341, 2023).

Adicionalmente para el uso de las placas se debe hacer una pre-configuración en el IDE Arduino para que sea posible establecer la conexión entre las placas y el servidor (Thingsboard dispositivos, 2023)

De forma general hay que descargar las librerías de Thingsboard al IDE y esto se hace ingresando a Tools > Manage Libraries busque e instale “ThingsBoard” y en el mismo apartado se debe buscar e instalar la librería “mbedtls”. Para el uso específico del microcontrolador se debe incluir la placa desde el IDE.

- NodeMCU

Se añade en Arduino IDE > Preferences... y agregue la siguiente URL al campo “Additional Boards Manager URLs” y adicione la siguiente url

```
"http://arduino.esp8266.com/stable/package_esp8266com_in  
dex.json"
```

Una vez agregada la URL, se añade la placa en Tools > Board > Board Manager y se selecciona la placa ESP8266

- ESP 32

Se añade en Arduino IDE > Preferences... y agregue la siguiente URL al campo “Additional Boards Manager URLs” y adicione la siguiente url

```
"http://arduino.esp8266.com/stable/package_esp8266com_in  
dex.json"
```

Una vez agregada la URL, se añade la placa en Tools > Board > Board Manager y se selecciona la placa ESP8266

Después de esto se llama a la librería de Thingsboard en Arduino

```
#if defined(ESP8266)  
  #include <ESP8266WiFi.h>  
  #define THINGSBOARD_ENABLE_PROGMEM 0  
#elif defined(ARDUINO_NANO_RP2040_CONNECT)  
  #include <WiFiNINA_Generic.h>  
#elif defined(ESP32) || defined(RASPBERRYPI_PICO) ||  
defined(RASPBERRYPI_PICO_W)  
  #include <WiFi.h>  
  #include <WiFiClientSecure.h>  
#endif
```

Con estas librerías se puede establecer la conexión hacia la aplicación de Thingsboard, acá es donde se usa el token mencionado anteriormente, se especifican los parámetros de conexión y una vez establecida la conexión se envían datos simulados que representan la telemetría este ejemplo se envía información simulada de sensores de temperatura y humedad como se puede ver a continuación.

```
// datos de conexión
constexpr char WIFI_SSID[] = "TP-Link_54D5";
constexpr char WIFI_PASSWORD[] = "psw";
constexpr char TOKEN[] = "A1_TEST_TOKEN";
constexpr char THINGSBOARD_SERVER[] = "192.168.5.116";
constexpr uint16_t THINGSBOARD_PORT = 1883U;

// iniciar cliente
WiFiClient wifiClient;
ThingsBoard tb(wifiClient, MAX_MESSAGE_SIZE);

// datos simulados
if (millis() - previousDataSend > telemetrySendInterval) {
    previousDataSend = millis();
    tb.sendTelemetryInt("temperature", random(10, 20));
    tb.sendAttributeInt("rssi", WiFi.RSSI());
    tb.sendAttributeInt("channel", WiFi.channel());
    tb.sendAttributeString("bssid", WiFi.BSSIDstr().c_str());
    tb.sendAttributeString("localIp", WiFi.localIP().toString().c_str());
    tb.sendAttributeString("ssid", WiFi.SSID().c_str());
}
```

Al subir el programa de forma exitosa en la placa se puede como se conecta al host de la aplicación y cómo queda suscrito a la cola de mensajes

```
05:36:48.194 -> Connecting to: 192.168.5.115 with token A1_TEST_TOKEN
05:36:49.118 -> Subscribing for RPC...
05:36:49.118 -> Subscribe done
05:36:49.118 -> [TB] Requesting shared attributes transformed from (ledState,blinkingInterval) into json ({"sharedKeys":"ledState,blinkingInterval"})
05:36:49.118 -> [TB] Requesting shared attributes transformed from (ledMode) into json ({"clientKeys":"ledMode"})
05:36:49.314 -> [TB] Callback onMQTTMessage from topic: (v1/devices/me/attributes/response/2)
05:36:49.347 -> [TB] Received shared attribute request
05:36:49.347 -> [TB] Calling subscribed callback for response id (2)
05:36:49.347 -> [TB] Callback onMQTTMessage from topic: (v1/devices/me/attributes/response/1)
05:36:49.347 -> [TB] Received shared attribute request
05:36:49.347 -> [TB] Calling subscribed callback for response id (1)
```

*Ilustración 26. Conexión a tópico MQTT*

Una vez se haya montado de forma exitosa el programa en la placa, se pueden ver los atributos del dispositivo y se puede proceder a crear un tablero.

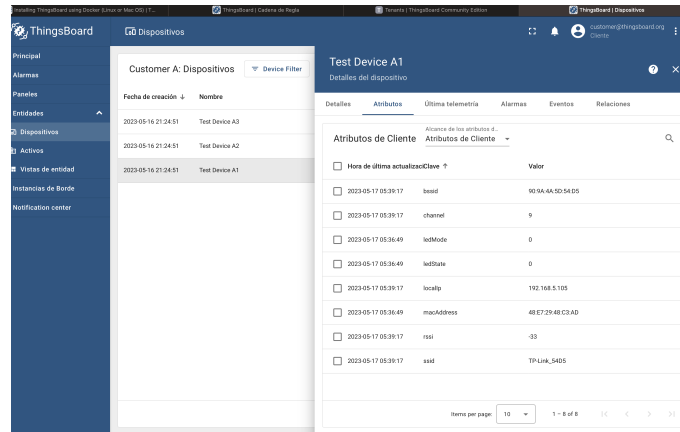


Ilustración 27. Atributos registrados en Thingsboard

Para crear un tablero se debe ir a la sección de paneles, crear uno nuevo y ponerle el nombre deseado, para este caso se pone el nombre del dispositivo A1.

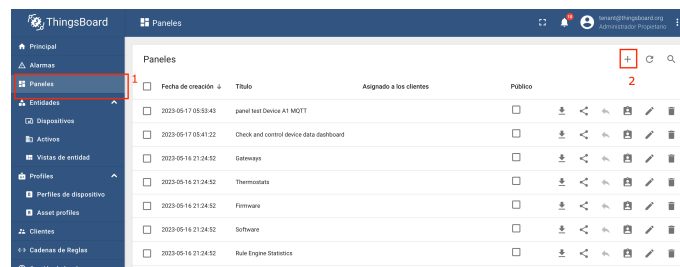


Ilustración 28. Creación de tablero Thingsboard

Una vez dentro del panel se pueden crear los widget que se deseen, en este caso se crea un card para mostrar la información del dispositivo embebido, un gráfico del histórico de temperatura actualizado en tiempo real y un indicador digital con la última medición obtenida que también se actualiza en tiempo real como se puede ver en la Ilustración 29.

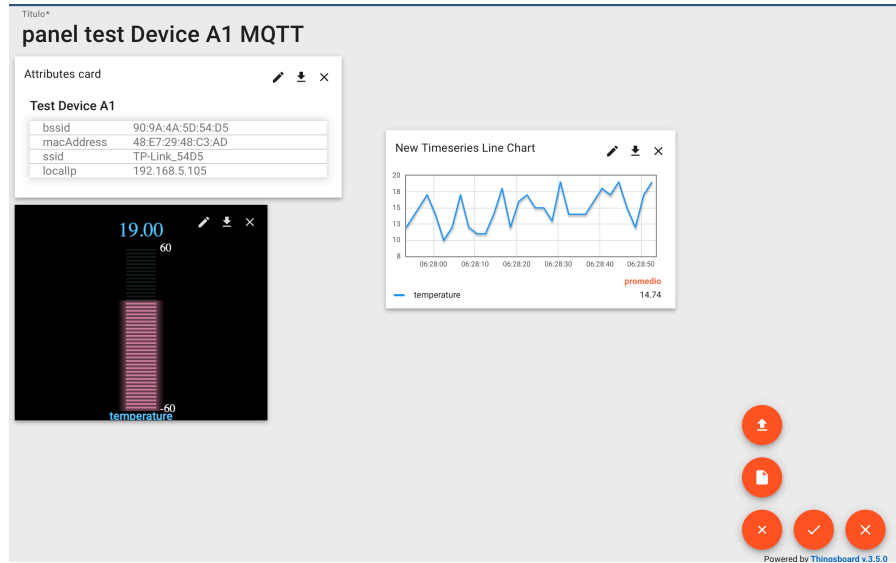
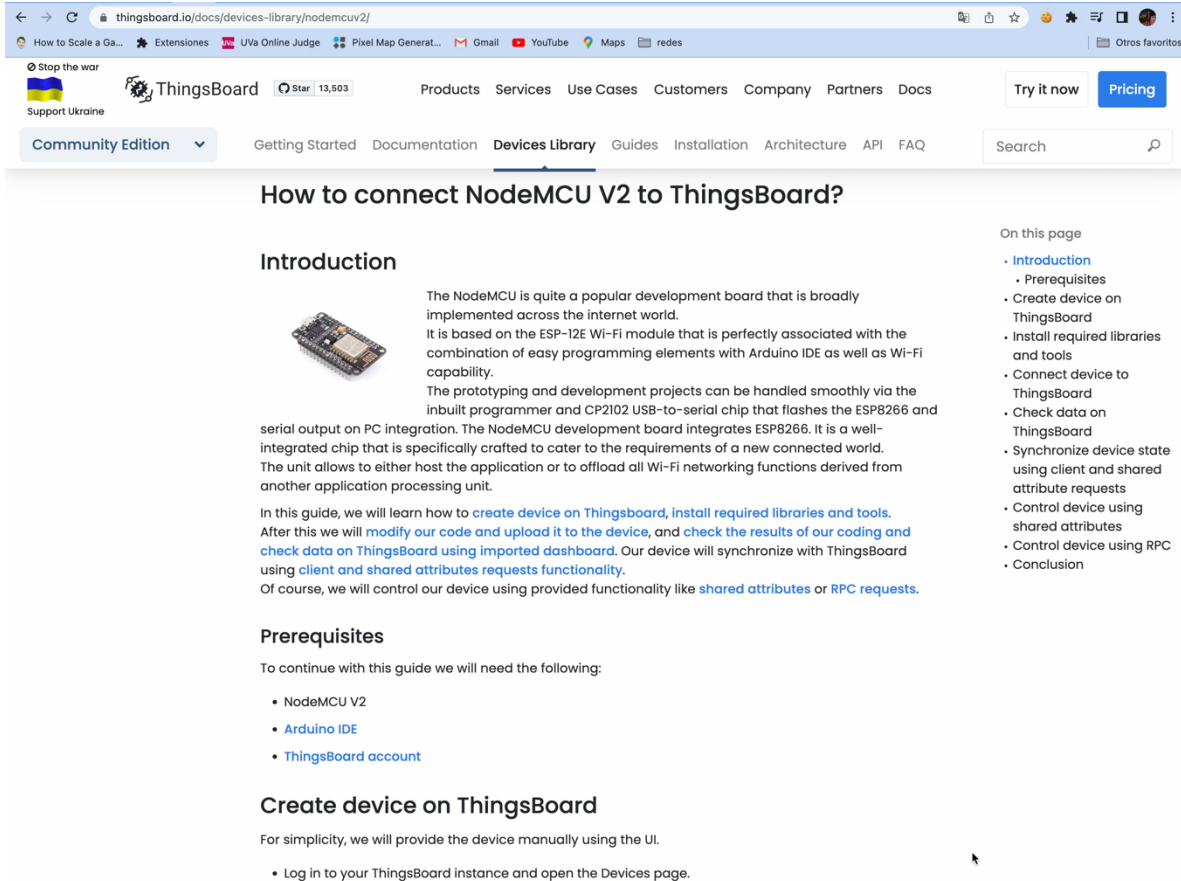


Ilustración 29. Tablero de temperatura simulada



### 6.3. Información proporcionada

Al curso de Sistemas Digitales y Ensambladores se les proporciona la información para acceder al servidor desplegado en el servidor de Google Cloud, se les envía un video donde se muestra la comunicación de un cliente HTTP con el servidor, se puede ver en el Video 1, por ultimo un programa de Arduino que contiene un cliente HTTP.



The screenshot shows a web browser displaying the ThingsBoard documentation page. The page title is "How to connect NodeMCU V2 to ThingsBoard?". The main content includes an "Introduction" section with an image of a NodeMCU V2 board, a "Prerequisites" section listing NodeMCU V2, Arduino IDE, and a ThingsBoard account, and a "Create device on ThingsBoard" section. A table of contents on the right side of the page lists the sections: Introduction, Prerequisites, Create device on ThingsBoard, Install required libraries and tools, Connect device to ThingsBoard, Check data on ThingsBoard, Synchronize device state using client and shared attribute requests, Control device using shared attributes, Control device using RPC, and Conclusion.

*Video 1. Muestra para el uso de Thingsboard en GCP*

Las variables de acceso para el acceso fueron las siguientes :

- **ip** <http://34.95.227.197/>
- **System Administrator:** sysadmin@thingsboard.org / sysadmin
- **Tenant Administrator:** tenant@thingsboard.org / tenant
- **Customer User:** customer@thingsboard.org / customer

En la siguiente página se adjunta el código proporcionado.

```

#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>

void setup() {
  Serial.begin(115200);
  WiFi.begin("nombre_red", "contraseña_red");

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Conectando al WiFi...");
  }
  Serial.println("Conectado al WiFi");
}

void loop() {

  WiFiClientSecure wifiClient;
  HTTPClient client;

  //Esto se cambia por las lecturas del montaje
  // Crea un objeto JSON
  DynamicJsonDocument json(1024);
  json["temperature"] = 42;
  json["humidity"] = 73;
  // Convierte el objeto JSON a una cadena de caracteres
  String jsonString;
  serializeJson(json, jsonString);

  // Crea una solicitud HTTP POST con la URL y los encabezados necesarios
  client.begin(wifiClient, "http://34.95.227.197/api/v1/{token}}/telemetry");
  client.addHeader("Content-Type", "application/json");
  // Realiza la solicitud HTTP POST con los datos JSON
  int httpResponseCode = client.POST(jsonString);

  // Si la solicitud fue exitosa, imprime la respuesta
  if (httpResponseCode > 0) {
    String respuesta = client.getString();
    Serial.println(respuesta);
  } else {
    Serial.print("Error en la solicitud HTTP: ");
    Serial.println(httpResponseCode);
  }
  // Cierra la conexión
  client.end();

  // Espera 10 segundos antes de realizar otra solicitud
  delay(10000);
}

```

#### 6.4. Entrevista grupal estudiantes

**Fecha de la encuesta:** 23/05/2023 20:00

**Grupo:** Sistemas Digitales y Ensambladores

**Docente:** Gabriel Eduardo Ávila Buitrago

**Cantidad de estudiantes:** 10

- ¿Qué dificultades presentaron para hacer uso de la herramienta?

Los estudiantes manifiestan que pudieron hacer el envío de los datos con al servidor desplegado, dentro de las dificultades mencionan que no fue tan fácil encontrar la documentación relevante para el desarrollo del proyecto y para hacer uso de las algunas funcionalidades de la plataforma.

- ¿Qué elementos consideran que pueden facilitar el uso de la herramienta dentro de las asignaturas?

De forma general los estudiantes manifiestan que les hubiera gustado con una guía orientada a la institución en la que se haga explicación de la forma en la que se envía información a la plataforma, cómo se puede visualizar y cómo se puede enviar información al dispositivo desde la plataforma de las funcionalidades de la plataforma, se muestren ejemplos y se haga referencia a la documentación oficial.

- ¿Considera útil el uso de la herramienta dentro de las asignaturas?

Del curso entrevistado 9 de 10 estudiantes manifiestan que la herramienta puede ser de gran utilidad para obtener más conocimiento a través de las implementaciones y que funciona muy bien para el ámbito académico.

## 6.5. Formulario conocimientos previos IoT

# Encuesta sobre herramientas y laboratorios de Internet de las Cosas

Estimado estudiante,

Desde el Semillero de Investigación en Radio-propagación y Telecomunicaciones (SIRT), se está trabajando en un proyecto de grado que busca adaptar herramientas de código abierto que permitan el desarrollo de prácticas en clase y proyectos de investigación formativa con Internet de las Cosas (Internet of Things, IoT) en nuestra universidad. La tecnología IoT es clave para el desarrollo de proyectos innovadores en diferentes áreas y creemos que su uso podría ser de gran beneficio para la formación de los estudiantes.

En este sentido, les solicitamos su colaboración para responder a la siguiente encuesta, que tiene como objetivo conocer sus conocimientos y necesidades en cuanto a herramientas y laboratorios de IoT. La información que recolectemos nos ayudará a entender mejor sus intereses y necesidades, y a diseñar un plan para la implementación de laboratorios de IoT en la universidad.

Agradecemos de antemano su colaboración, ya que su participación es fundamental para el éxito de este proyecto.

1. ¿Usted conoce qué es IoT?

Sí

No

2. ¿Tiene experiencia previa en proyectos de IoT?

Sí

No

3. ¿Qué herramientas de IoT ha utilizado en proyectos anteriores?

Por favor ingrese las herramientas separadas por comas.

4. ¿Cuáles son los principales desafíos que ha enfrentado al desarrollar proyectos de IoT?

5. ¿Le gustaría tener acceso a herramientas de IoT en la universidad?

Sí

No

6. ¿Considera que contar con laboratorios de IoT en la universidad le ayudaría a agilizar el proceso de desarrollo de proyectos relacionados con tecnologías de conectividad y dispositivos inteligentes?

Sí

No

7. ¿Qué elementos le gustaría que incluyera un laboratorio de IoT en la universidad?

## 6.6. Formulario encuesta de satisfacción

# Encuesta de satisfacción vocero

Nos gustaría conocer su opinión para seguir mejorando la manera en la que se trasmite la información y que esta llegue de manera fácil y oportuna.

\* Obligatoria

\* Este formulario registrará su nombre, escriba su nombre.

1. Indique su nivel de satisfacción con toda la información en general entregada por el vocero del evento \*

1	2	3	4	5
---	---	---	---	---

2. ¿Cree que la información suministrada por el vocero es util en el sector de las TIC? \*

1	2	3	4	5
---	---	---	---	---

3. ¿Le gustaría que en el Poli se utilizara la herramienta Thingsboard para apoyar los proyectos de aula? \*

Sí

No

4. ¿Qué le ha parecido el evento en cuanto a ? \*

	1	2	3
Orden del contenido	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Explicación	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Practica y demostración	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Casos de Uso	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Dominio del tema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

5. ¿ Tiene algún comentario o sugerencia adicional sobre el contenido entregado ?

## 6.7. Acceso para docentes

### GUIA ACCESO A THINGSBOARD

**Objetivo:** Conocer la forma de acceder a la instancia de Thingsboard desplegada en la universidad.

#### Primera parte acceso a la aplicación:

La herramienta se despliega en un raspberry pi, para acceder al dispositivo y verificar el funcionamiento, inicializar o actualizar la aplicación, se puede acceder por anyDesk utilizando las siguientes credenciales:

**Dirección remota:** xxxx

**Contraseña:** xxxx

El acceso a la aplicación se hace por la dirección `academiaco.poligran.edu.co` utilizando el puerto 2214, para utilizar los servicios de la aplicación se hace un mapeo de puertos que se muestra en la Ilustración 22.

Para acceder a la aplicación se utilizan las siguientes credenciales:

**Usuario:** xxxx

**Contraseña:** xxxx

Desde este usuario se puede ver un tablero con toda la información de la aplicación, uso de la cpu, número de usuarios, dispositivos conectados entre otros.

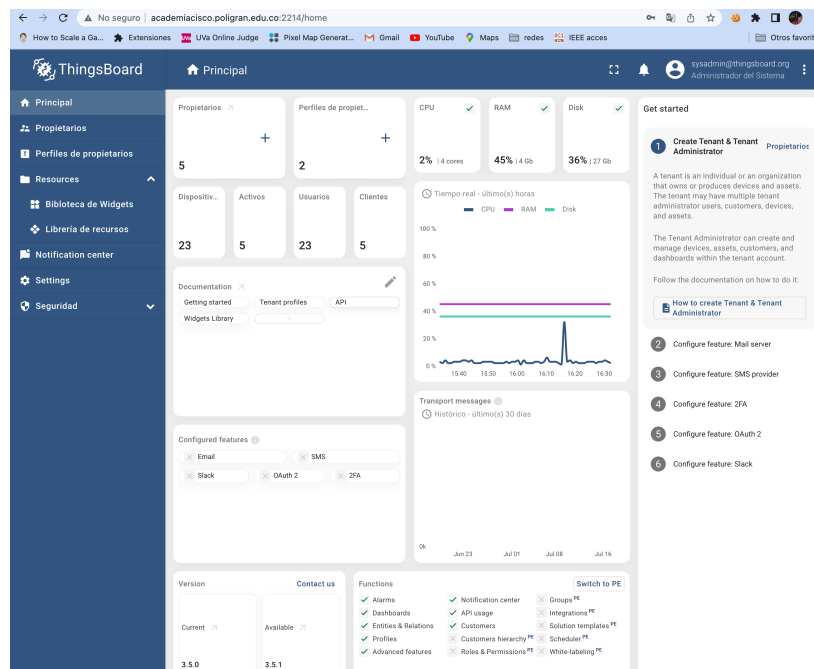


Ilustración 30. Panel de administrador Thingsboard

## Segunda parte iniciar servicio:

En caso de que sea necesario ejecutar de nuevo la aplicación se debe ingresar al servidor como se explica el punto anterior, después de esto hay que inicializar el servicio de base de datos y el servicio de la aplicación esto se puede hacer con los siguientes comandos.

```
sudo service postgresql start
```

```
sudo service thingsboard start
```

Si hay algún problema en el servidor se puede ingresar al directorio de logs para verificar que sucede, todo está en el directorio **/var/log/thingsboard**, para verificar si hay algún error por parte del servidor se puede utilizar el siguiente comando.

```
cat /var/log/thingsboard/thingsboard.log | grep ERROR
```

## Tercera parte creación de usuarios

Para darle acceso a los estudiantes se pueden crear propietarios que harían alusión a los grupos de trabajo de cada asignatura y después asignar como administradores a los estudiantes, en la siguiente imagen se puede ver la creación del propietario.

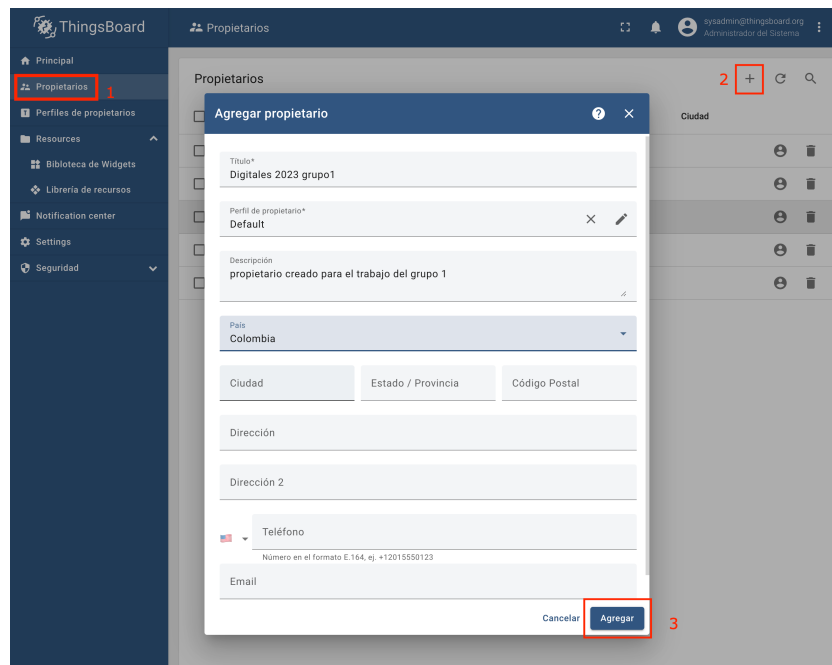


Ilustración 31. Creación de propietario Thingsboard

Después de esto se selecciona el propietario anteriormente creado y se va a la sección de administración de propietarios para dar acceso a los estudiantes.

<input type="checkbox"/>	Fecha de creación ↓	Título	Perfil de propietario	Email	Pais	Ciudad		
<input type="checkbox"/>	2023-05-23 18:37:26	Digitales 2023 grupo 3	Default					
<input type="checkbox"/>	2023-05-23 18:37:08	Digitales 2023 grupo 2	Default					
<input type="checkbox"/>	2023-05-23 18:36:24	Digitales 2023 grupo1	Default					
<input type="checkbox"/>	2023-05-17 12:01:50	Exposicion semanal	Default					
<input type="checkbox"/>	2023-05-11 15:45:53	Tenant	Default					

Ilustración 32. Agregar administradores de un propietario

Dentro de la gestión de los administradores propietarios se selecciona la opción de añadir y se ingresa la información del estudiante para crear la cuenta de acceso y así los estudiantes ya podrían ingresar a la plataforma y trabajar en sus proyectos.

ThingsBoard Propietarios > Admins propietarios

Digitales 2023 grupo1: Admins propietarios

<input type="checkbox"/>	Fecha de creación ↓	Nombre	Apellido	Email		
<input type="checkbox"/>	2023-05-23 18:38:52	propietario1		propietario1@thingsboard.org		

**Agregar usuario**

Email\*

Nombre

Apellido

Teléfono

Número en el formato E.164, ej. +12015550123

Descripción

Método de activación

Mostrar enlace de activación

Cancelar Agregar

Ilustración 33. Creación de usuario que administra un propietario



## 6.8. Manual funcionalidades Thingsboard

### GUIA FUNCIONALIDADES THINGSBOARD

**Objetivo:** Familiarizarse con las funcionalidades Thingsboard para el envío, recepción y visualización de telemetría generada por dispositivos embebidos.

#### Primera parte acceso a la aplicación:

Para el desarrollo de esta guía se utiliza a aplicación de Thingsboard desplegada dentro de la Universidad, El acceso a la aplicación se hace por la dirección [academiatico.poligran.edu.co](http://academiatico.poligran.edu.co) utilizando el puerto 2214, todo sobre hace sobre el usuario de la semana tic y las credenciales son las siguientes.

**Usuario:** [semanatic@thingsboard.org](mailto:semanatic@thingsboard.org)

**Contraseña:** semanatic

Este es un usuario administrador propietario por lo que tiene la capacidad de crear cualquier tipo de dispositivos o tableros, y puede ver el estado de los dispositivos dentro de su propiedad

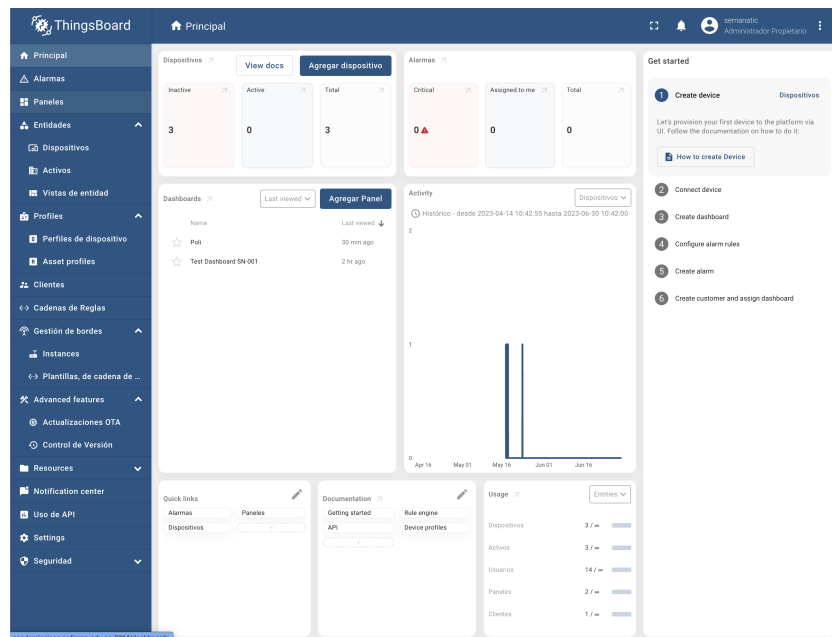
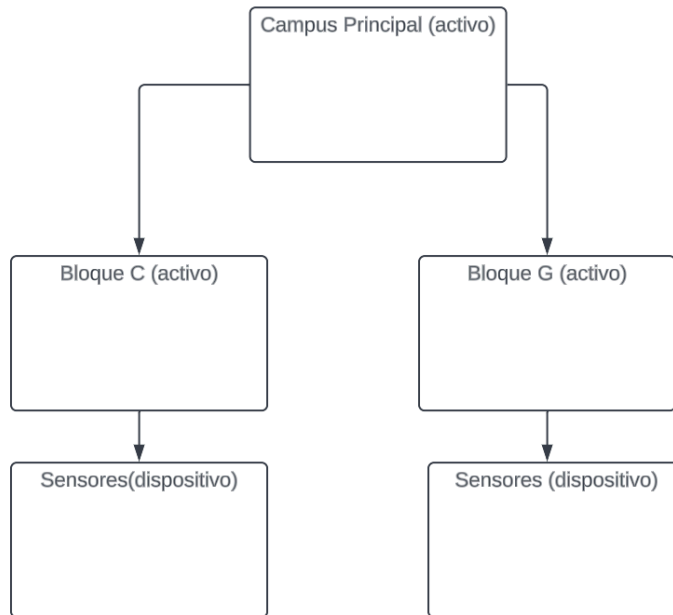


Ilustración 34. Panel de un propietario

## Segunda parte caso de uso:

Para la explicación de la herramienta se plantea ver dos edificios del campus principal de poli y poder ver la telemetría generada por los dispositivos de cada edificio, para esto se puede usar la función de la aplicación de relacionar activos con otros activos o con dispositivos para tener la siguiente relación.



*Ilustración 35. Relación de entidades Manual*

Teniendo clara la relación planteada, el objetivo es tener toda la información en un mismo tablero donde se pueda navegar entre los bloques y entre los dispositivos.

### Tercera parte creación de tableros:

Lo primero que se debe hacer es crear los activos asociados a lo que se quiere representar, en este caso el Campus Principal y los dos bloques.

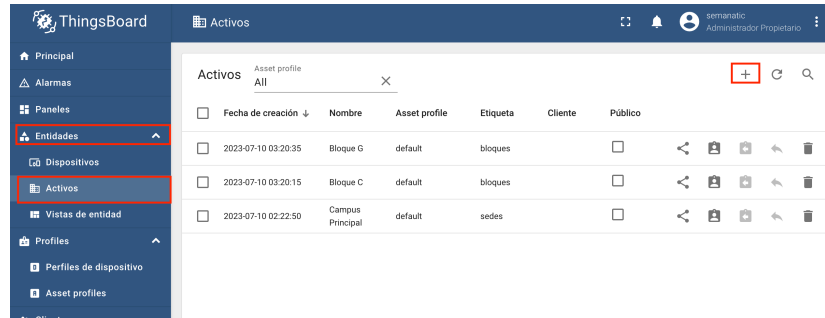


Ilustración 36.Manual, creación de activos

Una vez creadas las entidades se deben establecer las relaciones, para esto se le va a agregar la relación al Campus Principal que contienen los dos bloques, se ingresa al detalle desde la ventana de activos y se asigna.

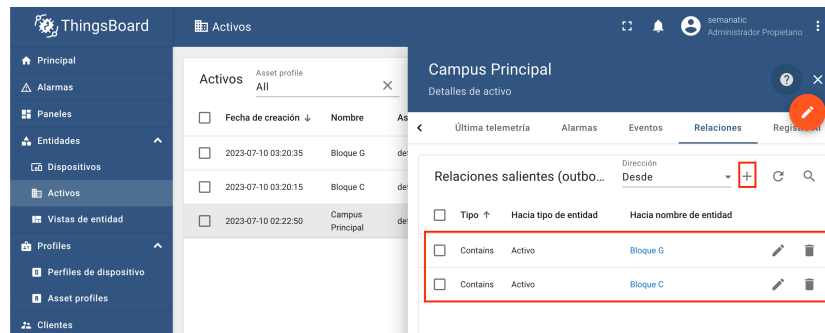


Ilustración 37.Manual, asociar activos

Con los activos creados y asociados el siguiente paso es crear los dispositivos que van a estar asociados a cada bloque.

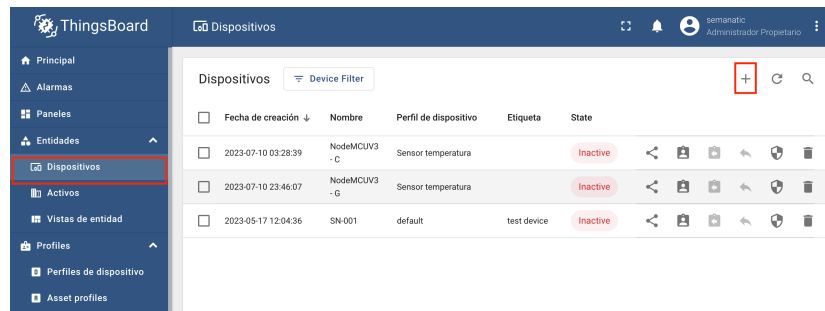


Ilustración 38.Manual, crear dispositivos

Una vez se hayan creado los dispositivos se debe volver a la sección de activos para asociar los dispositivos a los bloques correspondientes.

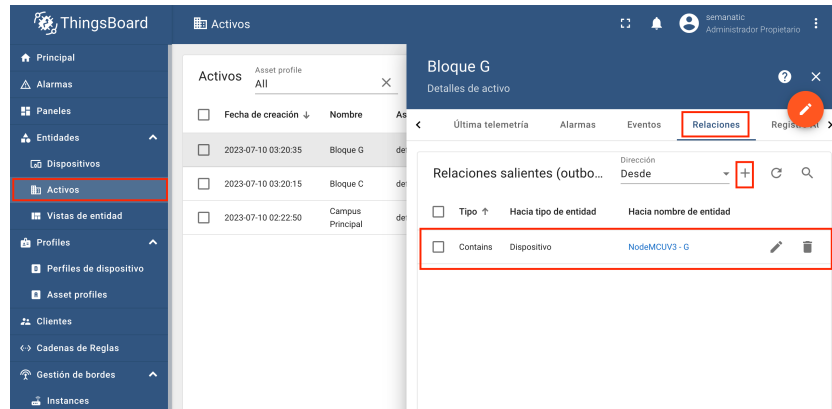


Ilustración 39.Manual, asociar dispositivos

La creación de estas relaciones y entidades va a permitir realizar la creación de los tableros que permite navegar entre los dispositivos de los bloques. Para poder identificar los bloques de forma visual se añaden los atributos de latitud y longitud a los activos.

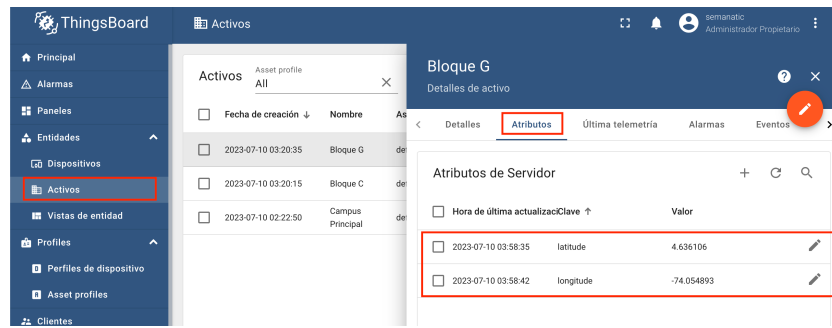


Ilustración 40.Manual, crear atributos de servidor

Añadiendo esta información ya se pueden empezar a generar los tableros, para esto se va a la sección de paneles y se crea uno nuevo.

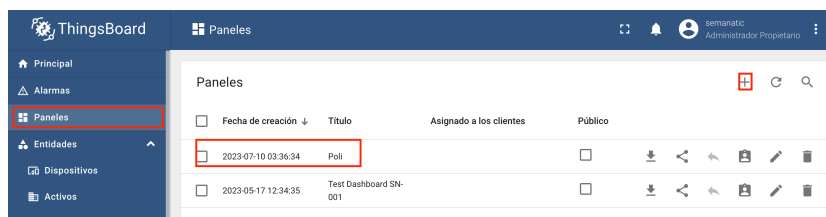


Ilustración 41.Manual, crear tablero

Para empezar a trabajar en el tablero se crea un alias que permite agrupar los bloques del Campus anteriormente creados.

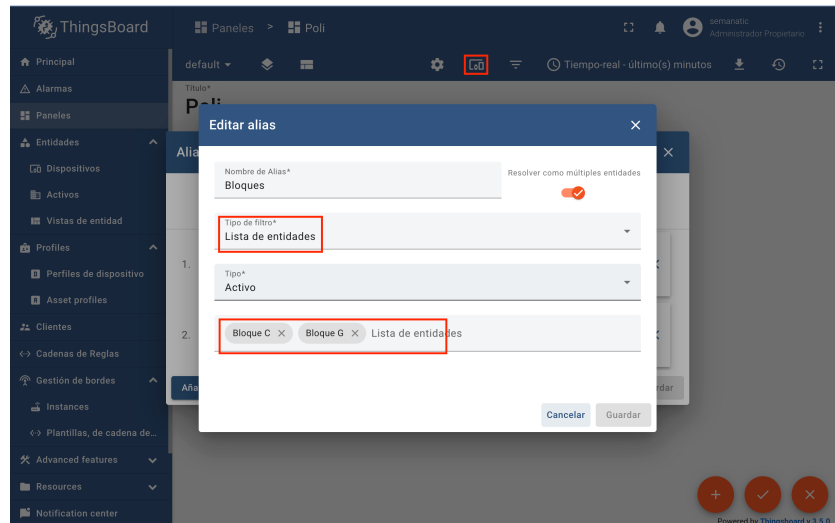


Ilustración 42. Manual, crear Alias activos

Hecho se procede a crear el primer widget, este va a ser tipo mapa y el set de datos que se va a seleccionar es el Alias anteriormente creado “Bloques”, y se va a mapear la información de los atributos asociados a latitud y longitud.

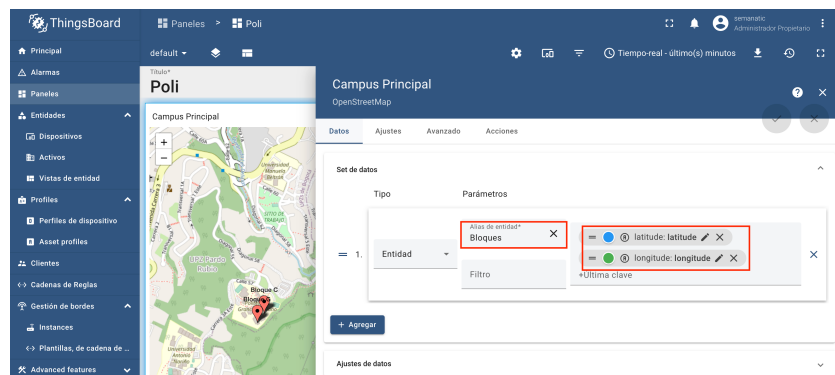


Ilustración 43. Manual, crear mapa por Alias

Adicionalmente se va a crear otro widget de tipo tabla de Entidades, acá también se va a utilizar como fuente de información el Alias “Bloques”.

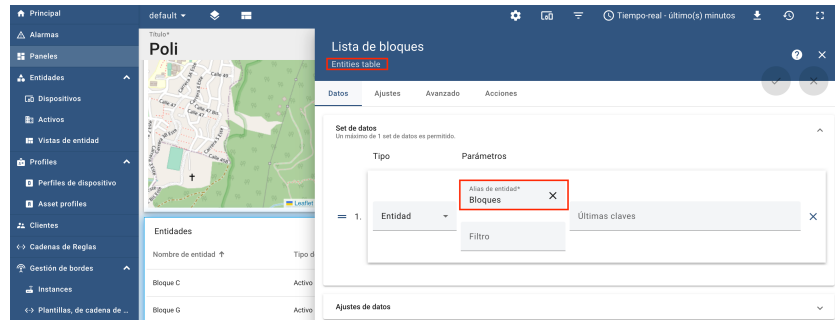


Ilustración 44. Manual, creación lista de bloques campus principal

Ahora para poder navegar al detalle de los bloques y ver los dispositivos asociados, es necesario crear un sub-estado del tablero, se utiliza la notación  $\{entityName\}$  en el nombre del sub-estado para crearlo de forma dinámica.

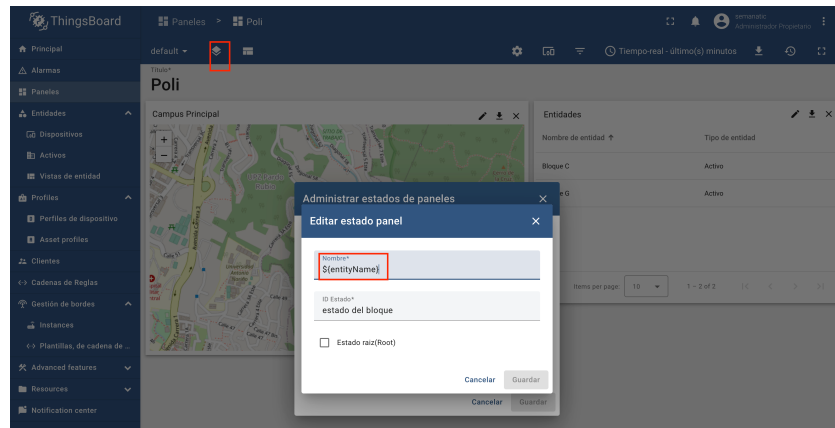


Ilustración 45. Manual, creación sub-estado por bloques del campus

Con el sub-estado creado, lo siguiente es asociar el mismo a una acción, para esto volvemos al widget que contiene la lista de los bloques y vamos a ir a la sección “acciones”, acá creamos una acción que se active al hacer clic sobre una fila y que nos envíe al sub-estado del bloque seleccionado.

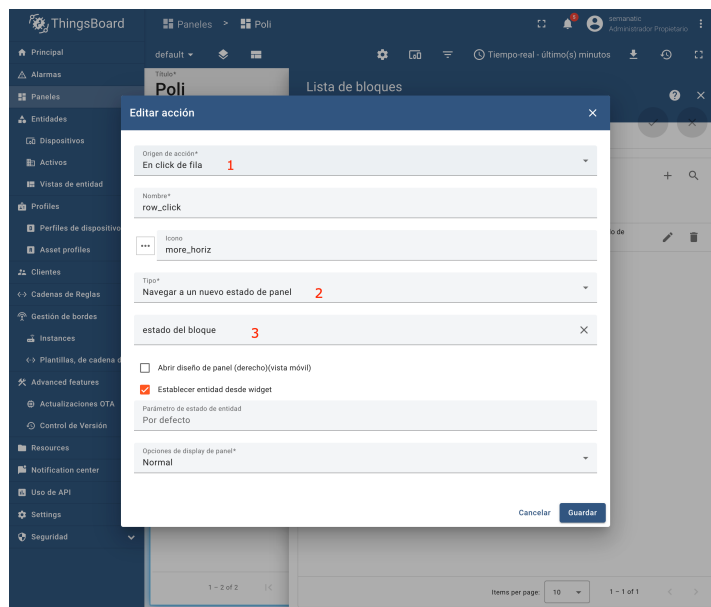


Ilustración 46.Manual, crear acción para sub-estado de dispositivos

Antes de utilizar la acción antes creada se crea un nuevo Alias que va a servir para identificar los dispositivos que están asociados a un bloque.

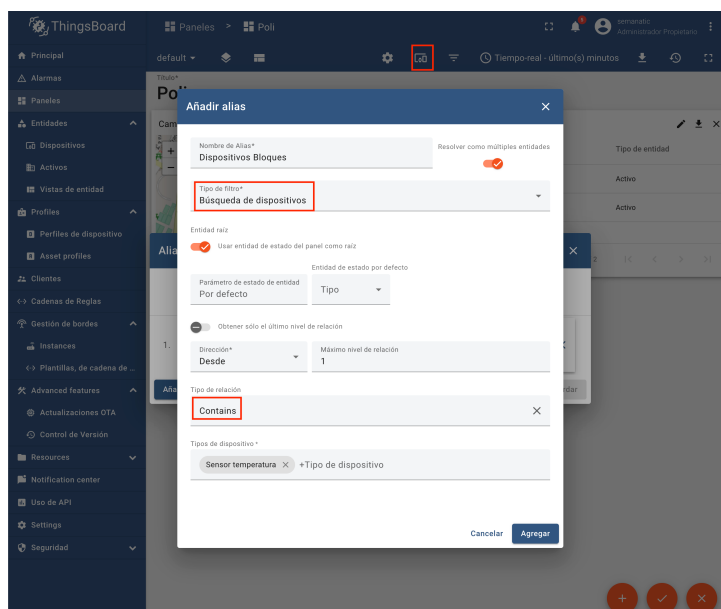
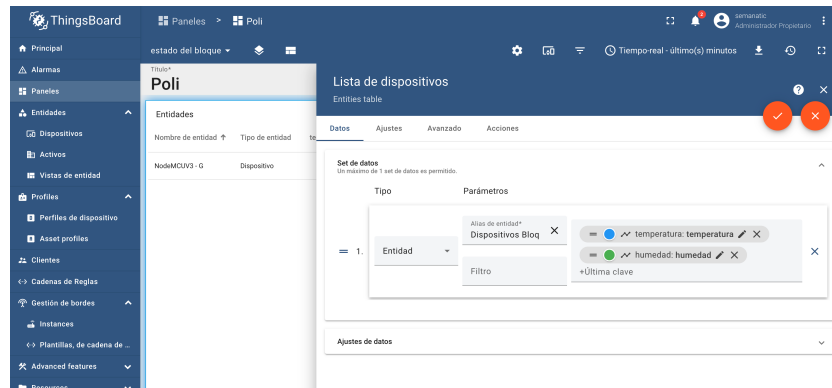


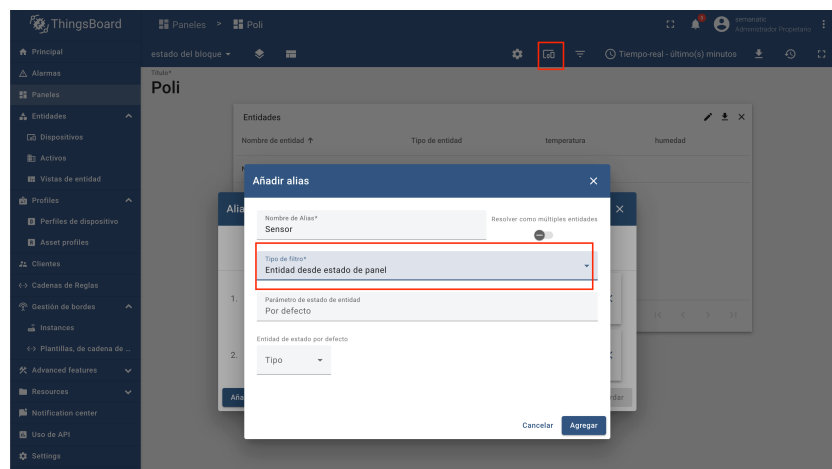
Ilustración 47.Manual, crear Alias para dispositivos de los bloques

Ahora ingresamos al sub-estado del bloque y crea un nuevo widget para mostrar la información de los dispositivos, para esto se usa el Alias recién creado “Dispositivos Bloques” y se mapea la telemetría de humedad y temperatura.



*Ilustración 48. Manual, creación de dispositivos asociados a un bloque*

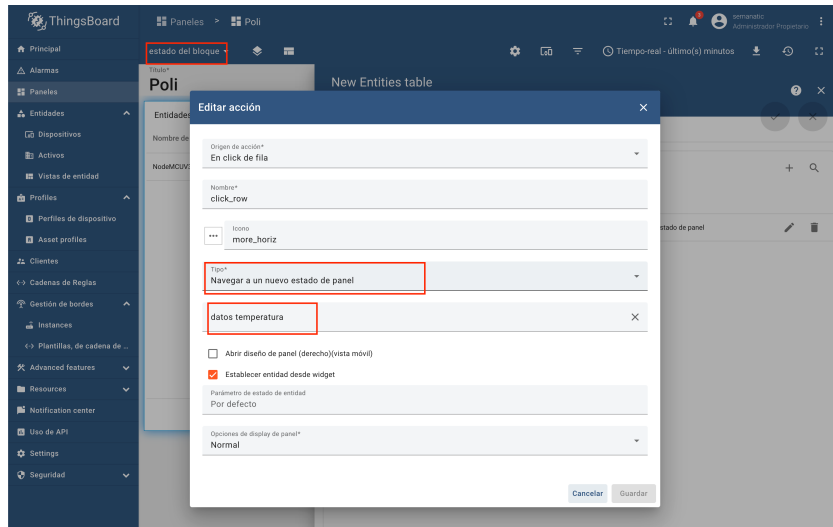
Ahora se crea un nuevo Alias asociado a los sensores que sería la información final, y poder ver finalmente los datos de telemetría.



*Ilustración 49. Manual, Alias para el dispositivo final*

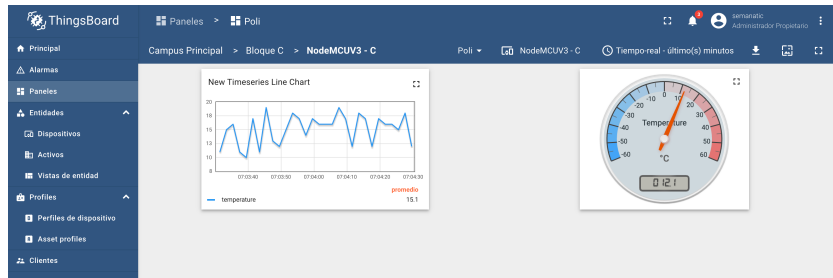
Con esto hecho se crea un nuevo sub-estado que va a contener la información asociada a cada sensor y se le añade el sub-estado a la acción de darle clic en el sensor para que muestre la información de la telemetría.





*Ilustración 50.Manual, acción para navegar a un panel del detalle de los sensores*

Por último, al entrar en el estado final se los sensores se pueden agregar widgets específicos para visualizar la telemetría utilizando el Alias “Sensor” que trae la información del dispositivo con el que se ingresa al panel.



*Ilustración 51.Manual, graficas específicas de los dispositivos embebidos.*

Para enviar información a los dispositivos se puede usar el mecanismo descrito en la sección 6.2.