EXPLORACIÓN DE LAS PLATAFORMAS IoT EN EL MERCADO PARA FOMENTAR EL CONOCIMIENTO, BUEN USO Y EFECTIVIDAD DE LOS DISPOSITIVOS IoT CREADOS EN LA FACULTAD DE INGENIERIA Y CIENCIAS BASICAS DE LA INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO.

JESSICA JULIETH MANOTAS CAMPOS

NICOLAS MARTINEZ MARIN

INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO

FACULTAD DE INGENIERÍA Y CIENCIAS BASICAS

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES

BOGOTÁ D.C.

2018

.

EXPLORACIÓN DE LAS PLATAFORMAS IoT EN EL MERCADO PARA FOMENTAR EL CONOCIMIENTO, BUEN USO Y EFECTIVIDAD DE LOS DISPOSITIVOS IoT CREADOS EN LA FACULTAD DE INGENIERIA Y CIENCIAS BASICAS DE LA INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO.

JESSICA JULIETH MANOTAS CAMPOS

NICOLAS MARTINEZ MARIN

GIOVANNY ANDRES PIEDRAHITA SOLORZANO

Trabajo de grado presentado a la Institución Universitaria Politécnico Grancolombiano de las carreras de Ingeniería en Telecomunicaciones e Ingeniería de Sistemas como requisito para la obtención del título de ingeniera en telecomunicaciones e ingeniero de sistemas.

INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO

FACULTAD DE INGENIERÍA Y CIENCIAS BASICAS

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y TELECOMUNICACIONES

BOGOTÁ D.C.

2018

.

Contenido

pág.

.

.

LISTA DE ILUSTRACIONES

Pág

.

LISTAS DE FIGURAS

Pág

LISTA DE TABLAS

Pág

.

Resumen

En este documento va a encontrar una descripción de arquitecturas (plataformas) de sistemas IoT que desde hace poco se han consensuado por expertos como los diseños que facilitan la implementación de maquinaria IoT, Además, encontrara un análisis de cómo se localizan, adquieren e implementan con un caso estudio los modelos expuestos de plataformas IoT que ofrece el mercado actualmente.

Palabras Clave

Innovación, sistema, plataformas y selección.

1. Introducción

Actualmente, el uso de lo que se conoce como IoT (Internet de las cosas) está aumentando de manera exponencial, debido a que se puede aplicar en diferentes campos de la vida cotidiana, usando un par de sensores u otros dispositivos que faciliten la adquisición de datos en el mundo real para su posterior tratamiento, pero con esto también aumentan las incertidumbres sobre la arquitectura y el desarrollo de los dispositivos para un buen funcionamiento, lo que nos lleva a cuestionarnos sobre las características, propiedades y actualizaciones de estas nuevas tecnologías.

Internet de las cosas es un término famoso en el área de la innovación y tecnología; una de sus características principales es la plataforma IoT, existen diferentes en el mercado, es una herramienta que permite el intercambio de mensajes entre los componentes de un sistema IoT

.

como, por ejemplo, hardware, puntos de acceso, redes de datos el aplicativo del usuario, entre otros…, con el fin de generar un propio ecosistema de conexión y permitir una gestión asequible, un análisis y visualización de datos. Es por esta razón que las plataformas IoT son tan significativas para el crecimiento de la nueva tecnología en los diferentes mercados en donde se usa IoT.

Derivado de un trabajo ejercido en el semillero de investigación redes y seguridad de la información el anterior semestre (penúltimo semestre de las carreras cursadas), se desarrolló esta investigación sobre plataformas IoT que es una de las características mas importantes para esta nueva modalidad de interconexión de objetos puesto que se encuentran diversas y disímiles fuentes de información.

El semillero influyo una contextualización sobre la importancia de conocer el manejo de las plataformas IoT. Se evidencio la falta de conocimiento sobre estas herramientas y que era necesario profundizar en el tema, aun mas en la composición de las arquitecturas y el modo de configurarlas para lograr convertir el proceso de construcción de aparatos IoT en una actividad formativa y con sentido.

Por lo tanto, la finalidad de la búsqueda es estudiar algunas plataformas existentes en el mercado para lograr crear un documento de trabajo de grado donde se describa cada una con sus condiciones, particularidades, además es importante incluir comparaciones de las plataformas teniendo en cuenta distintivos de las propiedades significativas, esto para concluir y escoger con criterio la plataforma que más se ajusta a las necesidades de un prototipo de dispositivo IoT creado en la Facultad de Ingeniería y Ciencias Básicas de la Institución Universitaria Politecnico Grancolombiano en el semillero de investigación redes y seguridad de la información, este dispositivo nació de ideas igualmente de un proyecto de aula de la materia de seguridad de la información, el nombre del dispositivo es; medidor de temperatura y humedad con GPS.

.

2. Justificación

Existen todo un mundo de variedades en cuanto a plataformas IoT, eso es positivo porque de alguna manera podemos escoger entre diferentes arquitecturas, pero no es fácil, porque de tanta variedad nacen las dudas al momento de elegir. Además, no es tan evidente conocer las diferencias entre; las cualidades y por ende los defectos y otras características.

La ejecución de este proyecto permite desarrollar un criterio de selección en el ámbito de herramientas tecnológicas; se es capaz de comparar objetivamente diferentes propuestas del mercado, se trabaja dicho criterio al examinar detalladamente las plataformas de IoT seleccionadas con el fin de determinar cuál de estas puede suplir las necesidades de la mejor manera para cada proyecto.

El lograr que la institución cuente con un listado de plataformas categorizada por sus características y propiedades, capaces de trabajar con dispositivos IoT supondría un valor agregado para los estudiantes al poder familiarizarse con una tecnología emergente y en un actual crecimiento.

3. Objetivo general

Evaluar plataformas de IoT existentes en el mercado para identificar aquellas que cumplan los requerimientos de captura y procesamiento de datos de sensores instalados en el campus del Politecnico Grancolombiano, y validar su funcionamiento a través de la implementación y prueba de un prototipo.

.

3.1. Objetivos específicos

- Investigar sobre arquitecturas del sistema IoT haciendo énfasis en sus tipologías y propiedades.

- Evaluar de manera técnica las plataformas IoT, así permitiendo una actividad comparativa de manera equitativa.

- Indicar las propiedades de las plataformas estudiadas, para que las personas cuando accedan al documento tomen una decisión acertada sobre la mejor que se ajusta.

- Mostrar la implementación de la plataforma en el prototipo, que es el dispositivo de medición climática con ubicación geográfica realizado en el semillero, para así poder enseñar el buen uso de la plataforma.

4. Marco teórico

El concepto de Internet de las cosas, según IES-Business School (2017), "se basa en la interconexión de cualquier producto con cualquier otro de su alrededor, es decir, conectar cualquier objeto y elemento a través de Internet" [29], es decir que IoT es una reciente modalidad de negocio, que permite el crecimiento global en cuanto a innovación y tecnología para cualquier sector económico-social.

.

Existen cuatro características primordiales que permiten la eficacia cuando se habla de Internet de las cosas y son las siguientes: placas de prototipado (Arduino y Raspberry), abaratamiento de las comunicaciones (disminución del precio del servicio), sensores (productos disponibles a bajo precio) y por último plataformas IoT.

4.1. Plataformas IoT

Para explicar de una mejor manera el funcionamiento del sistema de Internet de las cosas, y el papel que desarrolla la plataforma IoT en él, se muestra a continuación, la figura 1 en la cual se presenta el área en el que está implicada la plataforma en cualquier proyecto sobre Internet de las cosas.

Se entiende por medio de la figura que el grupo de los sensores no está directamente implicado con las plataformas y aunque en la figura no es evidente los protocolos de red (LAN, Bluetooth, etc) tampoco, por lo tanto, cuando se esté programando o configurando cualquier plataforma IoT, solo se tendrán en cuanta las otras variables, exceptuando estas dos.

.

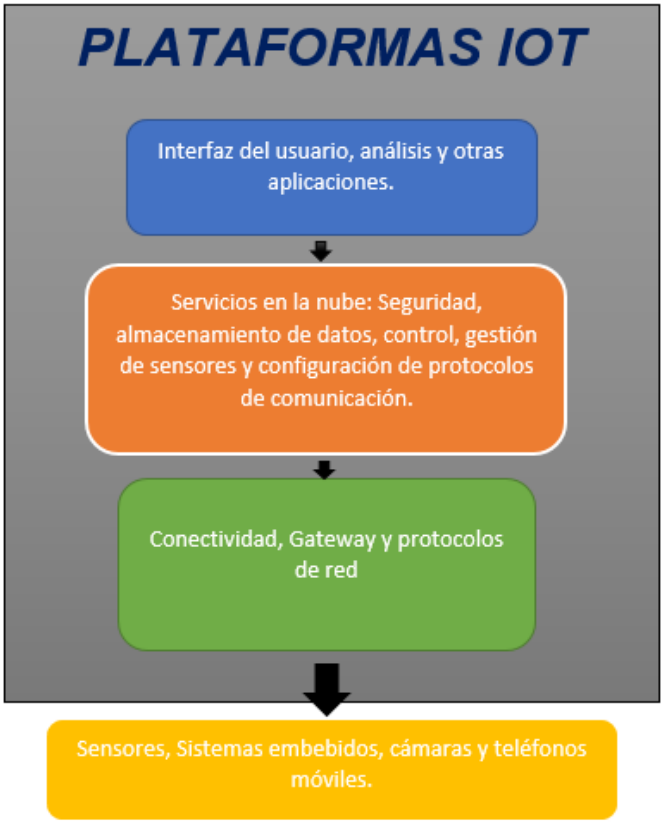


Figura 1 *Plataforma IoT en un sistema de Internet de las cosas. (Elaboración propia)*

De lo anterior cabe resaltar que para que la conectividad funcione, tiene que existir una adaptación de los protocolos de comunicación en la parte de servicios en la nube, ya que, en conectividad existiría la configuración propia del protocolo que se use para la comunicación de internet, como, por ejemplo; HTTP, TCP o UDP … etc.

4.1.1. Microsoft Azure IoT Hub.

Esta plataforma consiste en un servicio alojado en la nube que cuenta con un modelo capaz de ofrecerle al usuario soluciones predeterminadas de sistemas Iot para agilizar la creación de estos

.

o brindar soluciones personalizadas que se acomoden a las necesidades del modelo de negocio de los usuarios. El uso de esta plataforma facilita la conexión de un alto volumen de dispositivos IoT, el procesamiento y análisis de datos y por último la presentación del sistema como la conexión con el negocio.

La plataforma principalmente ofrece dos paquetes de servicios los cuales son IoT Suite Y IoT Central cada uno enfocado en cumplir las metas anteriormente descritas, pero con diferentes modos operacionales.

4.1.1.1. IoT Suite.

Este servicio está enfocado en las soluciones que requieren un alto grado de personalización ofreciendo una gran flexibilidad en el manejo de la infraestructura y de servicios web, ofreciendo la personalización tanto en el despliegue arquitectónico como en la configuración de microservicios al ser de código abierto. Para realizar estas configuraciones se requieren conocimientos en java o .Net y en JavaScrip.

4.1.1.2. IoT Central.

Este servicio está enfocado en soluciones preconfiguradas que no necesitan una personalización extensa, de igual forma el código de los servicios no está expuesto a cambios y la infraestructura tampoco es personalizable. Para este modelo no se necesitan habilidades en programación y las plantillas existentes permiten realizar las configuraciones tiempos cortos.

.

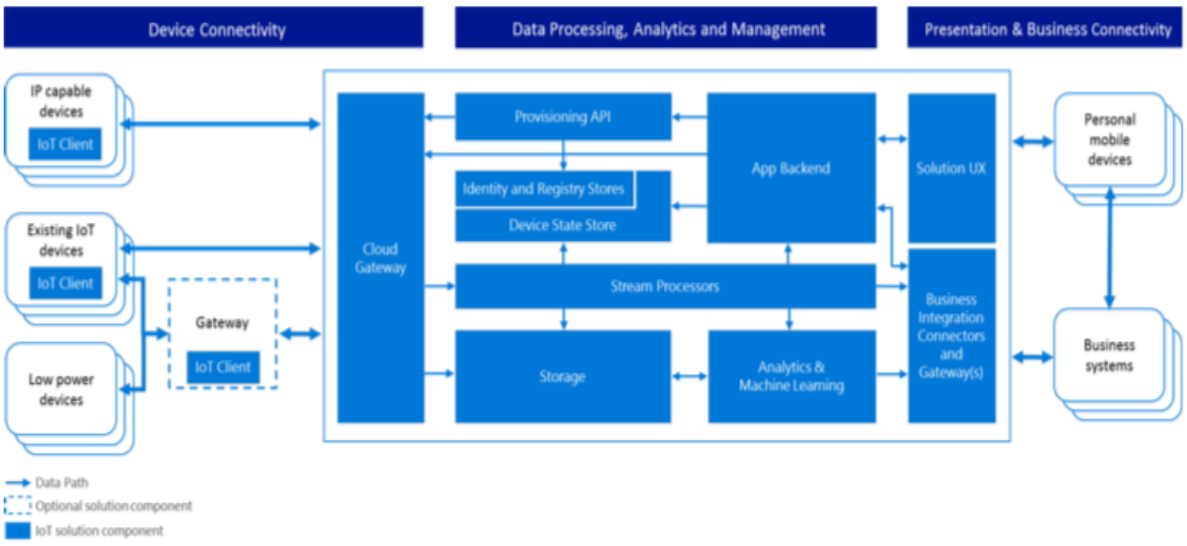


Ilustración 1 *Componentes de Microsoft Azure IoT (Imagen tomada de Microsoft Azure IoT Services Reference Architecture)*

Como se puede observar en la ilustración [1] el modelo arquitectónico implementado se basa en una conexión de los dispositivos IoT (si son capaces de obtener una Ip se conectan directamente de lo contrario utilizan un puente para conectarse al servidor) con un servidor encargado de recibir las conexiones de los dispositivos y encárguese de hacer su respectivo registro y/o verificación con los componentes de identidad, una vez hecho esto nos encontramos con un servidor que me permite la transmisión de simultánea de datos vía Streaming para conectar la recolección de datos con su debido procesamiento, una vez se ha terminado el procesamiento de datos tenemos un servidor dedicado a garantizar el acoplamiento del sistema del negocio con el sistema IoT y finalmente está el servidor encargado a interactuar con el usuario.

.

4.1.2. Thinger.io.

Es una plataforma de código abierto que ofrece la conexión de dispositivos IoT ya sea para la obtención de datos por medio de sensores o para enviar datos a través de internet para ser mostrados en el portal principal o en otra aplicación con la cual el portal sea capaz de establecer comunicación como lo hace con algunas redes sociales.

La plataforma tiene una versión gratuita que permite configurar tres dispositivos y dos tableros de control que permite mostrar los datos en 10 tipos de pantalla diferentes, en la siguiente ilustración [2] se puede observar la interfaz.

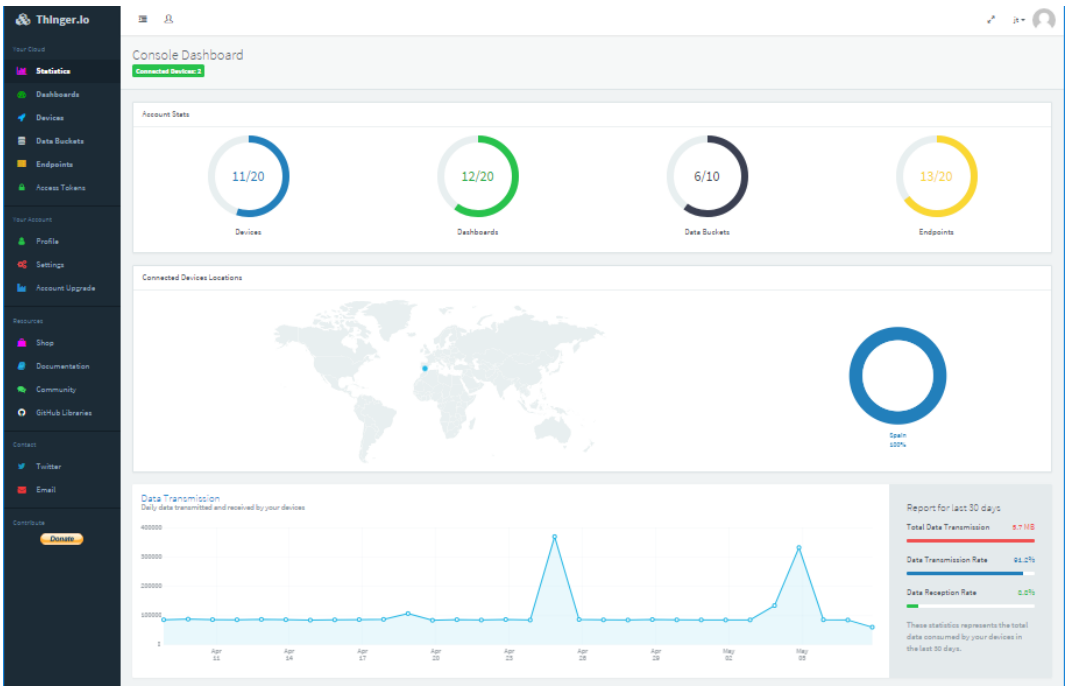


Ilustración 2 *Interfaz Thinger.io (Imagen tomada de https://thinger.io)*

.

Para hacer uso del sistema se configura un dispositivo IoT que sea capaz de enviar datos por HTTP ya sea directamente o con ayuda de un Gateway, después de esto tenemos la opción de enviar los datos del sensor en formato Json para el almacenamiento de Buckets y su visualización en las diferentes pantallas.

4.1.3. Carriots.

Carriots es una plataforma como servicio diseñada para proyectos de IoT de máquina a máquina, esto quiere decir que es una plataforma orientada a conectar un dispositivo a internet y viajar de internet a otra máquina. Por esta razón, esta plataforma permite integrar los dispositivos de IoT a una aplicación externa que requiera de los datos mientras ellos se encargan del almacenamiento y la comunicación.

Esta plataforma cuenta con un Rest API y un código SDK en java que facilita la integración con las aplicaciones haciendo uso de una arquitectura de siete capaz que hace posible todo el funcionamiento. La arquitectura implantada por los creadores de Carriots se muestra en la Ilustración [3].

.

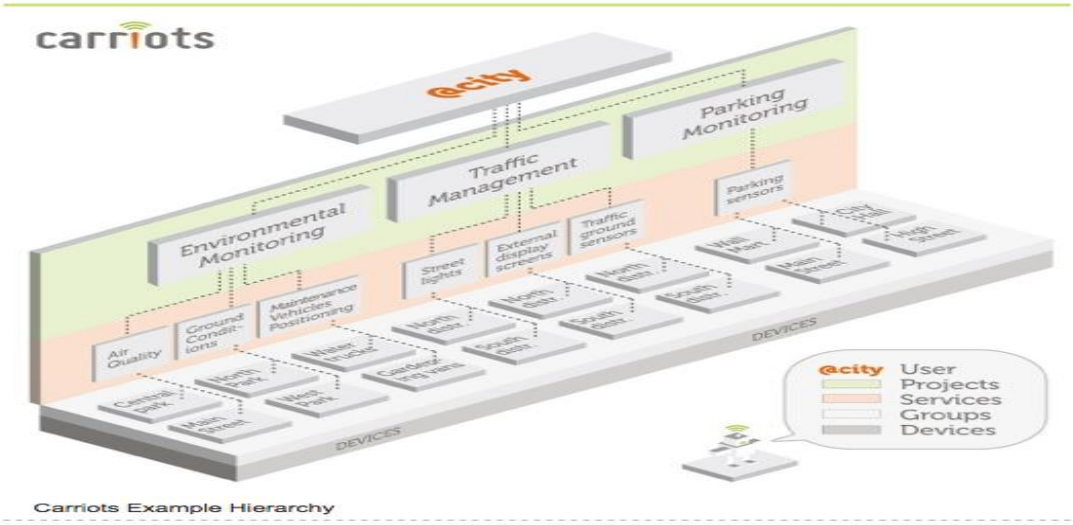


Ilustración 3 *Arquitectura de Carriot (Imagen tomada de https://www.carriots.com/)*

4.1.4. Kaa IoT.

La definición según los desarrolladores de Kaa "Es una plataforma de código abierto que proporciona la tecnología de software clave para habilitar productos y servicios con capacidad IoT comprobada" [20], con lo anterior se sabe que Kaa es una plataforma que brinda conectividad entre objetos y además como valor agregado tiene herramientas que brindan productos que al final facilitan la gestión de los datos, el control de sensores y la ciber seguridad.

Características: Desarrollo rápido de productos IoT, es independiente del hardware, puede manejar datos estructurados y no estructurados, compatible con Apache Avro, GUI para esquema de datos, librería SDK Endpoint disponible en Java, C++ y C, y están diseñados para integrarse en su aplicación cliente (o Firmware).

.

Igualmente, Kaa IoT también maneja comunicación cliente-servidor, autenticación, clasificación de datos, encriptación, persistencia, en la conectividad este tiene transporte-agnóstico, protocolos conectables y soporte multicanal, por tanto, permite crear aplicaciones que funcionan sobre cualquier tipo de conexión de red, presenta la opción de elegir una de las implementaciones del protocolo de transporte de Kaa o crear personalizados.

El servidor Kaa se diseñó para manejar millones de dispositivos conectados, es, fácilmente personalizable y ampliable, tiene equilibrio de carga en el clúster, no hay un solo punto de falla y tiene múltiples opciones de implementación (local, en la nube o mixta).

El procesamiento de datos Kaa es pre integrado con los populares sistemas de procesamiento de datos, posee herramientas que permiten la visualización de datos y es escalable para manipular grandes cantidades de datos.

4.1.5. Macchina.io.

Es una plataforma de software completa y potente para dispositivos IoT inteligentes que se conectan a sensores, otros dispositivos y servicios en la nube. Las aplicaciones de IoT se ejecutan en dispositivos basados en Linux, su entorno en bases de código es de C++ y JavaScript habilitado para web, seguro, modular y extensible.

Características: Proporciona API completas, se basa en un potente servidor de aplicaciones web integrado, el cual brinda interfaces de usuario estéticamente agradables y completas. En la ilustración [4] se puede observar la infraestructura funcional de macchina.io, puede estudiar cada proceso y cada paso que realiza esta plataforma de manera resumida.

.

Macchina.io agrega un motor de JavaScript V8 ayuda a la compilación para el código de máquina ARM, MIPS o x86 nativo generando un mejor rendimiento, la modularidad, extensión y escalabilidad flexible permiten actualizar y ampliar de forma segura agregando funciones a los campos, los protocolos de comunicación que usa macchina.io incluyen soporte para HTTP, MQTT, RESTO, JSON-RPC, MQTT, SOAP, UPnP, Modbus, OPC-UA, CANopen, etc.

Para la base de datos se usa SQLite integradamente es ideal para registrar datos de sensores y disponible en los dos lenguajes de programación. Usa contenedores de aplicación Docker y presenta una gestión remota segura “my-devices.net” igual para el acceso remoto a través de Web, SSH y VNC.

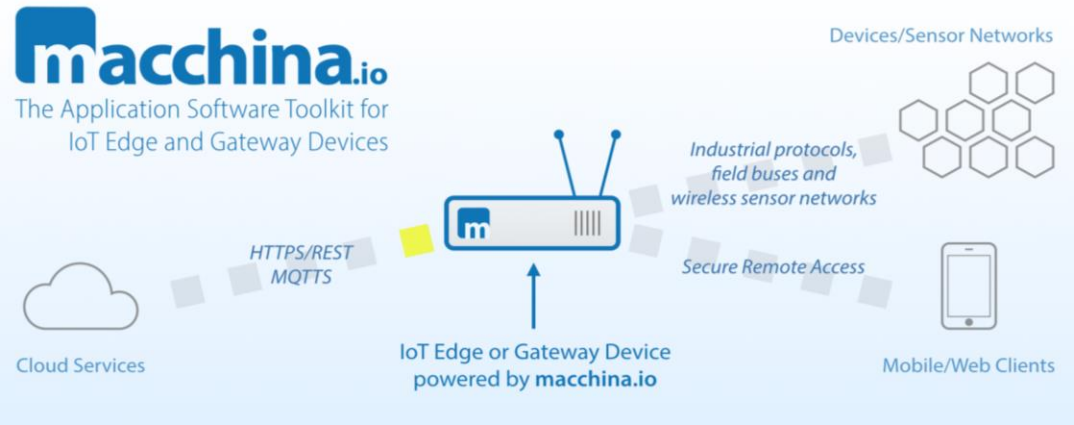


Ilustración 4 *Infraestructura funcional del producto macchina.io (Imagen tomada de https://macchina.io/)*

.

4.1.6. Thingspeak.

Plataforma de código abierto con análisis de MATLAB. Las Funciones principales de la plataforma son: Recoger datos enviados del sensor de forma privada a la nube, permite analizar y visualizar los datos con la herramienta de MATLAB y por último acto disparar una reacción.

Las principales características son: Recopilar datos en canales privados, compartir datos con canales públicos, API RESTful y MQTT, MATLAB analytics, activar alertas, programar eventos, integrar las aplicaciones y comunidad mundial.

Thingspeak funciona con los siguientes dispositivos: Arduino, fotón de partículas, módulo WIFI ESP8266, Raspberry PI, aplicaciones móviles, Twitter, Twilio, MATLAB. Esta plataforma tiene diferentes y destacados proyectos que la hacen reconocida, en la ilustración [5] se pueden ver.

Ilustración 5 *Aplicaciones existentes (Imagen tomada de https://thingspeak.com/)*

.

4.1.7. Ubidots.

Es una herramienta de recopilación de datos, análisis y visualización, en la nube listas para producción. Esta plataforma posee ventajas que la destacan de las demás, en la ilustración [6] se presentan.

Ubidots tiene las siguientes características; API y protocolos que se puede conectar de cualquier hardware a Ubidots Cloud por medio de HTTP, MQTT, TCP, UDP o Parse (protocolo personalizado). En cuanto a Dashboard se pueden analizar datos en tiempo real, porque la plataforma crea cuadros, por lo tanto, se pueden controlar los dispositivos.

También la plataforma posee una gestión de usuario, la cual ayuda a tener un orden en cuanto a permisos y restricciones dependiendo del operador final. Otras características son; comunicación con los sensores (entrada/salida) permite la creación API, extendiendo el monitoreo y análisis de datos de las aplicaciones de API de todo tipo. Por otro lado, posee un almacenamiento y Back-End el cual permite visualizar los datos todos los días durante dos años de retención en la fuente, también presentando un buen plan para el mantenimiento predictivo. La salida de datos se le llama informes programados y se entregan en formatos PDF o Excel y también se puede programar una entrega a cualquiera que los necesite.

.

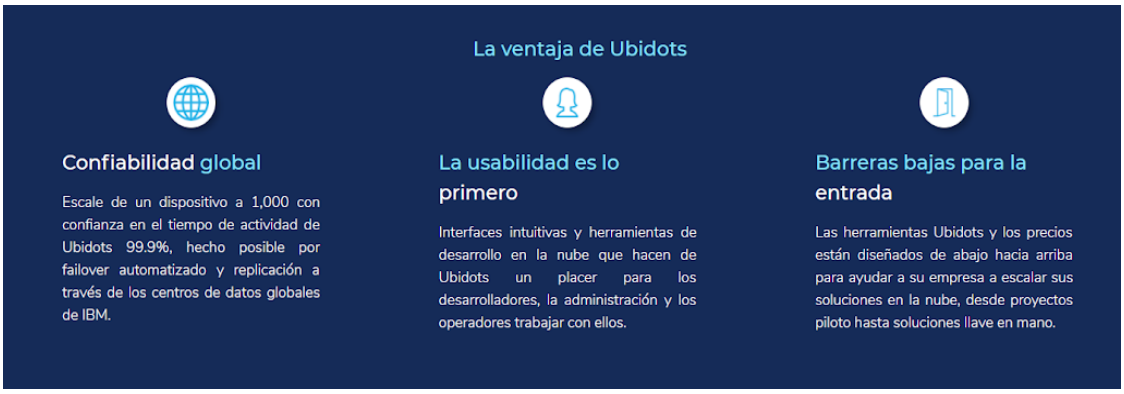


Ilustración 6 *Ventajas de la plataforma (Imagen tomada de https://ubidots.com/)*

4.1.8. MyDevices.

Es una herramienta para diseñar, prototipar y comercializar rápidamente soluciones IoT la plataforma “Cayenne es la que ofrece solución para un manejo fácil de las empresas IoT que conectan sensores y desean visualizar datos de forma inteligente” [25], eso dice Marc Pegulu el vicepresidente.

Características: “Cayenne es increíblemente fácil de usar al crear prototipos de soluciones de IoT. La solución ofrece a los desarrolladores la capacidad de diseñar códigos POC sin código y reducir el tiempo de comercialización” [20] - Tyler Smith gerente de aplicaciones y marketing. Cayenne IoT Ready Program para fabricantes, habilita de forma fácil los microcontroladores, puertas de enlace, sensores y otros dispositivos de la empresa en la que se va a implementar internet de las cosas. Una de las cosas muy positivos de esta plataforma es el largo alcance y la baja potencia y bajo costo para ecosistemas LoRaWAN.

.

4.1.9. InitialState.

Es una herramienta potente con precio asequible. Por otra parte, tiene Analytics para datos de series de tiempo, puede transmitir datos desde los dispositivos y aplicaciones estéticamente bien presentables sus visualizaciones en el navegador web.

Sus características son; puede enviar o subir de forma segura datos obtenidos de sensores, dispositivos, software o cualquier archivo, convierte estos datos en cuadros de mando, formas de onda, gráficas o mapas hasta en Emojis en tiempo real y se pueden compartir. Almacenamiento de los datos para siempre, con esta herramienta se pueden analizar los datos de cualquier día en el momento que se desee. En la ilustración [7] se muestra la cuenta inicial y el estado final del proceso de Initial State.

Ilustración 7 *Proceso Initial State (Imagen tomada de https://www.initialstate.com/)*

4.1.10. Temboo.

Temboo es una plataforma que permite gestionar dispositivos IoT desde diversos lenguajes de programación y además configurar todos los posibles componentes del sistema contando con

.

un SDK compatible para java, C++, Python, IOS, Android, Javascript. Una de las ventajas que ofrece esta plataforma consiste en la opción que ofrece de simular código para configurar cualquier tipo de sensor en las placas Arduino contando con un simulador de circuitos y a partir del mismo generar el código correspondiente a la configuración dada.

El sistema de esta plataforma tiene una estructura que permite asociar tantos sensores como se quieran, mientras cada sensor cuente con un Gateway que sea capaz de enviar datos por protocolo HTTP para poder comunicarse con la nube de servicios. En la ilustración [8] se muestra la estructura del proceso de la plataforma Temboo.

La plataforma es capaz de realizar gráficos históricos en base a los datos tomados y además permite enviar alertas por correo electrónico en base a los mismos.

Ilustración 8 *Proceso de comunicación Temboo (Imagen tomada de https://temboo.com/)*

.

4.1.11. IBM Watson IoT.

Watson internet de las cosas es una plataforma que brinda; Asistente de Watson, es un asistente de inteligencia artificial (AI) de la compañía que ayuda a las empresas a mejorar la lealtad de marca y mejorar las experiencias del cliente transformándolas.

También, soluciones industriales que permite que las necesidades de la industria se logren ver como diferenciadores competitivos. Transformando la industria usando datos, aplicando una herramienta llamada ABB que ofrece desbloquear nuevos servicios públicos para los clientes.

Por último, soluciones empresariales e integradas para resolver temas como la gestión de activos, gestión de las instalaciones y el desarrollo de productos. Permitiéndose alcanzar altos niveles de rendimiento, como por ejemplo el operador francés de ferrocarriles SCNF que utiliza Watson IoT Platform para ofrecer experiencias operativas con mayor seguridad ferroviaria.

4.1.12. AWS IoT.

AWS IoT Core es una plataforma que pertenece a la familia Amazon, AWS IoT permite que un sistema de unificado de dispositivos que conectan el mundo físico con la nube. Presenta diferentes servicios AWS IoT Core, que es la plataforma en la nube que administra los dispositivos y hace que sea de forma segura con las aplicaciones en la nube, AWS loT Device Management que es un servicio el cual facilita la incorporación, organización, monitorización y administración remota segura de cada dispositivo IoT a escala, AWS Greengrass es el software, AWS IoT Analytics es un servicio que permite la ejecución del análisis de los datos, Amazon FreeRTOS es

.

el sistema operativo para los microcontroladores que le facilita la programación, implementación, protección, conexión y administración de los dispositivos, AWS IoT-Click es el servicio que permite que los dispositivos activen las funciones específicas, como bloquear, desbloquear etc. Botón AWS IoT es un botón programable está diseñado para que los configuradores diferentes servicios de Amazon, AWS IoT Device Defender es un servicio complementario que ayuda asegurar el dispositivo IoT.

4.1.13. Google Cloud IoT.

Es una plataforma creada por Google, tiene la opción de prueba gratis, está disponible en diferentes sitios del mundo, posee una API video inteligente de Cloud que permite entender el contenido de los videos.

Los desarrolladores de esta multinacional dicen “Google Cloud Platform tiene una infraestructura diseñada para el futuro, que es, segura, global, de alto rendimiento, rentable, con constantes cambios y que está diseñada para funcionar a largo plazo” [17]. En cuanto a datos y análisis posee las herramientas más eficaces (BigData). No posee servidor, solo tiene código es escalable por esta razón.

.

4.1.14. ThingWorx.

Es una plataforma de innovación industrial líder que está diseñada para entregar rápidamente aplicaciones IoT y experiencias de realidad aumentada (AR) que desbloqueen el valor de los mundos digital y físico convergentes.

Incluye tecnologías y herramientas que permiten a las empresas desarrollar, implementar y extender aplicaciones IoT, ofrece una flexibilidad de implementación con el respaldo de un vibrante ecosistema de socios (PTC).

4.1.15. GE Predix.

Es una plataforma para aplicaciones industriales digitales, que optimiza, conecta y escala las aplicaciones industriales. En la ilustración [9] se puede observar el funcionamiento del proceso de la plataforma IoT industrial.

Una de las características en donde sobre sale más esta plataforma es la seguridad, Predix Platform garantiza su disponibilidad validez e integridad de los datos, utiliza cifrado bipartito y admite informes de cadena de custodia de extremo para código y datos. En cuanto a los entornos de desarrollo Predix Platform ofrece dos entornos de desarrollo, Predix Studio y Full Stack con habilidades de aplicaciones para los desarrolladores.

.

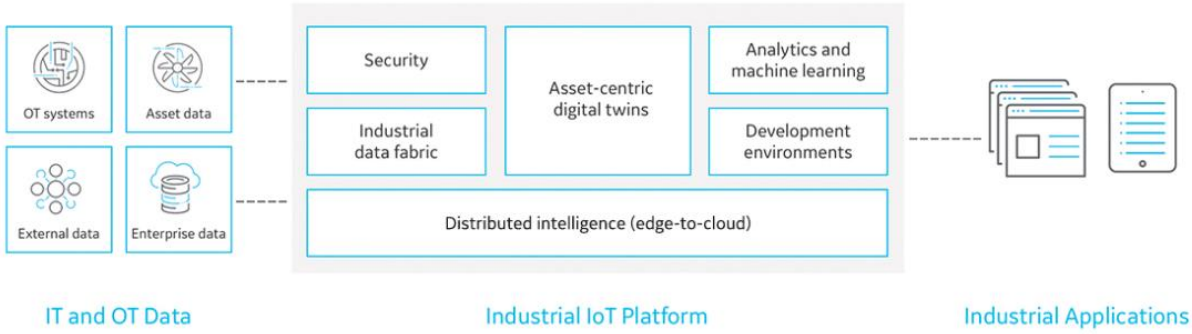


Ilustración 9 *Predix funcionamiento (Imagen tomada de https://www.ge.com/digital/predix-plataform-foundation-digital-industrial-applications)*

5. Marco metodológico

Teniendo en cuenta que en el sistema de Internet de las cosas existe una característica muy importante para la gestión de los datos, ya que permite el intercambio de mensajes entre los dispositivos. Esta característica es el enfoque principal de este proyecto y es conocida como plataforma IoT, pertenece a la arquitectura de los dispositivos IoT, entonces, el mundo de las plataformas para Internet de las cosas es enorme y por esto se debe plantear primero una estrategia para el desarrollo del trabajo.

Es por esta razón que, a partir de las quince plataformas IoT preseleccionadas y categorizadas en tres propiedades, en la figura 2 se puede observar la división por categorías de las quince plataformas. Se da inicio a la investigación y con esto también a la exploración a profundidad de las plataformas más completas, para que las personas pertenecientes a la Institución Universitaria Politécnico Grancolombiano al finalizar este trabajo de grado conozcan la mejor plataforma que se les acomode a sus necesidades y puedan aplicar esta herramienta en sus dispositivos y así lograr una funcionalidad extraordinaria.

.

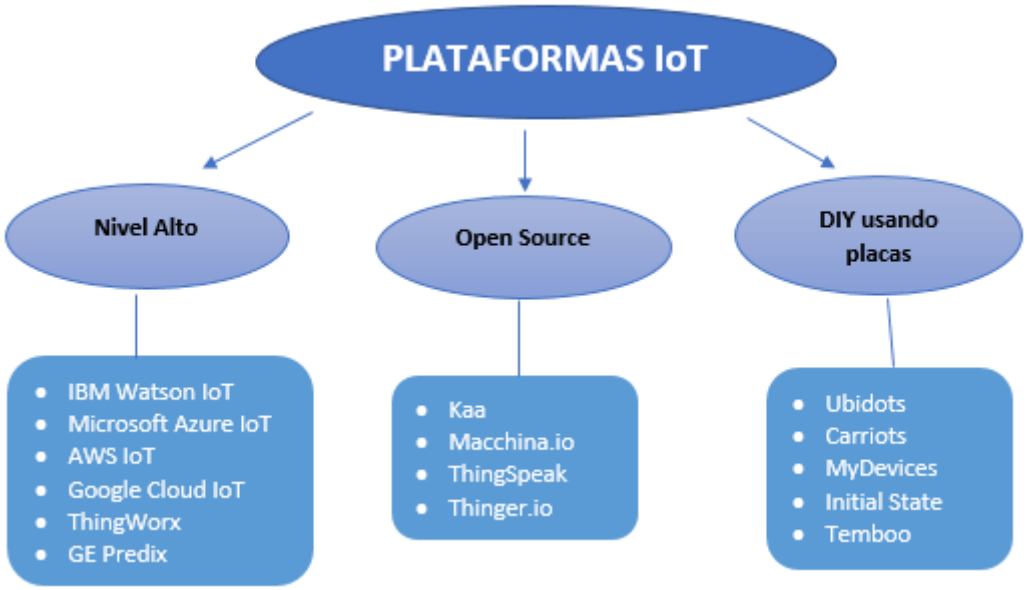


Figura 2 *Mapa conceptual de la categorización de las plataformas (Elaboración propia)*

Se debe basar en un criterio justo la elección de las plataformas, para lograr identificar cuáles se van descartando, por lo tanto, se tiene que desarrollar obteniendo un análisis a partir de las comparaciones entre estas. Presentar; tablas con las propiedades y mapas sobre sus características especiales funciona para lograr ese objetivo de buen juicio.

Cuando se logra obtener una vista más clara sobre las propiedades y características de las quince plataformas por medio de esas herramientas de presentación, se podrá identificar de una manera más eficiente la plataforma IoT para cada necesidad de los dispositivos IoT creados por las personas que deseen buscar ayuda en cuanto al momento de elegir cual de tantas plataformas en el mercado usar.

.

5.1. Niveles de categorización

5.1.1. Plataformas IoT Nivel Alto.

Estas plataformas se les llama nivel alto, porque, se implementan en proyectos magnos de Internet de las cosas, lo que significa que son las más completas y por ende la más costosas del mercado. Como dicen los desarrolladores de Azure "Microsoft Azure le otorga la libertad de crear, administrar e implementar aplicaciones en una tremenda red mundial con sus herramientas y marcos favoritos" [15], lo anterior nos confirma que estas plataformas nivel alto son las mejores cuando se trata de un proyecto obeso.

5.1.2. Plataformas IoT Open Source.

Open Source significa en español Código Abierto y hace referencia a una iniciativa que lleva 20 años. El código abierto nace de la necesidad de promover y proteger códigos fuente, como lo nombran en la página oficial de Open Source [30] "la definición de código abierto fue originalmente derivada de pautas de software libre de Debian (DFSG)", para que pertenezca al grupo de código abierto debe cumplir con ciertos criterios nombrados en la página oficial [30].

Después de leer los criterios que se deben cumplir para que sea un software de código abierto, se comparan con los que las plataformas tienen, entonces como dicen los desarrolladores de la plataforma Kaa "La plataforma Kaa proporciona un conjunto de herramientas abierto y rico en funciones para el desarrollo de productos IoT, por lo que reduce drásticamente los costos

.

asociados, los riesgos y el tiempo de comercialización.” [20], lo anterior significa que estas plataformas por ser de código abierto tienen mejor salida en el mercado por sus bajos costos.

5.1.3. Plataformas IoT DIY.

DIY lo que sus siglas significan (do it yourself) y lo que traduce (hazlo tú mismo). El termino hace referencia a hacer uno mismo el desarrollo del dispositivo seguramente usando placado como material de integración. Los desarrolladores de temboo dicen “temboo permite fácil acceso a API, IoT y tecnología emergente” [26], API es la interfaz de programación de aplicaciones, por tanto, este tipo de plataformas hacen énfasis en “hazlo tú mismo” por el fácil acceso a esta interfaz permitiendo editarla, pensando en lo mejor de acuerdo con la placa que se esté usando.

5.2. Metodología para Prototipo

Por otra parte, se quiere mostrar un caso estudio de un prototipo de dispositivo IoT creado en el semillero, el dispositivo mide temperatura y ubicación geográfica, la adquisición de los datos permite manipularlos (gestión de datos, gráficos con datos, seguimiento del lugar donde se estén obteniendo, entre otros) y la plataforma IoT seleccionada debe cumplir con unos requisitos que nosotros como creadores planteamos. A continuación, se presentan las quince plataformas en la tabla 1 mostrando las propiedades significativas para que con esto logremos concluir cuál de las plataformas se acomoda mejor a la necesidad del dispositivo anteriormente nombrado.

.

PLATAFORMAS IOT	Open Source	Visualización rápida de datos	Escalabilidad	Análisis	Administración de datos	Almacenamiento de bases de datos	Seguridad
IBM Watson IoT		✓	✓	✓	✓	✓	✓
Microsoft Azure IoT Suite	✓	✓	✓	✓	✓	✓	✓
AWS IoT		✓	✓	✓	✓	✓	✓
Google Cloud IoT		✓	✓	✓	✓	✓	✓
ThingWorx (ptc)		✓		✓	✓		✓
GE Predix			✓	✓			✓
Kaa	✓	✓	✓	✓	✓	✓	✓
Macchina.io	✓	✓	✓	✓	✓	✓	✓
Thingspeak	✓	✓		✓	✓	✓	
Ubidots		✓	✓	✓	✓	✓	
Carriots		✓	✓	✓		✓	
MyDevices		✓	✓	✓	✓	✓	✓
Temboo		✓	✓	✓	✓	✓	✓
Initial State	✓	✓	✓	✓	✓		
Thinger.io	✓	✓	✓	✓	✓	✓	

Tabla 1 *Plataformas y propiedades*

Por otra parte, a partir de la exploración de las quince plataformas IoT se pudo saber si las plataformas tenían o no las anteriores propiedades mostradas en la tabla [1], es un buen método no acelerarse y escoger con tan solo esta herramienta (como lo es la tabla), es importante también descartar las que no sirven y escoger por lo menos tres plataformas para un estudio más profundo, que consiste en una instalación de cada plataforma para explorarla.

Los criterios o propiedades anteriormente nombrados en la tabla 1, gozan de capital importancia para identificar las cualidades de cada una de las plataformas, en la siguiente figura 3 se muestra cada criterio con su correspondiente descripción.

.

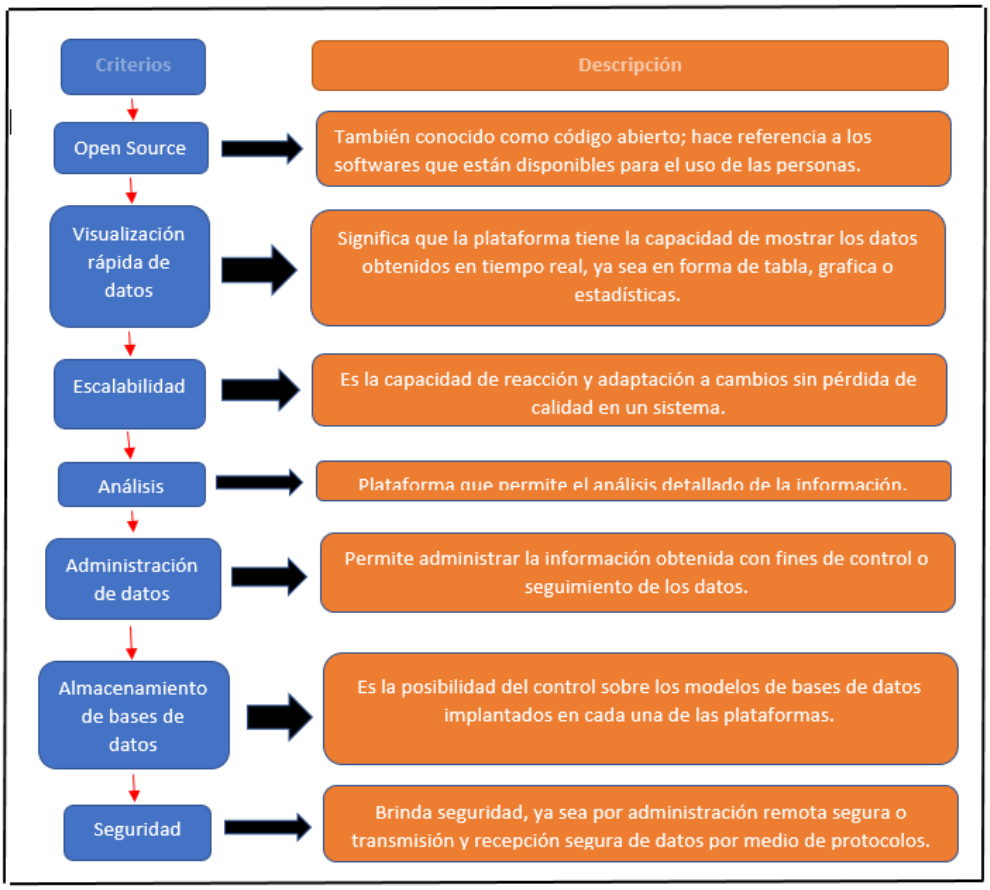


Figura 3 *Descripción de los criterios de elección (Elaboración propia)*

Entonces, teniendo las plataformas elegidas para la exploración, se recomienda hacer una tabla de contenido de estas, como se muestra en la tabla [2], con otro tipo de variables como precios, memoria consumible, etc… para conocer el comportamiento de la plataforma después de la instalación en los equipos.

.

Microsoft Azure	
Virtualización	750 horas Windows/Linux
Almacenamiento	64 GB x2
BD	250 GB/SQL -5 GB/distribuido
RED	15 GB transferencia
APP	Creación de aplicaciones multiplataforma
Solicitudes	Permite 1.000.000 al mes
Usuarios	Permite almacenar 50.000 usuarios al mes
Prueba gratuita	Tiene prueba gratuita, pero es muy básica
Kaa	
Virtualización	Habilitada en BIOS
SO	64 bits
RAM	4 GB - mismo nodo
RAM	256 MB - forma remota
Sistemas Operativos	CentOS 6, Oracle Linux 6 y Ubuntu 16.04
BD	SQL
Costos	Servicio + asistencia
Thinger.io	
RAM	más de 2 GB
BD	MongoDB
SO	64 bits
No. Conexiones	10 servicios Web
Tiempo histórico	Rápido acceso y estadísticas de 30 días
No. Sensores	3 dispositivos
Costos	Cuenta gratuita para Markers

Tabla 2 *Propiedades de instalación de las plataformas*

Después de la exploración ya cada persona tendrá una idea más clara de lo que compone e implica una plataforma IoT en el sistema de Internet de las cosas. La exploración tiene la finalidad de lograr descartar dos plataformas IoT (si se instalaron o exploraron 3) y llegar a una conclusión clara sobre una única plataforma.

.

6. Desarrollo e implementación

En esta sección del documento se pretende mostrar los detalles que se encontraron, en cada una de las exploraciones de las tres plataformas finalistas. Cabe resaltar las propiedades (open source, visualización rápida de datos, escalabilidad, análisis, administración de datos, almacenamiento de bases de datos y seguridad) que se tuvieron en cuenta para la elección de cada una de estas estas plataformas IoT.

PLATAFORMAS IOT	Open Source	Visualización rápida de datos	Escalabilidad	Análisis	Administración de datos	Almacenamiento de bases de datos	Seguridad
Microsoft Azure IoT Suite	✓	✓	✓	✓	✓	✓	✓
Kaa	✓	✓	✓	✓	✓	✓	✓
Thinger.io	✓	✓	✓	✓	✓	✓	

Tabla 3 *Plataformas elegidas y propiedades*

En la tabla [3] se encuentran las tres plataformas elegidas con sus correspondientes cualidades, la elección nace de los requerimientos para el prototipo (medidor climático con GPS), por tanto, esas fueron las elegidas ya que son las más completas en cuanto a lo que se necesita de propiedades y características estudiadas en el marco teórico.

6.1. Exploración Microsoft Azure IoT Suite (IoT Hub)

Como se explica en la documentación oficial “Azure IoT Hub es el componente PaaS central de Azure que utilizan tanto Azure IoT Central como Azure IoT Suite” [15], esto quiere decir que

.

este es un servicio de gestión que se expone en la nube, sirve como un punto central para facilitar el envío de mensajes entre los dispositivos IoT y como un modelo de negocio que esté alojado en la nube.

Este intercambio de mensaje que ofrece IoT Hub no es dirigido; es decir que el intercambio puede iniciar desde cualquiera de los dos puntos, de la nube al dispositivo o del dispositivo a la nube, este nos abre a la posibilidad de manejar o monitorear un dispositivo remotamente.

Para conocer mejor las características que ofrece el IoT Hub de Microsoft se hace un laboratorio que consiste en agregar un dispositivo a esta Plataform. Para utilizar este servicio de Microsoft se debe ingresar al portal de Microsoft Azure y seleccionar la opción de crear un nuevo recurso, seguido de la opción de Internet de las cosas, en la siguiente ilustración [10] se puede ver donde se encuentra lo descrito anteriormente.

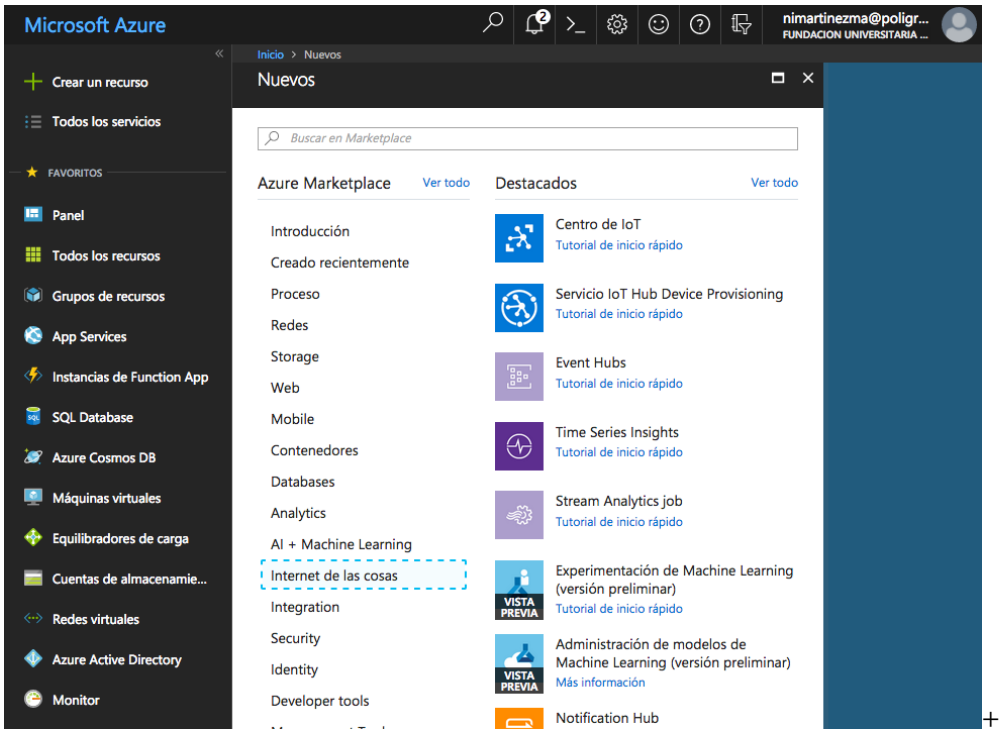


Ilustración 10 *Imagen del portal de Microsoft Azure (Imagen tomada al realizar la exploración de IoT Hub)*

.

Al ingresar a la opción de centro de IoT se mostrará el formulario para configurar nuestro Hub con un nombre y con una localización donde se va a alojar el servicio, también se debe seleccionar la capacidad de mensajes que tendrá el mismo. La capacidad de mensajes que puede tener el servicio va ligada a los paquetes que ofrece la plataforma, a continuación, mostraremos una tabla, en la ilustración [11] tomada de la documentación oficial donde se muestran los beneficios en relación con el costo de cada paquete existente.

Nivel Basic

TIPO DE EDICIÓN	PRECIO POR UNIDAD (AL MES)	NÚMERO TOTAL DE MENSAJES POR DÍA Y POR UNIDAD	TAMAÑO DEL MEDIDOR DE MENSAJES
B1	€8,433	400.000	4 KB
B2	€42,17	6.000.000	4 KB
B3	€421,65	300.000.000	4 KB

Nivel Standard

TIPO DE EDICIÓN	PRECIO POR UNIDAD (AL MES)	NÚMERO TOTAL DE MENSAJES POR DÍA Y POR UNIDAD	TAMAÑO DEL MEDIDOR DE MENSAJES
Gratis	Gratis	8.000	0,5 KB
S1	€21,09	400.000	4 KB
S2	€210,83	6.000.000	4 KB
S3	€2.108,25	300.000.000	4 KB

Ilustración 11 *Tipos de servicios que se ofrecen al crear un IoT Hub (Imagen tomada de la documentación de Microsoft Azure IoT)*

Una vez se configure el Hub se podrán añadir dispositivos desde el panel de control del Hub recientemente creado, para realizar esta acción se debe seleccionar la opción de explorador de dispositivos ubicada en la barra lateral, en la siguiente ilustración [12] se puede observar. Seguido de esto podremos poner un Id que con él se logra identificar nuestro dispositivo y al momento de realizar la creación el sistema nos proporcionará un Id de conexión que necesitaremos al momento de configurar nuestro dispositivo físico; para realizar nuestra conexión física se puede utilizar un dispositivo que sea capaz de comunicarse con Internet mediante un protocolo que soporte la

.

plataforma y a su vez que sea capaz de compilar un lenguaje de programación sobre el cual Microsoft proporciona librerías de conexión.

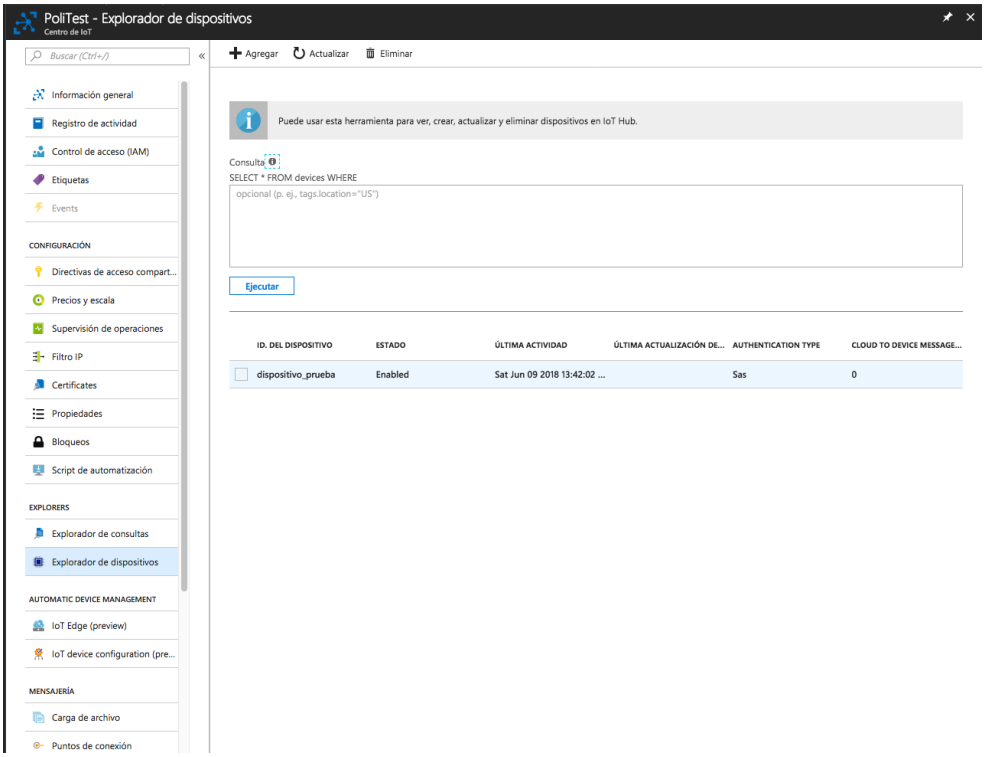


Ilustración 12 *Panel para añadir dispositivos IoT (Imagen tomada al realizar la configuración de un dispositivo en IoT Hub)*

Conectar un dispositivo de IoT a esta plataforma resulta bastante sencillo ya que Microsoft proporciona una alta documentación al respecto, además, este servicio de IoT Hub se puede combinar con otros servicios de Azure que sirven para el manejo y la administración, en general la gestión de los datos recibidos.

.

6.2. Exploración Kaa

Para hacer la exploración de esta plataforma se ingresa al portal [20] y se descarga el Sandbox de Kaa que es una máquina virtual que viene con el software instalado y está listo para utilizar. Una vez importada e inicializada la máquina virtual se mostrará el despliegue de la plataforma y la URL del localhost para ingresar a la misma.

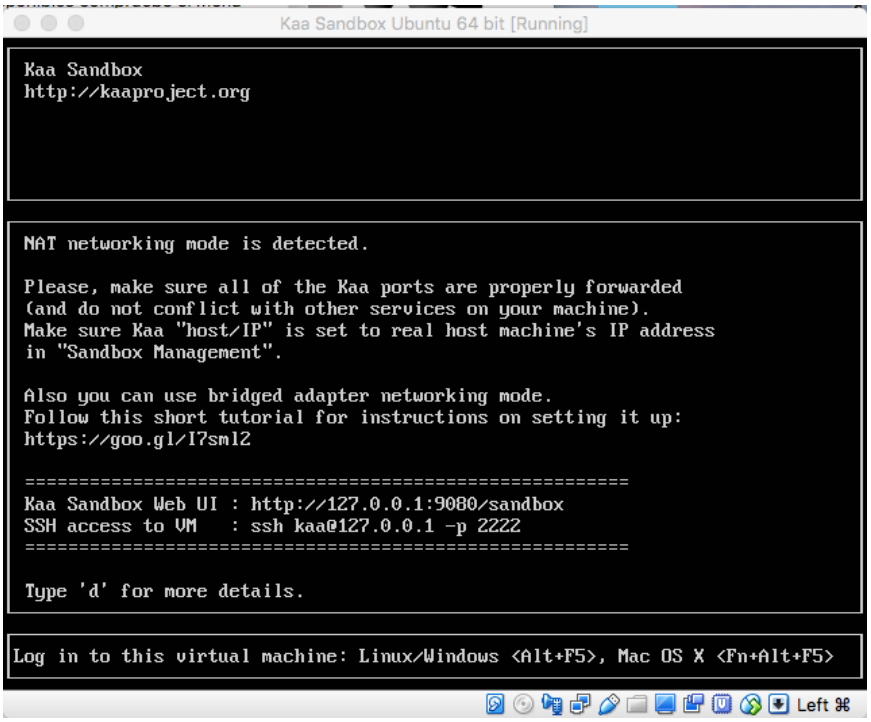


Ilustración 13 *Ventana que se genera al iniciar la imagen que proporciona la plataforma de Kaa*

En la anterior ilustración [13] se muestra la URL para acceder a la plataforma (127.0.0.1:9080/Sandbox), al ingresar a la dirección que se indica se podrá acceder a los servicios de la plataforma para elegir unas de las aplicaciones predeterminadas que trae la herramienta o por otra parte se puede empezar a crear una propia solución. En la siguiente ilustración [14] se muestra el ambiente de la plataforma y las aplicaciones que ofrece Kaa para el desarrollo e implementación de dispositivos en el sistema de Internet de las cosas.

.

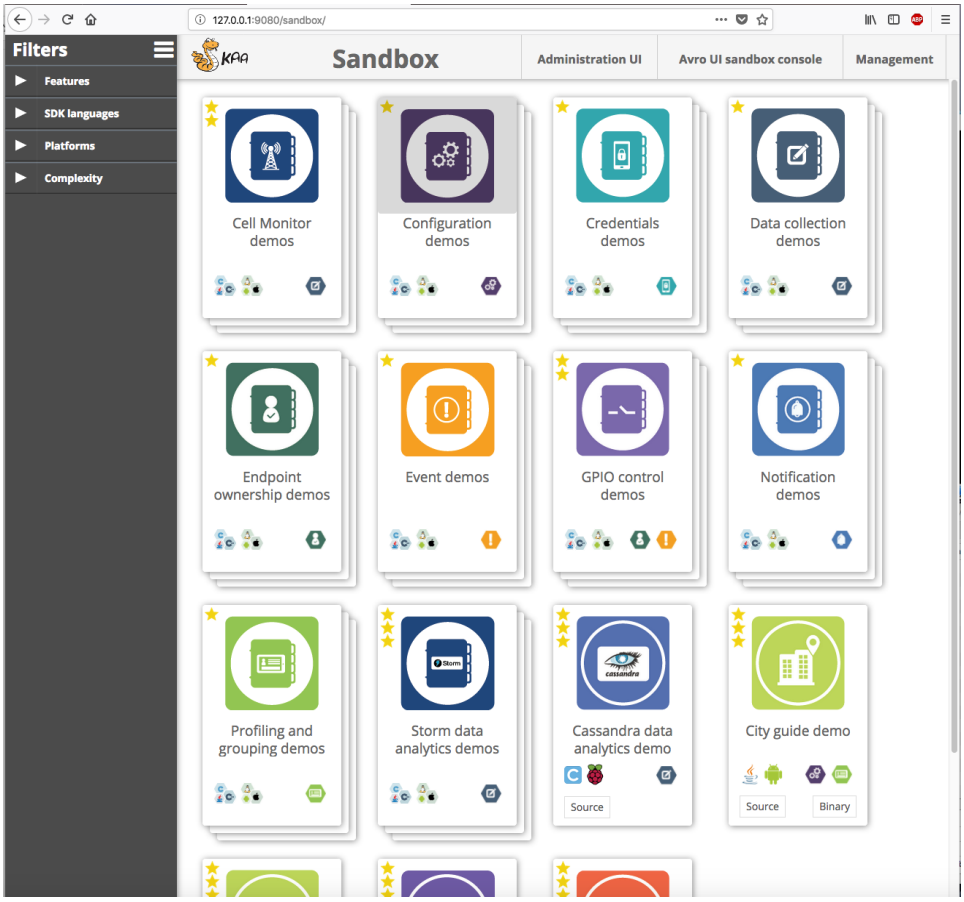


Ilustración 14 *Ventana principal de la plataforma de Kaa (Imagen tomada de la exploración de Kaa)*

La plataforma permite generar SDK para múltiples sistemas operativos con el propósito de conectar los dispositivos que soporten dichos SDK con el fin de monitorear o interactuar con los dispositivos IoT.

Para probar una solución predeterminada se tiene que acceder a la pestaña de administración UI la cual nos solicitará un login, como se muestra en la siguiente ilustración [15], en este caso se ingresa con el usuario *admin* y la contraseña *admin123*. Dentro de este panel podemos seleccionar las aplicaciones que se pueden observar en la ilustración [14] anterior, al seleccionar una podremos acceder al token de aplicación que se necesitará para activar la aplicación una vez se descargue.

.

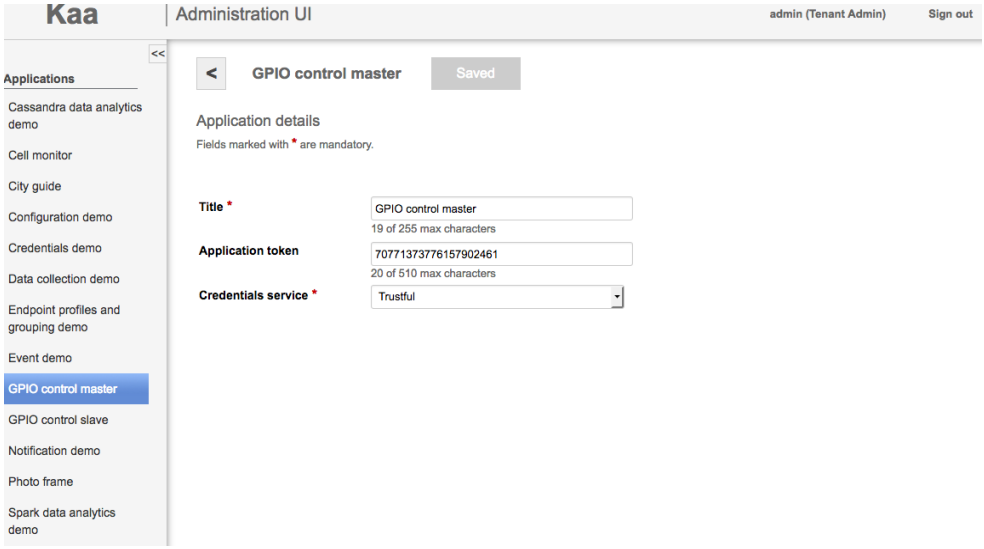


Ilustración 15 *Visualización de la administración UI (Imagen tomada de la exploración de Kaa)*

Una vez hecho esto regresamos al Home y filtramos el sistema operativo Android que es sobre el cual se va a realizar la instalación del SDK, seleccionamos el ejemplo GPIO correspondiente (GPIO control demos) como se puede observar en la siguiente ilustración [16].

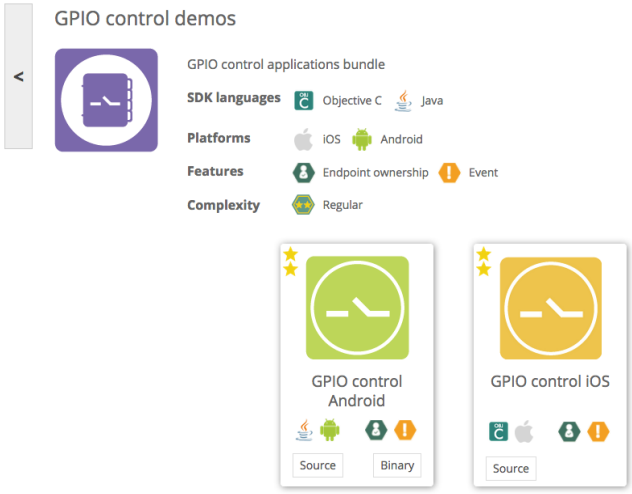


Ilustración 16 *Sección GPIO (Imagen tomada de la exploración de la plataforma Kaa)*

.

Continuando, se selecciona la opción source, al hacer eso el sistema permitirá descargar la aplicación para el sistema operativo que se haya seleccionado. Una vez descargado el archivo se podrá configurar el dispositivo que se va a gestionar desde la aplicación, después de la descarga de este archivo en formato .tar utilizaremos un sistema operativo Linux con el propósito de compilar el SDK con el comando make, al hacer esto se ejecuta una solicitud de credenciales de una red WIFI para que el SDK pueda conectarse al Sandbox de kaa que se instaló previamente.

Después de haber compilado correctamente el SDK se ingresa a la dirección en la que está alojado el Sandbox de kaa (127.0.0.1:9080/Sandbox) desde el dispositivo móvil, se accederá a la misma solución anteriormente mencionada pero esta vez se seleccionará la opción binary, si el paso anterior fue correctamente ejecutado se descarga la aplicación para instalarla en el dispositivo.

Al instalar la aplicación nos aparecerá una pestaña que nos pedirá el token ya mencionado, para que la aplicación nos muestre todas las funcionalidades. De forma paralela tendremos que instalar un SDK en el dispositivo IoT que vayamos a utilizar para que sea posible establecer el intercambio de mensajes entre el dispositivo IoT y la aplicación por medio de kaa. La plataforma ofrece una guía detallada para poder configurar los dispositivos que soportan este SDK.

Una vez se tiene hecha esta configuración, aparecerá desde la aplicación móvil la opción de relacionar el dispositivo IoT con la aplicación, para realizar la comunicación que se desee.

.

6.3. Exploración Thinger.io

Thinger.io tiene librerías de código abierto, que permiten gestionar datos obtenidos tanto del sensor de temperatura, humedad y el módulo GPS, eso teniendo en cuenta que lograr generar una obtención de datos se puede configurando estadísticas en esta plataforma, como las que ya explicaremos más adelante. Por otra parte, en cuanto a la propiedad de visualización rápida de datos se puede configurar de formas como; gráficos, barras estadisticas y hasta mapas (ubicación geográfica).

En su consola de tableros se pueden encontrar, como se muestra en la ilustración [17] en primer lugar Statistics, allí hay cuatro variables importantes; los Devices, es donde se crean los dispositivos como sensores o placas conectados a la plataforma, Dashboard es la interfaz gráfica donde se muestra la información adquirida de los dispositivos creados en la cuenta en tiempo real, Data Buckets es donde salen los datos del dispositivo en pocas palabras un depósito de datos adquiridos por los dispositivos y por ultimo Endpoints que como dice en la documentación de thinger.io “Es el punto de entrada a un servicio, un proceso o cualquier otro destino” [31] lo que significa que esta plataforma IoT puede gestionar las llamadas que los dispositivos de medición solicitan directamente. Estas variables para que funcionen deben ser creadas y configuradas como primer paso.

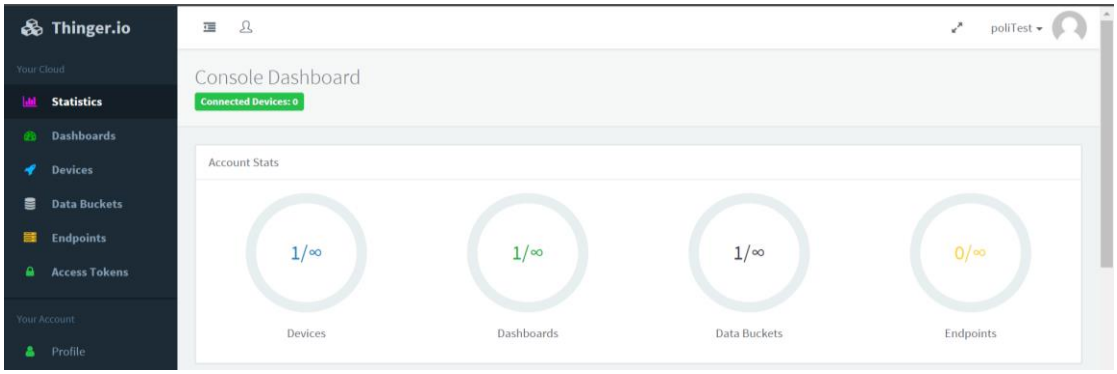


Ilustración 17 *Visualización interactiva de datos (Imagen obtenida de la exploración de Thinger.io)*

.

Cuando es necesario aplicar una medida de control, ya sea porque, los niveles sobre pasan un punto crítico. Thinge.io no permite generar alertas por medio del tablero de control sino, es necesario modificar el código colocando un Endpoint para que envié correos, sí una medición sobre pasa algún valor.

7. Resultados

En esta sección se presentan los resultados de la implementación de la plataforma elegida en un caso estudio predeterminado, esto para evidenciar que el método de búsqueda de la mejor plataforma para un caso específico funciona, la investigación a profundidad ayuda y que fácil es desarrollarlo. A continuación, en la figura 4 se muestra la razón de la elección de la plataforma final, teniendo en cuenta las necesidades del caso estudio.

Figura 4 *Puntos a favor por elección (Elaboración propia)*

.

7.1. Descripción caso estudio

Se tiene un prototipo de dispositivo IoT diseñado en un semillero de investigación de la universidad, este dispositivo posee tres variables de medición las cuales son; sensor de humedad, sensor de temperatura y modulo GPS. Todos estos dispositivos de obtención de datos se ensamblaron en una tarjeta Sparkfun, como se muestra en la ilustración [18]. En el ambiente de desarrollo que brinda el software de Arduino se realizó la correspondiente programación para la placa de desarrollo (el código lo podrá encontrar al final del documento en los anexos). La finalidad de este dispositivo es llevar un seguimiento por tanto un control de los datos de las tres variables de medición en cualquier espacio en el que se encuentre naturalmente todo el sistema de IoT.

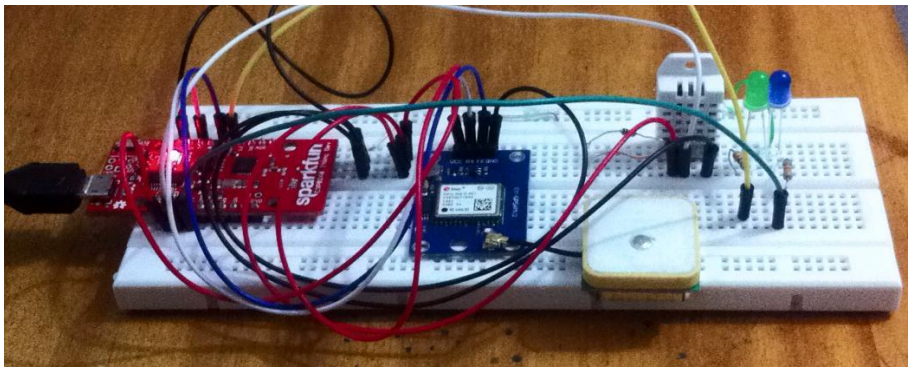


Ilustración 18 *Ensamble real físico del dispositivo IoT*

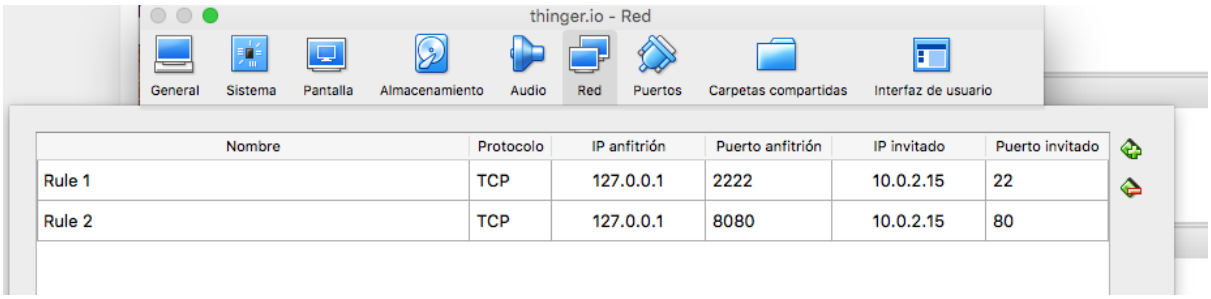
7.2. Resultados Thinger.io

.

Para poder realizar el prototipo sobre la plataforma thinger.io que fue la escogida, descargamos una imagen iso de Ubuntu para crear la máquina virtual donde se monta el servidor de thinger.io. Como esta plataforma utiliza Mongo DB como gestor de base de datos lo primero que se debe realizar es la descarga y posterior instalación en nuestro host de este servidor. Para realizar una correcta instalación de esta plataforma se puede guiar de la documentación [1].

Lo primero que se tiene que tener listo es el ambiente, los diseñadores recomiendan un ambiente con una arquitectura de 64 bits ya que mongo DB tiene una limitante 2 GB de almacenamientos en un sistema de 32 bits e indican que si se hace en un sistema operativo Ubuntu (sistema que se utiliza en este caso) se debe emplear una versión igual o superior a la 16.0.4. [1]

Para facilitar la instalación de la plataforma se crean unos convenios en la red NAT de la máquina virtual que redirecciona puertos de la máquina real con puertos de la máquina virtual, con el propósito de establecer una conexión por SSH desde la terminal de la máquina física para agilizar el paso de los comandos de instalación, ya que de esta forma se podrá utilizar la función de copiado y pegado entre las dos máquinas, función que no permite VirtualBox al utilizar la terminal desde la máquina virtual, el otro convenio se utilizara con el fin de acceder desde la máquina física a la plataforma que se expondrá desde la máquina virtual desde el puerto 80. En la ilustración [19] se muestran la pestaña donde se agregan los convenios a la cual se accede desde el panel de configuración de la máquina virtual al ingresar en las configuraciones de red y seleccionar la opción de reenvío de puertos desde el adaptador NAT, en la documentación se podrán encontrar más detalles [2].



.

Ilustración 19 *Reglas de red para el servidor de la plataforma Thinger.io (Imagen tomada de la exploración de Thinger.io)*

Una vez se tenga el ambiente de desarrollo listo podemos verificar que los servicios que necesitamos hayan quedado funcionando correctamente con los siguientes comandos:

- *sudo systemctl status mongodb*
- *sudo service snap.thinger-maker-server.thingerd status*

En la ruta /var/snap/thinger-maker-server/common/ se encuentran los archivos de configuración por defecto; acá se puede configurar la forma de acceder a la plataforma (host - puertos) los sistemas de autenticación y la distribución de servicios dentro de la plataforma (como restricciones para un usuario).

Para empezar a utilizar las funciones de la plataforma se debe crear un usuario para poder acceder al panel principal como se muestra en la ilustración [19], la diferencia que más se puede distinguir inicialmente respecto a la plataforma alojada en el servidor inicial es que no existe restricción en la cuenta para añadir los recursos que ofrece la misma.

.

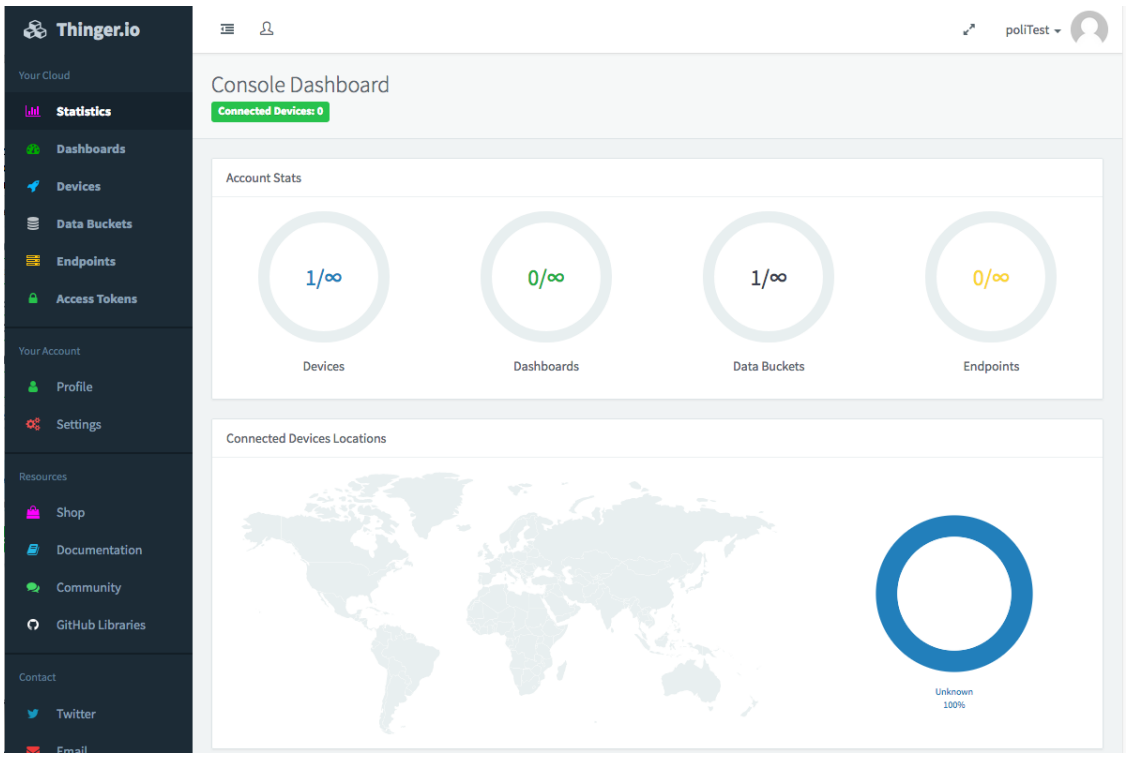


Ilustración 20 *Panel principal de thinger.io instalado en el servidor local (Imagen tomada de la exploración de Thinger.io)*

Para empezar a utilizar las funciones de la plataforma se procede a crear un dispositivo, para esto se debe seleccionar la opción Devices que como se puede ver en la ilustración [20] está ubicada en la barra lateral del panel principal, al seleccionarla se podrán crear nuevos dispositivos y gestionar los que ya existen, como en este momento no hay dispositivos creados, se selecciona la opción para añadir uno, entonces se debe llenar el formulario que aparece como se puede ver en la ilustración [21].

.

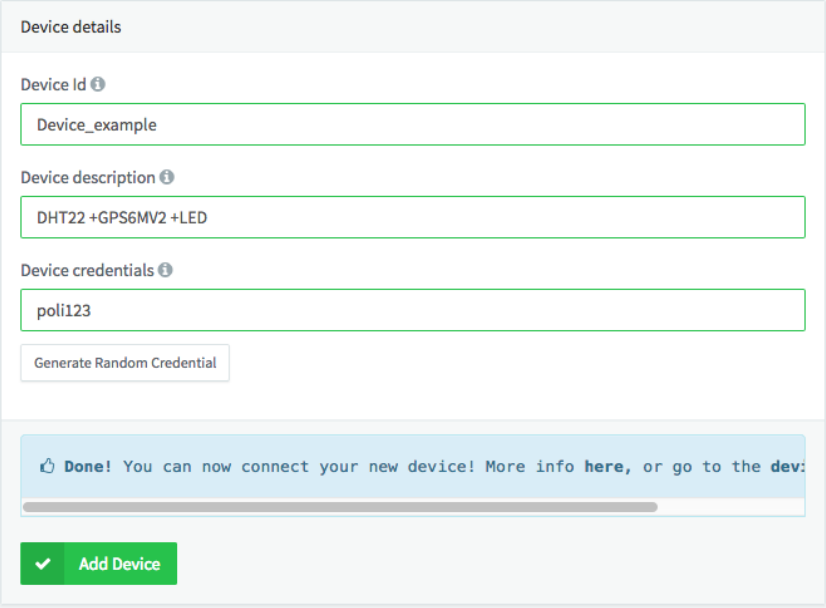


Ilustración 21 *Datos que se solicitan al crear un dispositivo (Imagen tomada de la exploración de Thinger.io)*

Una vez hecho esto se tiene el dispositivo creado en la plataforma, ahora estas credenciales se tienen que asociar a un montaje físico, la plataforma no tiene restricciones con el número de datos que se reciban por un dispositivo, es decir que por un montaje físico se pueden enviar múltiples datos mientras la placa que se esté utilizando lo permita; para este ejemplo se utilizará una placa Sparkfun (ESP8266 Thing Dev) que tiene integrado un módulo WIFI, a la placa se le conecta un sensor DHT22 para monitorear la temperatura y la humedad, un módulo GPS, y se configuran dos leds para probar el control remoto que facilita la plataforma.

En este caso se va a utilizar el software Arduino para configurar los elementos anteriormente descritos. Lo primero que se hace para lograr la conexión del montaje y la plataforma es la importación de las librerías que proporciona Thinger.io para realizar dicha conexión. A continuación, se muestran las líneas de código que se deben utilizar para lograr esto [1].

```
//Librerías que se utilizan
#define THINGER_SERVER " ****** " // Servidor donde se aloja la
plataforma.
#include <ESP8266WiFi.h> //Librería de conexión WiFi del módulo ESP8266
#include <ThingerESP8266.h> //Librería de la plataforma thinger.io
```

.

```
//Acá deben ir las credenciales que se crearon en la plataforma
// Parámetros de la conexión con thinger.io
#define usuario "poliTest"
#define device_Id "Device_example"
#define device_credentials "poli123"

//Se instancia el objeto que se encarga de llevar a cabo la conexión;
ThingerESP8266 thing (usuario, device_Id,device_credentials);
//Se agregan unas credenciales de acceso a una red para poder completar la
conexión
thing.add_wifi("SSID", "PASSWORD");
```

Ahora que el programa es capaz de realizar la conexión con la plataforma se deben configurar los formatos en los que se enviaran la información, para esto se utilizara el objeto “thing” que se instancia utilizando la siguiente estructura:

```
thing["Data"] >> [](pson& out){
        out["Value1"] = value1;
        out["Value1"] = value2;
 };
```

Este formato es el que se va a utilizar para todo el traspaso de información, el valor “Data” va a ser el nombre con el que llegue toda información encapsulada y el símbolo “>>” quiere decir que la información va a ser de salida si se quisiera dejar un recurso para acceso remoto se deberá poner “<<”, dentro del método se agregan los Labels para identificar y poder acceder los datos que se envían a la plataforma.

Una vez se tengan estos datos definidos en el método “loop” Arduino se utiliza de nuevo la instancia “thing” para constantemente estar enviando datos a la plataforma.

```
thing.handle();
```

A continuación, se muestra el formato de datos que se utilizó para llevar a cabo el montaje que se explicó anteriormente:

.

```
//Esta estructura va a almacenar la información de los sensores de
temperatura y gps
 thing["Data"] >> [](pson& out){
        out["Temperatura"] = dht.readTemperature();
        out["Humedad"] = dht.readHumidity();
        out["Latitud"] = latitud;
        out["Longitud"] = longitud;
 };
//Esta estructura indica que desde la plataforma va a ser posible enviar
señales al pin que está asociado al led

        thing["LED_REMOTO"] << [](pson& in){
        if(in.is_empty()){
        in = (bool) digitalRead(LET_OUT);
        }else{
        digitalWrite(LET_OUT, in ? HIGH: LOW);
        }
 };

//Esta estructura indica que desde la plataforma va a ser posible enviar
señales para regular la intensidad del led
   thing["LED_REG_REMOTO"] << [](pson& in){
        analogWrite(LET_REG_OUT, in["Value"]);
   };
}
```

Al tener este formato listo igual que la configuración de conexión se sube el programa a la placa y al volver al panel principal de thinger.io, se selecciona de nuevo la opción “Devices” se encuentra con que el dispositivo que se había creado anteriormente está correctamente conectado como se muestra en la ilustración [22].

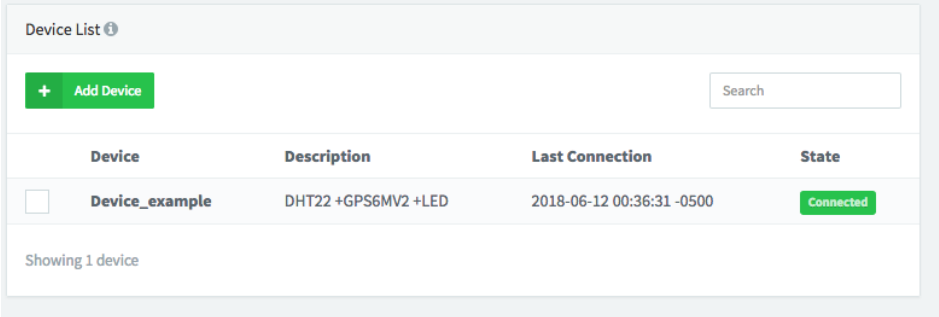


Ilustración 22 *Muestra del dispositivo físico está conectado correctamente con la plataforma (Imagen tomada de la exploración de Thinger.io)*

.

Si ingresamos al dispositivo que hemos creado podremos observar como se muestra en la ilustración [23] un pequeño tablero de control donde podremos ver la dirección Ip que tiene el dispositivo el tiempo que lleva conectado el mismo y una tabla que se actualiza en tiempo real que indica la transferencia de datos que se realiza de la plataforma al dispositivo y/o del dispositivo a la plataforma, también se encuentra la opción de generar un token para el dispositivo con el fin de permitir que otras aplicaciones puedan tener acceso a él y por último encontramos la opción para interactuar con los recursos del montaje que se configuró desde Arduino al seleccionar la opción "View API"; como se muestra en la ilustración [24] en este panel llamado ejemplo del api, se van a encontrar tres bloques con los mismos nombres que se le asignaron a las estructuras de datos que se configuraron para comunicarse con la plataforma.

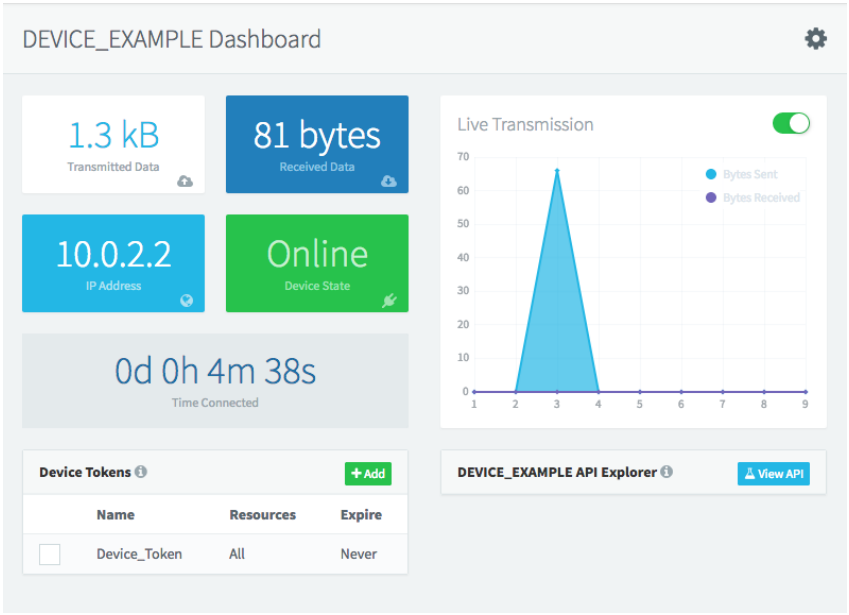


Ilustración 23 *Panel de monitoreo del dispositivo ejemplo (Imagen tomada de la exploración de Thinger.io)*

.

Ilustración 24 *Panel que permite interactuar con el API de conexión del dispositivo de ejemplo (Imagen tomada de la exploración de Thinger.io)*

Cabe recordar que de los recursos que se pueden ver en la ilustración [24] dos de ellos fueron configurados para recibir datos desde la plataforma y el otro para enviar a la plataforma datos sobre su eterno; en la ilustración [25] podemos ver que al desplegar la opción "Data" se van a encontrar los datos que el montaje físico está enviando y cada vez que seleccione la opción "run" se mostrarán los últimos datos recopilados, esto también ayuda a confirmar que el sistema está recibiendo de forma correcta.

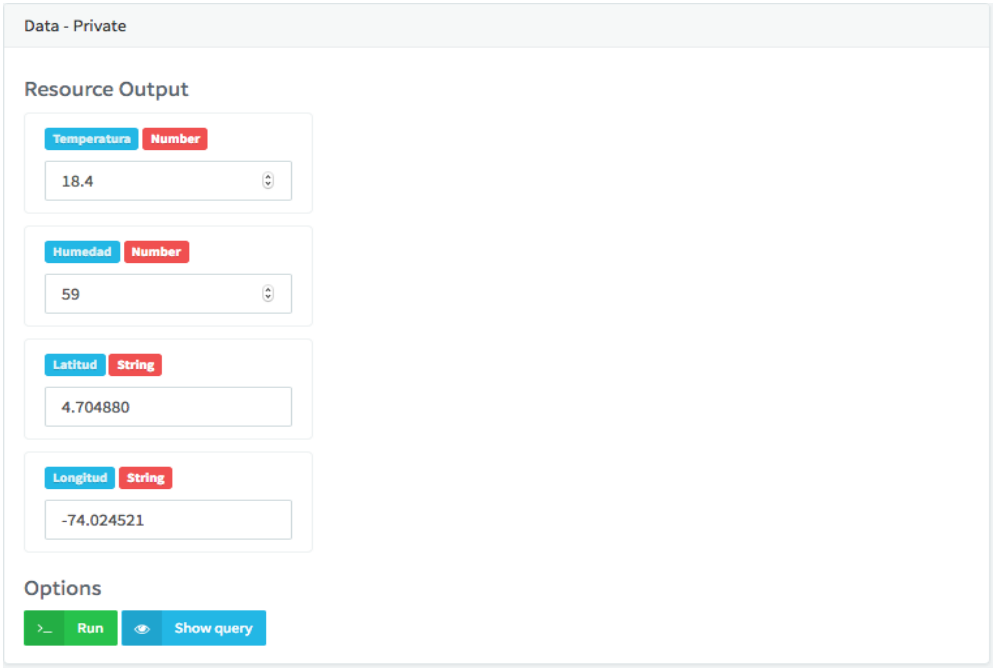


Ilustración 25 *Datos enviados por el montaje (Imagen tomada de la exploración de Thinger.io)*

En este panel que expone el API del dispositivo de prueba, también se encuentran dos instrucciones que en vez de recibir datos están a la espera de que se les asigne un valor para enviar información al montaje físico, en la ilustración [26] podemos observar el campo LED_REMOTO que contiene un elemento visual y este hace alusión a un valor booleano, al oprimir este elemento se enviará una señal al montaje físico que apagará o encenderá el led que se tiene asociado al pin

.

que recibe la señal. También encontraremos el campo LED_REG_REMOTO donde se observará un campo numérico en el que podremos asignar un valor que se enviará a la placa para regular la intensidad lumínica del led al que se le tenga asignado ese pin.

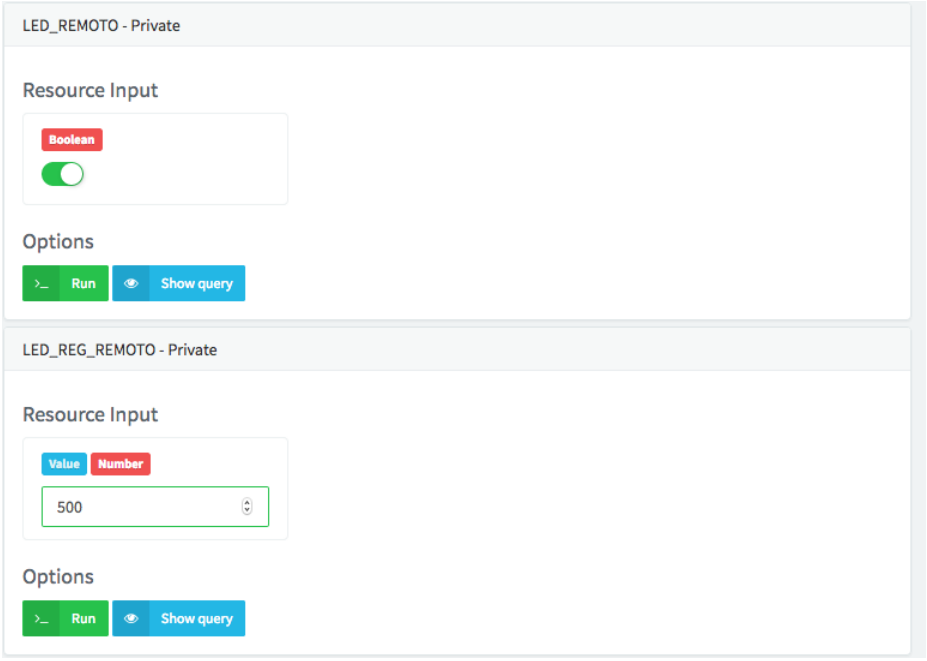
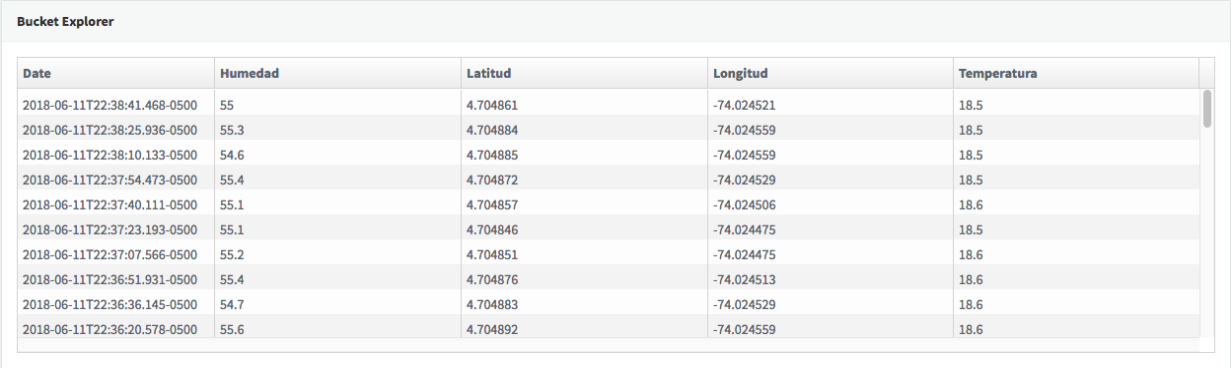


Ilustración 26 *Datos con los que se prueba la comunicación entre la plataforma y el dispositivo físico (Imagen tomada de la exploración de Thinger.io)*

Ahora si lo que se quiere es almacenar la información histórica de los datos que se reciben, por parte del dispositivo se debe hacer uso de otra funcionalidad de la plataforma que se le conoce como Data Buckets, funcionalidad me permite guardar todos los datos que se reciban por parte de un recurso de un dispositivo, para crear uno de estos recursos se debe asignar un nombre para el mismo y asociar un recurso de un dispositivo, un ejemplo de esto se puede ver en la ilustración [27] la cual pertenece al Data Bucket que se creó para almacenar la información del recurso Data creado para el dispositivo (prototipo) de ejemplo.

.



Bucket Explorer

Date	Humedad	Latitud	Longitud	Temperatura
2018-06-11T22:38:41.468-0500	55	4.704861	-74.024521	18.5
2018-06-11T22:38:25.936-0500	55.3	4.704884	-74.024559	18.5
2018-06-11T22:38:10.133-0500	54.6	4.704885	-74.024559	18.5
2018-06-11T22:37:54.473-0500	55.4	4.704872	-74.024529	18.5
2018-06-11T22:37:40.111-0500	55.1	4.704857	-74.024506	18.6
2018-06-11T22:37:23.193-0500	55.1	4.704846	-74.024475	18.5
2018-06-11T22:37:07.566-0500	55.2	4.704851	-74.024475	18.6
2018-06-11T22:36:51.931-0500	55.4	4.704876	-74.024513	18.6
2018-06-11T22:36:36.145-0500	54.7	4.704883	-74.024529	18.6
2018-06-11T22:36:20.578-0500	55.6	4.704892	-74.024559	18.6

Ilustración 27 *Data Bucket que almacena la información del dispositivo de ejemplo (Imagen tomada de la exploración de Thinger.io)*

Esta información recopilada por una data Bucket puede ser exportada para que se utilice por un tercero o también se puede utilizar para otra utilidad de thinger.io que son los Dashboard, se pueden crear paneles de controles en los que se muestren de una forma más amigable o cómoda para el usuario los resultados de los datos tomados por los dispositivos como se puede ver en la ilustración [28].

.

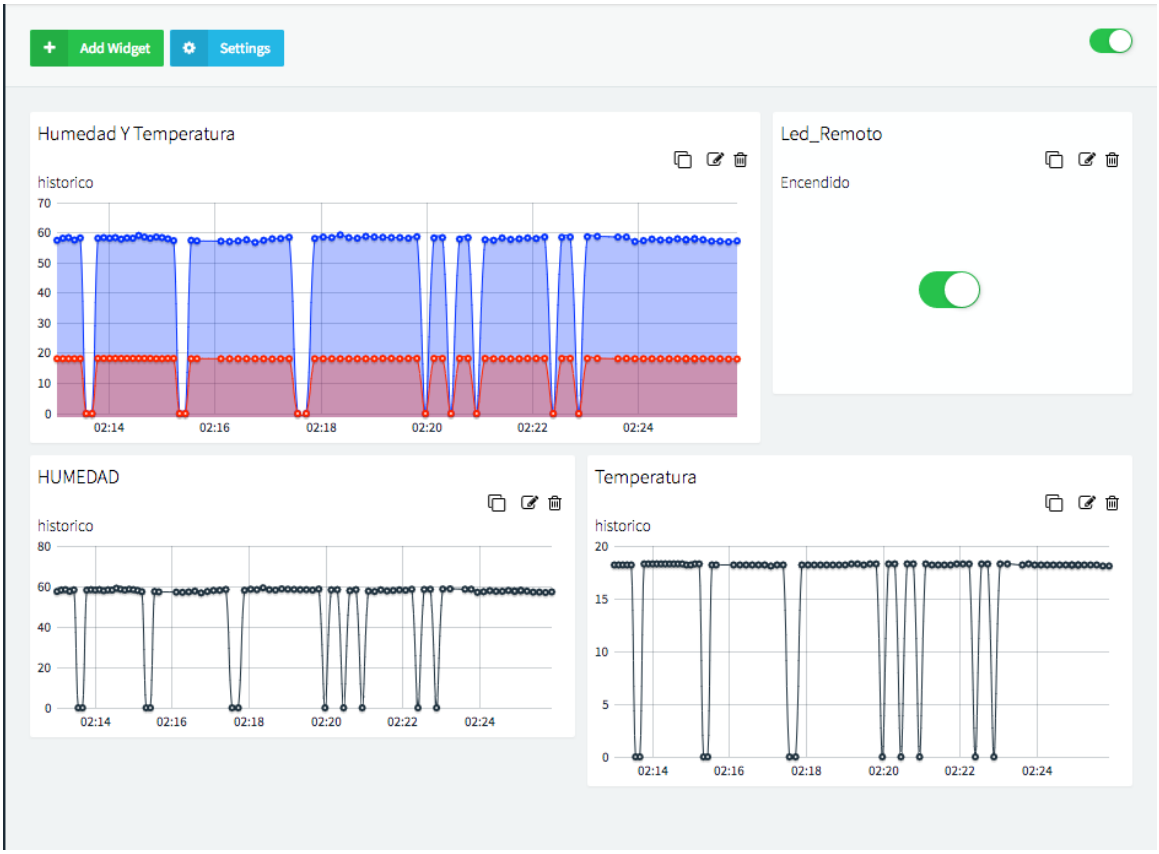


Ilustración 28 *Visualización de los recursos del dispositivo de prueba (Imagen tomada de la exploración de Thinger.io)*

8. Conclusión

En búsqueda de una plataforma que se adaptará al dispositivo de prueba, realizamos la investigación sobre plataformas que se encuentran en el mercado, al revisar las plataformas más grandes encontramos que; tienen muchas funcionalidades, a pesar de eso no cumplía nuestras necesidades ya que el costo que estas tienen es elevado e incluían más funciones de las que se tienen pensado utilizar (según las necesidades del prototipo) por lo tanto no se justifica el gasto.

.

Pasando por las plataformas tipo Open Source encontramos que las que más se adaptaba a nuestra necesidad son; Kaa Project y la plataforma Thinger.io ya que permiten instalar la plataforma en un servidor propio y configurarlo de la forma que se desee para poder implementar un sistema IoT completo.

La plataforma Kaa proporciona una serie de SDKs para poder crear Endpoints que se conecten a esta misma, a pesar de que estos SDKs están diseñados para múltiples plataformas, toda la configuración necesaria para que los mismos funcionen está definida de tal forma que se deben realizar sobre un sistema Linux, lo que limita el uso de estos. Además, no proporciona una interfaz gráfica que facilite el administrar o monitorear los datos que se reciban de los sensores configurados.

Por lo tanto, se concluye que la plataforma Thinger.io es la más apropiada, ya que las librerías que proporciona están disponibles para múltiples plataformas y se puede trabajar con ellas desde un IDE como Arduino, lo cual permite aumentar el rango de trabajo que se le pueden dar a las aplicaciones. Agregando que facilita un entorno gráfico amigable e intuitivo; que permite monitorear y controlar los dispositivos. Esta plataforma es Open-Source, como ya se nombró, por lo que permite tener un general control sobre la personalización del sistema de alojamiento de back-end. Igualmente cuenta con un API de servicios REST para llevar acabo la comunicación entre los dispositivos y la plataforma, para poder distinguir la información utiliza una estructura Json en la que se envían los mensajes. Por último, cabe resaltar que soporta cualquier tipo de placa que se pueda configurar desde Arduino y también expone librerías para las plataformas de que se trabajen en Linux, lo que significa que brinda una gran variedad de clientes que se pueden unir a configurar con la plataforma.

.

9. Referencias bibliográficas

[1] Thinger.io "*Thinger.io - Server Deployment*" (s.f) [En línea]. Disponible en: http://docs.thinger.io/deployment/. [Accedido: 23-may-2018]

[2] Jorgepuente "*Acceder a una máquina virtual en VirtualBox a través de NAT*" (2017) [En línea]. Disponible en: https://jorgepuente.es/sistemas/acceder-una-maquina-virtual-virtualbox-traves-nat/. [Accedido: 21-may-2018]

[3] Thinger.io "*Thinger.io – Server API*" (s.f) *[En línea] Disponible en: http://docs.thinger.io/api/. [Accedido: 23-may-2018]*

[4] L. Mainetti, L. Manco, L. Patrono, and R.Vergallo, *"A Cloud Architecture for Managing IoT-aware Applications According to Knowledge Processing Rules".* Journal of Communications Software & Systems. Mar2016, Vol. 12 Issue 1, p45-52. 8p.

[5] T.Carpenter, J.Hatcliff, E.Vasserman, *"A Reference Separation Architecture for Mixed-Criticality Medical and IoT Devices"*. SafeThings'17 Proceedings of the 1st ACM Workshop on the Internet of Safe Things Pages 14-19. Delft, Netherlands – November 05 -05 ,2017.

[6] A.Taherkordi, F.Eliassen and G.Horn, *"From IoT big data to IoT big services".* Proceedings of the Symposium on Applied Computing Pages 485-491. Marrakech, Morocco – April 03 – 07, 2017.

[7] P.Rosenkranz, M.Wahlisch, E.Baccelli and L.Ortmann *"A Distributed Test System Architecture for Open-source IoT Software".* IoT-Sys'15 Proceedings of the 2015 Workshop on IoT challenges in Mobile and Industrial Systems Pages 43-48. Florence, Italy – May 18 -18, 2015.

[8] E.Cavalcante, M.Alves, T.Batista, F.Delicato, P.Pires *"An Analysis of Reference Architectures for the Internet of Things".* CobRA'15 Proceedings of the 1st International Workshop on Exploring Component-based Techniques for Constructing Reference Architectures Pages 13-16. Montréal, QC, Canada –May 06 -06, 2015.

[9] Secmotic Alvaro Cárdenas "*¿Qué es una plataforma?"* (2016) [En línea] Disponible en: https://secmotic.com/blog/plataforma-iot/ [Accedido: 21-may-2018]

[10] FOSTEC&COMPANY "*Internet of Things (IoT)*" (2018) [En línea] Disponible en: https://www.fostec.com/es/competencias/estrategia-de-digitalizacion/internet-of-things-iot/ [Accedido: 21 may-2018]

[11] Fernando Ramírez Navia "*Datos estructurados vs datos no estructurados: Gestión de información en BigData*" (2017) [En línea] Disponible en: https://fireosoft.com.co/blogs/datos-estructurados-vs-no-estructurados/ [Accedido: 25-may-2018]

.

[12] Microsoft Azure "*Visualize real-time sensor data from Azure IoT Hub using Power BI*" (2018) [En línea] Disponible en: https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-power-bi [Accedido: 18-may-2018]

[13] E.Rico "*Ingeniería y diseño de Internet of Things*" (2018) [En línea] Disponible en: http://www.ermesh.com/plataformas-iot-internet-de-las-cosas/ [Accedido:2-abril-2018]

[14] Watson Internet of Things "*IBM Platform*" (2018) [En línea] Disponible en: https://www.ibm.com/internet-of-things?lnk=hmmpr_iot&lnk2=learn [Accedido: 2-abril-2018]

[15] Microsoft "*Microsoft Azure*" (2018) [En línea] Disponible en: https://azure.microsoft.com/es-es/ [Accedido: 3-abril-2018]

[16] An amazon company "*AWS IoT*" (2017) [En línea] Disponible en: https://aws.amazon.com/es/iot/ [Accedido: 3-abril-2018]

[17] Google "*Google Cloud*" (2018) [En línea] Disponible en: https://cloud.google.com/ [Accedido: 4-abril-2018]

[18] Ptc "*ThingWorx Delivers Industrial Innovation*" (2018) [En línea] Disponible en: https://www.ptc.com/en/products/iot [Accedido: 4-abril-2018]

[19] GE Imagination at work "*PREDIX*" (2018) [En línea] Disponible en: https://www.ge.com/digital/predix-platform-foundation-digital-industrial-applications [Accedido: 5-abril-2018]

[20] Supported by KaaIoT "*Kaa*" (2018) [En línea] Disponible en: https://www.kaaproject.org/ [Accedido: 5-abril-2018]

[21] Applied informatics "*Macchina.io*" (2014-2018) [En línea] Disponible en: https://macchina.io/ [Accedido: 6-abril-2018]

[22] The MathWorks, Inc "*ThingSpeak*" (2018) [En línea] Disponible en: https://thingspeak.com/ [Accedido: 6-abril-2018]

[23] Copyright Ubidots "*Ubidots*" (2018) [En línea] Disponible en: https://ubidots.com/ [Accedido: 7-abril-2018]

[24] Copyright Carriots "*Carriots by altair*" (2018) [En línea] Disponible en: https://www.carriots.com/ [Accedido: 7-abril-2018]

[25] My Devices "*Simplify the Connected World*" (2018) [En línea] Disponible en: https://mydevices.com/ [Accedido: 8-abril-2018]

[26] Temboo "*Tools for Digital Transformation*" (2018) [En línea] Disponible en: https://temboo.com/ [Accedido: 8-abril-2018]

[27] Initial State Technologies, Inc "*Initial State*" (2018) [En línea] Disponible en: https://www.initialstate.com/ [Accedido: 9-abril-2018]

.

[28] S.A. Wolters Kluwer España., "*Proyecto Technos: Internet of Things y su impacto en los recursos humanos y en el marco regulatorio de las relaciones laborales*", Salvador del rey Guanter, (primera edición: octubre 2017).

[29] IEBS Bussiness School "*Internet de las cosas y Smart Hotel*" (2017) [En línea] Disponible en: https://www.itop.es/blog/item/que-es-iot-internet-of-things.html [Accedido: 23-feb-2018]

[30] Open Source Initiative "*The Open Source Definition*" (2017) [En línea] Disponible en: https://opensource.org/osd [Accedido: 13-marzo-2018]

[31] Thinger.io "thinger.io platform" (2018) [En línea] Disponible en: https://thinger.io/ [Accedido: 31-marzo-2018]

Anexos

```
#define THINGER_SERVER "192.168.0.3"      // Servidor Interno
//#define THINGER_SERVER "186.28.198.164" // Servidor Externo

#include <ESP8266WiFi.h>       //Librería de conexión WiFi del módulo
ESP8266
#include <ThingerESP8266.h>    //Librería de la plataforma thinger.io
#include "DHT.h"               //Librería de los sensores DHT11, DHT22,
etc.

//librerias gps
#include <SoftwareSerial.h>
#include <TinyGPS.h>
TinyGPS gps;
SoftwareSerial ss(4,5);
char dato=' ';
String latitud="";
String longitud="";
String floatToString( float n, int=8, int=6, boolean=true);

// Parámetros del DHT
#define DHTPIN 2      //Pin de conexión - GPIO02
#define DHTTYPE DHT22   //Modelo
DHT dht(DHTPIN, DHTTYPE,15);

//Parámetros del Led para manejo Remoto
  int LET_OUT = 14;      //Pin de conexión
  int LET_REG_OUT = 13;      //Pin de conexión

// Parámetros del conexión con thinger.io
#define usuario "poliTest"
```

.

```
#define device_Id "Device_example"
#define device_credentials "poli123"

ThingerESP8266 thing (usuario, device_Id,device_credentials);


// Convierte un float en una cadena.
// n -> número a convertir.
// l -> longitud total de la cadena, por defecto 8.
// d -> decimales, por defecto 2.
// z -> si se desea rellenar con ceros a la izquierda, por defecto true.
String floatToString( float n, int l, int d, boolean z){
 char c[l+1];
 String s;

 dtostrf(n,l,d,c);
 s=String(c);

 if(z){
 s.replace(" ","0");
 }

 return s;
}

void getLocation(){
  bool newData = false;
  unsigned long chars;
  unsigned short sentences, failed;
  String respuesta="";
  for (unsigned long start = millis(); millis() - start < 1000;){
      while (ss.available()){
        char c = ss.read();
        // Serial.write(c); // uncomment this line if you want to see the
GPS data flowing
        if (gps.encode(c)) // Did a new valid sentence come in?
         newData = true;
      }
  }
  if (newData){
    float flat, flon;
    unsigned long age;
    String lat="";
    String lon="";
    gps.f_get_position(&flat, &flon, &age);
    latitud=flat ==
TinyGPS::GPS_INVALID_F_ANGLE?"0.0":floatToString(flat,6);
    longitud=flon == TinyGPS::GPS_INVALID_F_ANGLE ? "0.0" :
floatToString(flon,6);
  }
}
void setup() {
  Serial.begin(115200);
```

.

```
  //Inicialización del Led
  pinMode(LET_OUT,OUTPUT);

  //Inicialización del Led Regulado
  pinMode(LET_REG_OUT,OUTPUT);



  // Inicialización del DHT22
  dht.begin();
  //thing.add_wifi("AndroidAP","1410012695"); //Datos Jessica
  thing.add_wifi("FibraETB8168", "35918168");
  //thing.add_wifi("iPhone de Nicolas", "nicolas11");
  //thing.add_wifi("ACiscoK05", "5717455555"); //red del poli;

  thing["Data"] >> [](pson& out){
    out["Temperatura"] = dht.readTemperature();
    out["Humedad"] = dht.readHumidity();
    out["Latitud"] = latitud;
    out["Longitud"] = longitud;
   };

    thing["LED_REMOTO"] << [](pson& in){
      if(in.is_empty()){
        in = (bool) digitalRead(LET_OUT);
      }else{
        digitalWrite(LET_OUT, in ? HIGH : LOW);
      }
   };

   thing["LED_REG_REMOTO"] << [](pson& in){
     analogWrite(LET_REG_OUT, in["Value"]);
   };
}
void loop(){
  getLocation();
  thing.handle();
  delay(2000);
  Serial.println(latitud);
  Serial.println(longitud);
}
```

.