



INV/ Segundo - Bloque Telecomunicaciones – Grupo I

Proyecto análisis comparativo plataformas IoT

Presentado por:

Baquero Baquero Hector Javier

Hernandez Gomez Nelson Mauricio

Rodríguez Coca Andrés Felipe

Presentado a:

Garzon Felipe

# Introducción

El presente documento tiene como objetivo realizar un análisis entre las 26 plataformas IoT descritas en el artículo *A survey of IoT cloud providers* donde se listarán las ventajas y desventajas de cada una de las plataformas y luego se elegirá una de estas para realizar una guía de uso.

## **Planteamiento del problema**

Se requiere crear un análisis comparativo entre las diferentes plataformas IoT descritas en el artículo A survey of IoT cloud providers.

## **Planteamiento del problema**

Es necesario conocer las 26 plataformas descritas en el artículo A survey of IoT cloud providers para establecer una guía de uso de una de estas plataformas donde se evidenciará el funcionamiento de esta.

## **Objetivos**

### **General**

Realizar un análisis comparativo entre las 26 plataformas IoT descritas en el artículo anteriormente mencionado.

### **Específicos**

Realizar la lectura del artículo A survey of IoT cloud providers.

Listar las ventajas y desventajas de cada una de las plataformas expuestas en el artículo.

Crear la guía de uso de una de las plataformas IoT expuestas en el artículo.

## **Alcance**

Este documento esta dirigido al profesor y los estudiantes del módulo de telecomunicaciones del programa de Ingeniería de Software del Politécnico Grancolombiano.

## **Marco teórico**

### **Kaa**

#### **Ventajas**

NoSQL y aplicaciones de base de Big Data soportadas.

Arquitectura de microservicio personalizable: Los microservicios simplifican la personalización y aseguran una separación efectiva de las preocupaciones entre las diferentes partes de la plataforma.

Libertad tecnológica: Kaa utiliza interfaces bien definidas basadas en REST y NATS, lo que hace que sea una tecnología independiente y le permite desarrollar e implementar aplicaciones escritas en prácticamente cualquier lenguaje de programación.

Escalable y elástico: Kaa es escalable por diseño y se puede expandir elásticamente simplemente agregando más servidores al clúster de Kaa.

Libertad de despliegue: Se es libre de implementar los servicios de Kaa en cualquier lugar que prefiera . Puede hacerlo en las instalaciones de su propio centro de datos, ya sea de metal o máquinas virtuales, o puede hospedar Kaa en alguna infraestructura de nube pública.

Protocolos abiertos de IoT: Los protocolos de nivel de aplicación de Kaa utilizan por defecto la codificación JSON generalizada disponible en todos los lenguajes de programación.

## **Desventajas**

Menos módulos de hardware soportados

## **Carriots**

### **Ventajas**

Se admiten aplicaciones basadas en triggers.

API: ofrece una API propia basada en REST,(la cual puede descargarse desde Github) para poder comunicarnos con la plataforma y gestionar los datos de una forma sencilla.

Integración: ofrece esta integración con sistemas externos, a través de su API REST, el PUSH de datos y con peticiones HTTP/s o sockets. Con estos mecanismo podremos tener integración con bases de datos, ERPs, CRM, data warehouse , etc.

Hardware: En este apartado Carriots ofrece un amplio catálogo de hardware compatible, la verdad que es muy completo y prácticamente soporta cualquier tipo de hardware que haya en el mercado.

## **Desventajas**

Diseño menos amigable para el usuario.

## **Temboo**

### **Ventajas**

Se soportan aplicaciones basadas en choreos: Los dispositivos interactúan con cientos de api, bases de datos y utilidades de código con los procesos de Choreo basados en la nube.

Flexibilidad de grado industrial: Temboo trabaja con cualquier entorno u objeto que se desee poner en línea.

Hardware independiente: Bibliotecas de software diseñadas para funcionar en cualquier conjunto de chips integrados.

Integración de sensores: trabaja con cualquier sensor para monitorear visualmente tu infraestructura física en unidades del mundo real.

### **Desventajas**

No es adecuado para aplicaciones de uso intensivo de recursos.

# Sensor cloud

## Ventajas

Se puede gestionar una gran cantidad de dispositivos sensores.

SDK y API de datos abiertos: proporciona una API RESTful que permite que cualquier dispositivo o aplicación cargue datos en SensorCloud. La API se implementa como un servicio web utilizando comandos de solicitud HTTP estándar.

También se proporciona un SDK que contiene bibliotecas y código de ejemplo para Python, Java, C # .

Seguridad: Todos los datos y transacciones se producen a través de la Seguridad de la capa de transporte (TLS). SensorCloud está actualmente construido sobre Amazon Web Services, que proporciona una plataforma segura y confiable de clase mundial .

LiveConnect: LORD MicroStrain proporciona una puerta de enlace WSDA ® que recopila datos de los sensores MicroStrain y los envía a SensorCloud directamente. LiveConnect le permite interactuar con esta puerta de enlace WSDA y los sensores correspondientes desde cualquier parte del mundo. Configure los ajustes de red del sensor, inicie el muestreo y vea datos en tiempo real, todo de forma remota.

## Desventajas

Los dispositivos de código abierto son difíciles de obtener.

# Xibely – Google Cloud Platform

## Ventajas

Seguridad completa: Utiliza un sistema de seguridad integral gracias a la autenticación mediante claves asimétricas con TLS 1.2 y a los certificados firmados por una autoridad de certificación, que se pueden utilizar para verificar la propiedad de los dispositivos. Los dispositivos que cumplan los requisitos de Cloud IoT Core son capaces de proteger toda la pila.

Completamente administrado y escalable

El servicio no utiliza servidores ni requiere ninguna instalación previa de software. Se escala instantáneamente sin límites mediante el escalado horizontal de Google Cloud Platform.

Control de acceso a nivel de roles

Aplica los roles de gestión de identidades y acceso a los registros de dispositivos para controlar el acceso de los usuarios a los dispositivos y a los datos.

Un solo sistema global

Conecta todos los dispositivos y pasarelas a Google Cloud siguiendo protocolos estándar, como MQTT y HTTP, a través de los puntos de conexión de protocolos y gestiona todos tus dispositivos como si se tratara de un único sistema global. El servicio utiliza Cloud Pub/Sub de forma subyacente, que retiene datos durante siete días.

Despliegue de dispositivos a escala



Usa las API REST para gestionar automáticamente el registro, despliegue y funcionamiento de dispositivos a escala. También puedes usar las API para visualizar y actualizar el estado y las propiedades de los dispositivos, aunque no estén conectados.

## **Ayla's IoT cloud fabric**

### **Ventajas**

El Agente de Producción Wi-Fi de Ayla es una pila de software de redes y comunicaciones que viene precargada en una serie de conjuntos de chips diferentes de los principales fabricantes de módulos. El Agente de producción de Wi-Fi no requiere codificación adicional y es independiente de la aplicación, y es compatible con la mayoría de los sistemas basados en microcontroladores. Los clientes deben construir su aplicación en un Host MCU separado y conectarse al módulo Ayla a través de SPI o UART.

### Gateway dinámico

Las bibliotecas de la API de Dynamic Gateway de Ayla se proporcionan para que los clientes puedan desarrollar soluciones de pasarela que admitan cualquier protocolo local, incluidos ZigBee, Z-Wave, BLE, protocolos propietarios y mucho más. Las bibliotecas de Dynamic Gateway API están integradas en productos de pasarela basados en Linux para permitir que los nodos (dispositivos finales) en una red local se virtualicen en Ayla Cloud y estén disponibles para todos los dispositivos nativos de administración, incluidas las actualizaciones de firmware OTA y las capacidades de habilitación de aplicaciones.

**Adaptador de nube** Para enriquecer los análisis y las aplicaciones aprovechando los datos del dispositivo, puede ser necesario ingerir datos de fuentes externas de la nube, como SAP, Salesforce o sistemas ERP heredados. El adaptador de nube de Ayla proporciona un mecanismo sin problemas para conectarse a estas fuentes externas para la comunicación de datos bidireccional.

**Herramientas de aplicación móvil** El Ayla Mobile SDK es una biblioteca que habilita la funcionalidad móvil central, como la incorporación de usuarios, la administración de sesiones, las notificaciones y los horarios.

La aplicación White Label (AMAP) de Ayla reduce aún más el tiempo de comercialización al proporcionar una aplicación preconfigurada y personalizable a la que los desarrolladores pueden aplicar fácilmente gráficos y funciones adicionales.

## **Controles de seguridad**

### **Dispositivos autenticados con un par de claves único.**

Comunicación en modo LAN encriptada entre la aplicación móvil y el dispositivo.

Cifrado de datos: HTTPS cifrado con TLS, canal UDP con cifrado AES-128.

Control de acceso en capas para evitar la violación de un dispositivo que comprometa todo el sistema.

Pruebas de penetración por parte de terceros proveedores.

## **Desventajas**

No es adecuado para desarrolladores de pequeña escala.

## **Thethings.io**

### **Ventajas**

Funciones de código de nube, activadores y trabajos

Crea disparadores, trabajos y funciones y ejecuta tema.

Enviar notificaciones a través de twitter, sms o llamadas de voz

### **Desventajas**

Carece de autosostenibilidad, depende de servicios web de terceros.

## **Exosite**

### **Ventajas**

El desarrollo del sistema es fácil.

Visualización de datos en tiempo real y soporte analítico a los usuarios.

Los kits son compatibles para el diseño y la implementación de soluciones de IoT. facilita que los dispositivos conectados se conecten y transfieran los contextos a través de CoAP, y

UDP de un solo disparo API.

## **Desventajas**

Falta de provisión de Big Data

## **Arrayent Connect TM**

### **Ventajas**

Flexible de usar.

Ayuda a alojar todos los dispositivos virtualizados a través de actualizaciones de firmware Over-the Air (OTA) en una baja tasa de latencia de datos.

Fuentes de datos seguros, confiables y escalables ayudan a los usuarios a ser recuperados, procesados y entregados.

Se admiten plataformas de computación multiplataforma que van desde microcontroladores de 8 bits a procesadores de 32 bits.

Las alertas de correo electrónico y SMS se envían a los clientes con notificaciones push basadas en iOS y Android

### **Desventajas**

Los servicios basados en activadores están retrasados.

## **Open remote**

### **Ventajas**

Servicio de nube abierta soportado.

Permite al usuario integrar cualquier dispositivo: diseño de protocolo y uso de recursos disponibles como iOS, Android o navegadores web

Al integrar Open Remote se pueden desarrollar soluciones que pueden integrar una gran cantidad de protocolos desde WiFi hasta ZigBee

### **Desventajas**

Demasiado costoso para los desarrolladores

## **Arkessa**

### **Ventajas**

Faceta de diseño habilitada para empresas.

Capacita a las empresas para aprovechar el IoT para el desarrollo de nuevas fuentes de ingresos a través de una mejor satisfacción del cliente y mejores valores potenciales recibidos como flujos de datos desde dispositivos remotos.

Modelo PaaS para formular un único portal de administración empresarial para una administración de dispositivos eficiente y optimizada mediante la integración de flujos de datos de

máquina con el CRM disponible, ERP, big data y otros sistemas de análisis.

### **Desventajas**

Las aplicaciones de visualización no son lo suficientemente adecuadas.

### **Axeda**

#### **Ventajas**

Gestión de datos basados en M2M.

Integra el aprendizaje M2M en el proceso comercial diario, por ejemplo, desde medidas preventivas de seguridad de datos hasta el aprovisionamiento de dispositivos y configuración

Realiza funciones clave como el servicio de aplicaciones, el marco de integración y la gestión de datos

#### **Desventajas**

Carece de autosuficiencia, dependiente de terceros.

### **Oracle IoT cloud**

#### **Ventajas**

Soporte de base de datos.

Proporciona seguridad de extremo a extremo normalizada para todo tipo de dispositivos, datos y conectividad heterogénea

Mueve la idea muy rápidamente a la ejecución con un riesgo mínimo.

Mientras analiza, procesa flujos de datos entrantes en tiempo real con filtrado de eventos, correlación, y agregación.

El análisis predictivo y de big data hace que Oracle sea cada vez más receptivo con la detección de anomalías y mecanismos de alerta basados en reglas.

## **Desventajas**

Carece de conectividad de dispositivos de código abierto debido a la restricción de tamaño.

## **Nimbits**

### **Ventajas**

Fácil de adoptar para desarrolladores.

Está diseñado para permitir a los desarrolladores construir clusters integrados, altamente escalables y altamente disponibles para las necesidades de registro de datos.

Puede ejecutarse en el motor de aplicaciones de Google <sup>TM</sup>, Amazon EC2, máquina virtual basada en Ubuntu Linux KVM e infraestructuras basadas en Jetty (un servidor web J2EE)

## **Desventajas**

Procesamiento de consultas en tiempo real.

## **Thing Worx**

### **Ventajas**

La creación intensiva de datos de aplicaciones es fácil.

Proporciona Infraestructura basada en M2M e IoT como un servicio donde el diseño basado en modelos se incorpora con SQUEAL (Búsqueda, Consulta, Análisis) para incluir en él la inteligencia basada en la búsqueda.

La función de codificación cero está pensada para que los desarrolladores reduzcan el tiempo de la situación del mercado en la entrega del producto.

Las plataformas de hardware IoT desarrolladas por Intel son fácilmente compatibles.

### **Desventajas**

Se puede conectar un número limitado de dispositivos.



## **InfoBright**

### **Ventajas**

Arquitectura de Knowledge Grid y procesamiento de big data.

Viene con una variedad de versiones de mercado (por ejemplo, comunidad y empresa) para las empresas que requieren problemas de rendimiento y capacidad compatibles con InfoBright.

Está bien equipado con motores de rendimiento de carga y consulta distribuidos capaces de cargar TB por hora y puede manejar Peta Bytes de datos mientras facilita la compresión de datos en el rango de  $10: 1e40: 1$

### **Desventajas**

Faltan servicios estadísticos.

## **AerCloud**

### **Ventajas**

Servicios escalables M2M.

Permite a las aplicaciones de los usuarios conexión a muchos dispositivos, garantizando la fiabilidad, seguridad y bases de datos.

Plataforma de servicio PaaS, ofreciendo un modelo de pago a medida que crece, implicando un motor de reglas que procesa los datos en tiempo real.

## **Desventajas**

No es adecuado para desarrolladores.

## **ARKESSA.**

### **Características principales**

Disponible como una solución PaaS, arrendado o licenciado

Alojado en la nube, en el centro de datos o en las instalaciones

Compatible con los nuevos dispositivos IO o dispositivos M2M legado

Control en tiempo real y datos consolidados

Portal de gestión de patrimonio único para la gestión de dispositivos

Oracle certificada como Exastack Ready

Integrar datos de la máquina con el CRM, ERP, los grandes datos y los sistemas de análisis

Datos totalmente encriptada cumple con las regulaciones internacionales

Soporta GSM, GPRS, 3G, 4G, y el satélite

### **Ventajas**

Ayuda en la solución de problemas como conectividad general, monitoreo, control y administración entre dispositivos basados en IoT y empresas.

Ayuda a capacitar a las empresas para aprovechar el IoT para el desarrollo de nuevas fuentes de ingresos a través de una mejor satisfacción del cliente y mejores valores potenciales recibidos como flujos de datos desde dispositivos remotos.

Se desarrolla por medio del modelo de PaaS para formular un único portal de administración empresarial para servicios de administración de dispositivos eficientes y optimizados.

Arkessa permite las conexiones de Internet de las Cosas (IoT) y de Máquina a Máquina (M2M) en cualquier parte del mundo.

## **Desventajas**

Las aplicaciones de visualización no son lo suficientemente adecuadas.

## **Arrayent Connect**

### **Ventajas**

Interfaz de programación de aplicaciones que permite a los fabricantes de marcas aprovechar fácilmente otros ecosistemas importantes.

El Marco de la API EcoAdaptor permite la interoperabilidad de nube a nube, al tiempo que preserva la seguridad operativa y permite el desarrollo de terceros.

Para permitir que sus marcas gestionen, controlen y monitoreen productos conectados e interoperables, brindando una mejor experiencia a los consumidores

Permite que los usuarios se conecten a dispositivos de mano inteligentes y aplicaciones web

de valor agregado.

Es un sistema operativo de IoT que aprovecha el modelo de Software como Servicio (SaaS).

Sus fuentes de datos seguros, confiables y escalables ayudan a los usuarios a ser recuperados, procesados y entregados.

Permite a las marcas conectarse y acercarse más a sus clientes al permitirles conectar, monitorear y controlar sus productos IoT de forma remota desde cualquier parte del mundo.

### **Desventajas**

Los servicios basados en activadores están retrasados.

## **IBM IOT**

### **Ventajas**

La identidad del dispositivo está habilitada.

Al pertenecer a IBM puede acceder a características, No-SQL, Big data.

### **Desventajas:**

Difícil en la creación de prototipos.

## **InfoBright**

### **Ventajas**

Aprovecha las empresas al proporcionar su arquitectura de cuadrícula de conocimiento-esquema.

Permite a las empresas almacenar, analizar y actuar sobre la pila de datos generados por la maquina al mismo tiempo.

Viene con una variedad de versiones de mercado (por ejemplo, comunidad y empresa) para las empresas que requieren problemas de rendimiento y capacidad compatibles con InfoBright.

Está equipado con motores de rendimiento de carga y consulta distribuidos capaces de cargar TB por hora y puede manejar Peta Bytes de datos mientras facilita la compresión de datos.

### **Desventajas**

Faltan servicios estadísticos.

## **JASPER CONTROL CENTER**

### **Ventajas**

Patrones de comportamiento basados en reglas habilitados.

Altamente configurable.

Alto rendimiento basado en la monitorización en tiempo real.

Permite envío y recepción de datos a través de protocolo como COAP.

Permite conexión con APIS.

Adecuado para servicios de automatización.

### **Desventajas**

No se reportan desventajas relevantes.

## **ThingSpeak.**

### **Ventajas**

Habilitación de nube publica con facilidad de activación.

Permite realizar acciones automáticas en momentos específicos.

Permite conexión con Twitter y reacciona en tiempo real.

### **Desventajas**

Menos cantidad de conectividad de dispositivos simultáneamente.

Limitado en el número de peticiones.

Interface de visualización muy básica.

# XIVELY

## Ventajas

Es un servicio empresarial en la nube de IoT basado en la tecnología Gravity Cloud.

Esta plataforma propiedad de LogMeIn ayuda a las empresas a administrar su negocio de productos conectados al abordar una serie de necesidades prácticas mediante una conectividad escalable, segura y confiable.

También allana los servicios de procesamiento de datos empresariales correctos hacia sus clientes, socios y proveedores habilitados para IoT a través de conectores API flexibles.

Xively emplea una novedosa plataforma IoT como servicio (IoTaaS) construida en su nube publica elástica. La escalabilidad elástica de la nube proporciona capacidades intuitivas de administración del ciclo de vida del dispositivo, lo que implica actividades de series de tiempo en ella.

Además, Xively facilita el archivado de datos, la activación condicional, el aprovisionamiento de dispositivos en tiempo real con la activación, la gestión de mensajes y el enrutamiento.

Facilita el control sobre los dispositivos, Xively ha creado una mesa de trabajo para desarrolladores y una consola de administración de dispositivos que puede ser operada por un principiante.

También es capaz de soportar millones de dispositivos por medio de APIS RESTful. El uso de los formatos de datos JSON, XML y CSV ha evolucionado su efectividad en términos de dispositivos asociativos que se pueden monitorear a través de bibliotecas de clientes pree

valuadas construidas sobre iOS, Android, JavaScript, así como bibliotecas de servidores diseñadas para aplicaciones basadas en web de gama alta. lenguajes como, Ruby, Python, y Java.

Fácil de integrar con dispositivos.

### **Desventajas**

Los servicios de notificación son mínimamente presentes.

## **ECHELON.**

### **Ventajas**

Perspectiva industrial.

Ofrece una suite completa de recursos que incluyen interfaces de comunicación y paquetes de software para el desarrollo de dispositivos.

Tiene alta compatibilidad en los servicios como APIS.

Permite comunicación basada en publicador suscriptor.

### **Desventajas**

Carece de escenario de desarrollo para principiantes.

No se recomienda el uso para pequeños productos.

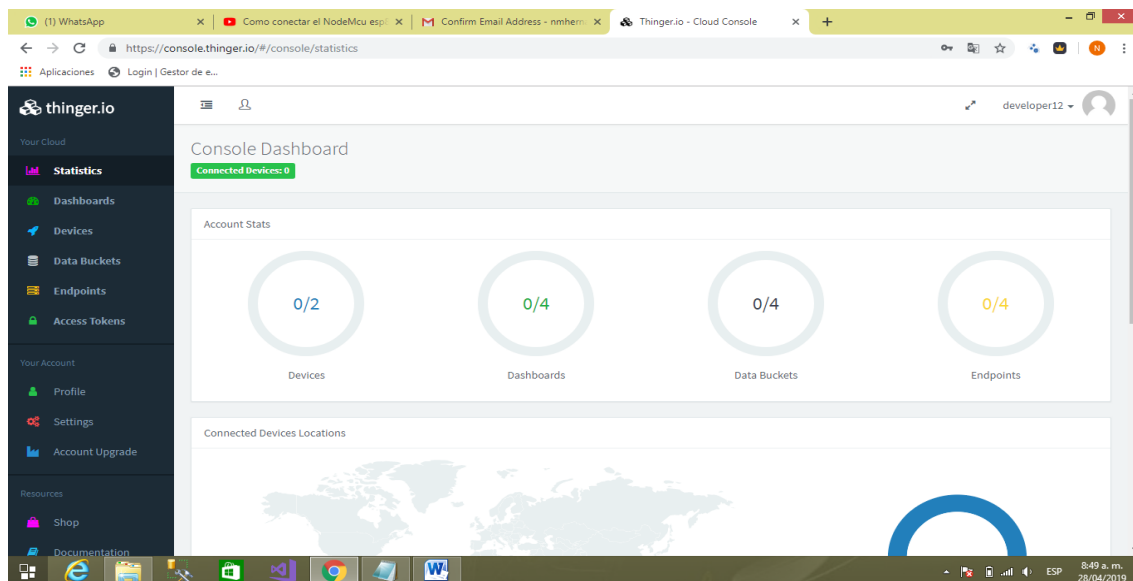


# Implementación Iot

Después de realizar el análisis comparativo de las 26 plataformas de IoT se eligió la plataforma Thinger.io para crear la guía de uso que se expondrá a continuación.

## Guía de uso de Thinger.io

Creamos una cuenta en thinger.io

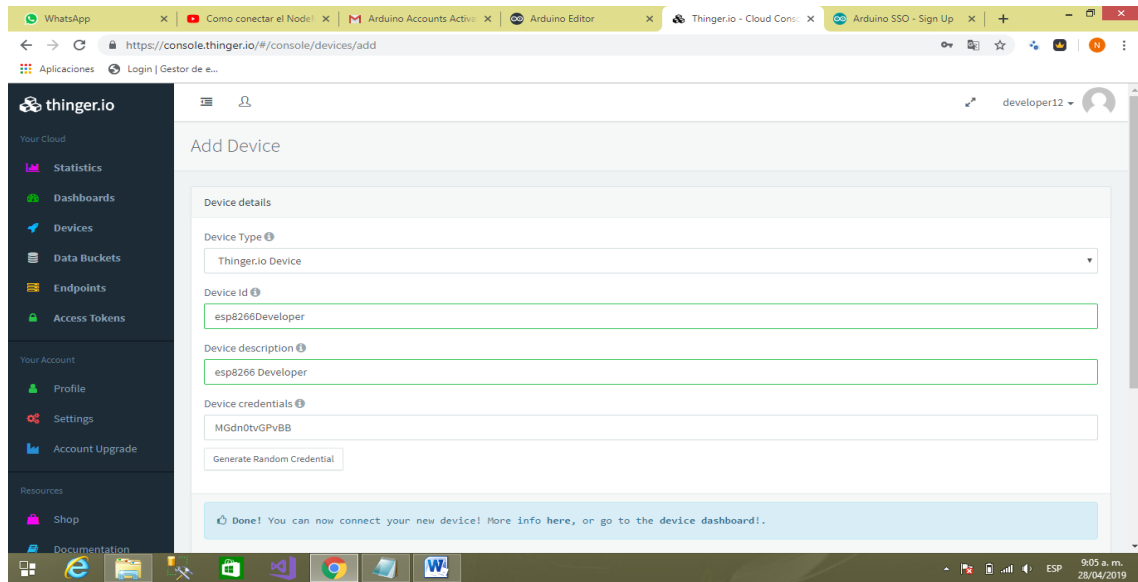


Elaboración propia

Figura 1. Dashboard Things.io

## Devices/Add Device

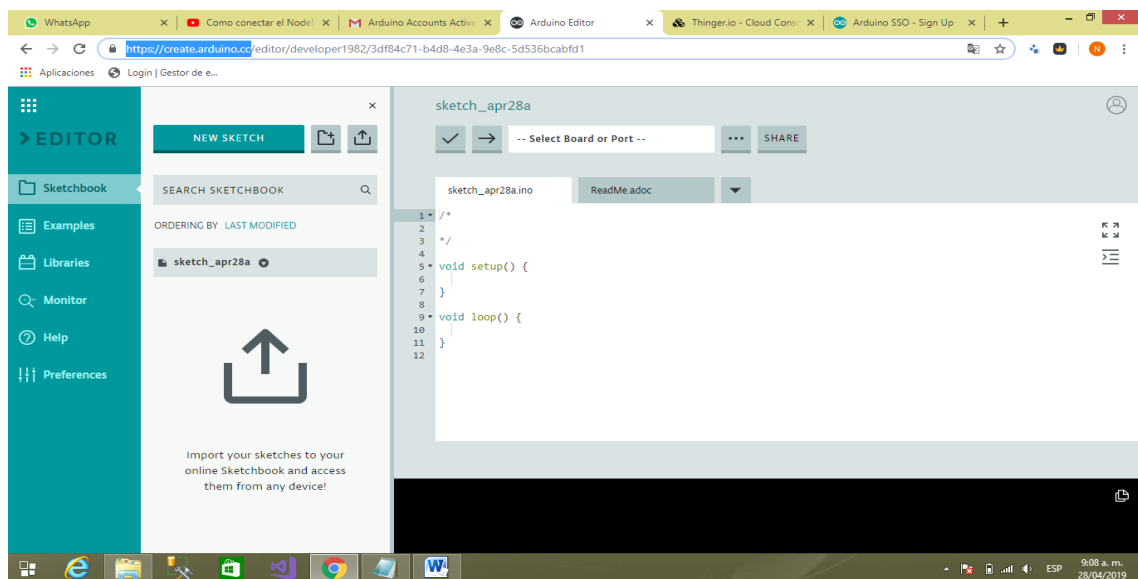
Ingresamos id y descripción (las que queramos). Para device credentials le damos click en generate random.



Elaboración propia

Figura 2. Add Device

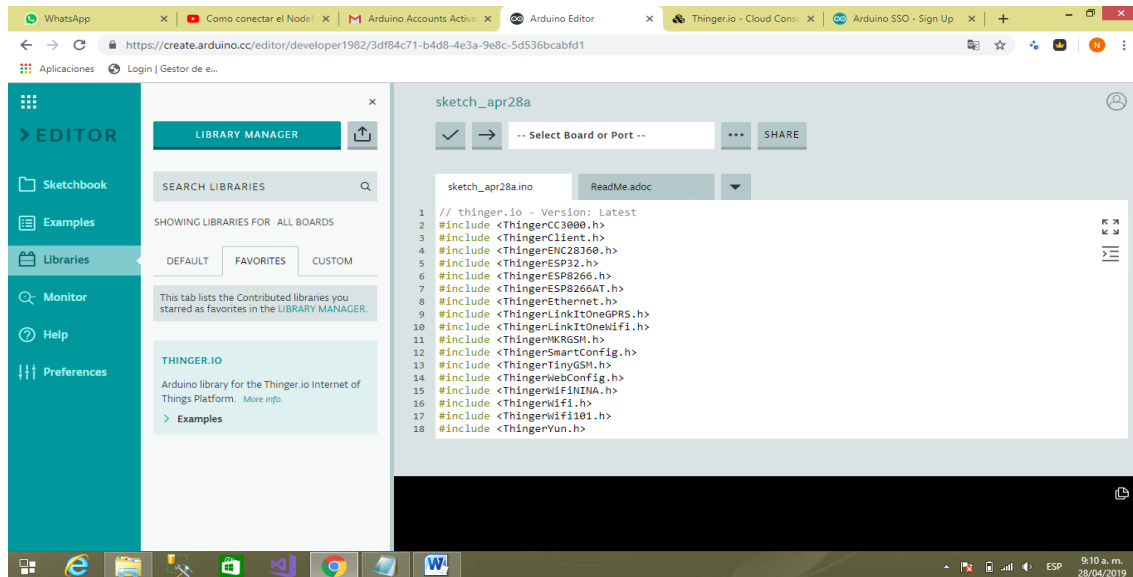
Por otra parte creamos cuenta en arduino online <https://create.arduino.cc>



Elaboración propia

Figura 3. Dashboard Arduino Online

Importamos la librería de thinger.io

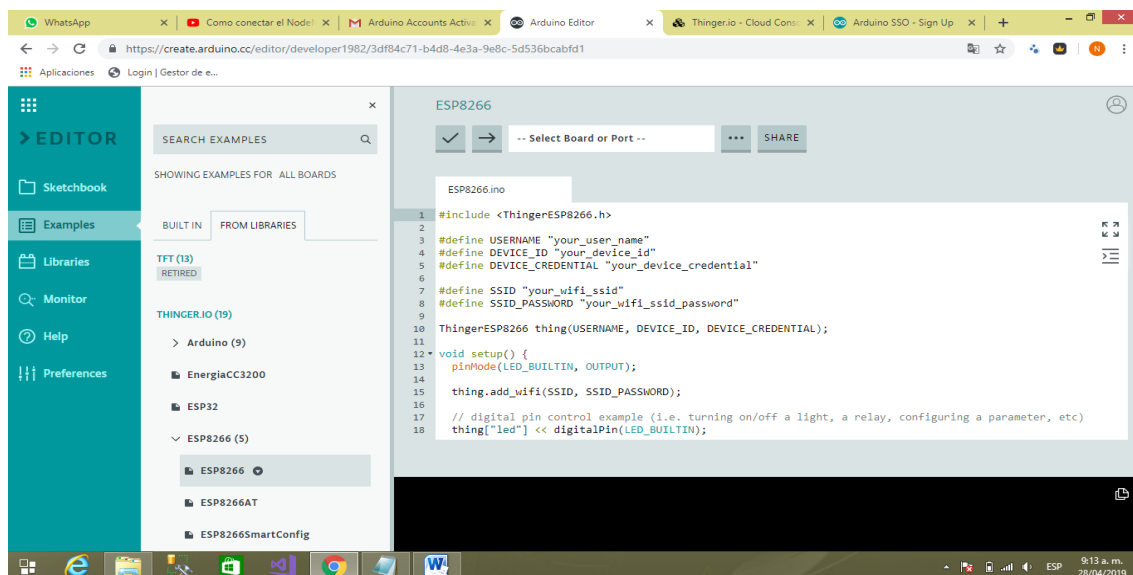


Elaboración propia.

Figura 4. Importación librería Thinger.io

Examples/From libraries/thinger.io/esp8266

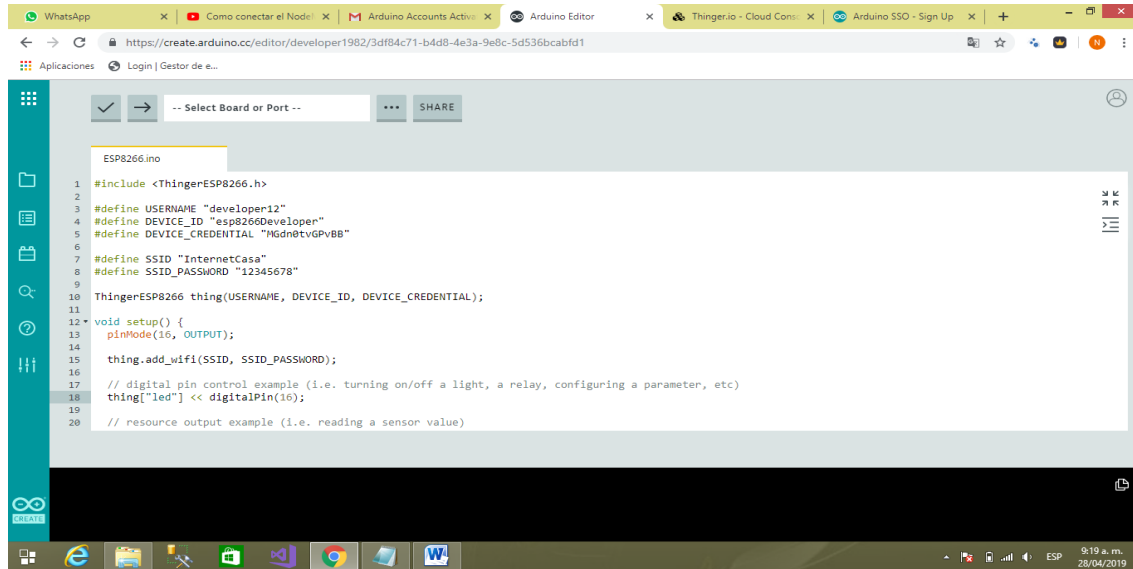
Así cargamos la plantilla con el código para la conexión



Elaboración propia.

Figura 5. Plantilla con el código de conexión

Ingresamos lo datos del dispositivo creado en thinger.io; además el pin de la placa esp8266 con el que queremos hacer la demostración, en este caso es el 16(pinMode)

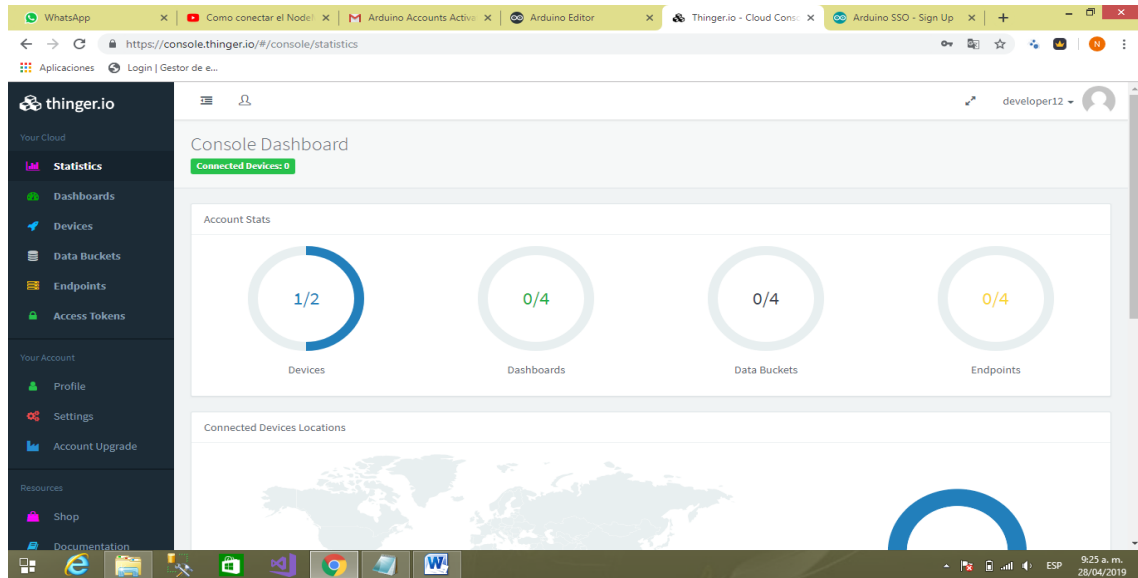


```
ESP8266.ino
1 #include <ThingyESP8266.h>
2
3 #define USERNAME "developer12"
4 #define DEVICE_ID "esp8266Developer"
5 #define DEVICE_CREDENTIAL "HgdntvGPvBB"
6
7 #define SSID "InternetCasa"
8 #define SSID_PASSWORD "12345678"
9
10 ThingyESP8266 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);
11
12 void setup() {
13   pinMode(16, OUTPUT);
14
15   thing.add_wifi(SSID, SSID_PASSWORD);
16
17   // digital pin control example (i.e. turning on/off a light, a relay, configuring a parameter, etc)
18   thing["led"] << digitalPin(16);
19
20   // resource output example (i.e. reading a sensor value)
```

Elaboración propia.

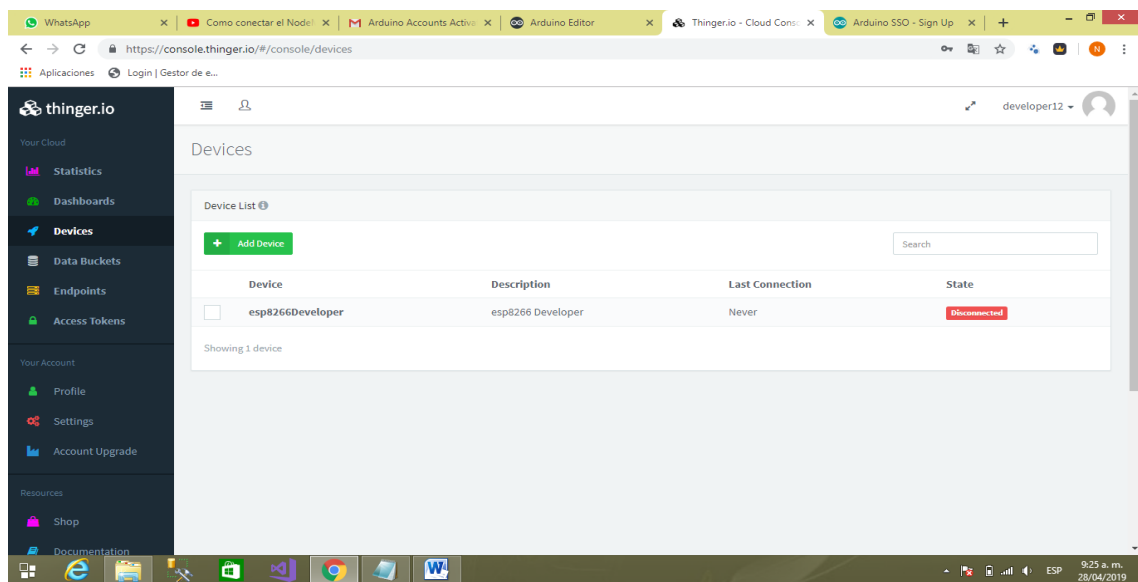
Figura 6. Ingreso de datos del dispositivo.

En thinger.io vamos a Statistics/Devices



Elaboración propia.

Figura 7. Statistics Devices



Elaboración propia.

Figura 8. Devices.

Nos aparece el dispositivo desconectado, eso dado obviamente porque no tenemos la placa esp8266.

## Conclusión

En búsqueda de una plataforma que fuese objeto de la implementación, realizamos la investigación sobre plataformas que se encuentran en el mercado, al revisar las plataformas cloud 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 la inversión. Pasando por las plataformas tipo Open Source encontramos que las más compatibles nuestra necesidad son Kaa Project y Thinger.io. Por lo tanto, se concluye que Thinger.io es la más apropiada, dado 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.

# Referencias

Partha Pratim Ray.2016. A survey of IoT cloud platforms. Politécnico Grancolombiano

Kaa Enterprise IoT platform. 2019. Kaa Enterprise IoT platform. Recuperado de

<https://www.kaaproject.org/>

Altair SmartWorks. 2019. Open scalable IoT solutions. Recuperado de

<https://www.altairsmartworks.com/>

Temboo. 2019. Temboo. Recuperado de

<https://temboo.com/>

SensorCloud. 2019. SensorCloud. Recuperado de

<https://sensorcloud.com/>

Google Cloud IoT. 2019. Google Cloud IoT. Recuperado de

<https://cloud.google.com/solutions/iot/>

Ayla Networks. 2019. Ayla's IoT cloud fabric. Recuperado de

<https://www.aylanetworks.com/>

Thethings.io. 2019. Thethings.io. Recuperado de

<https://thethings.io/>

Exosite. 2019. Exosite. Recuperado de



<https://exosite.com/>

Prodea. 2018. Arrayent Connect TM. Recuperado de

<https://www.arrayent.com/>

Openremote. 2016. Openremote. Recuperado de

<http://www.openremote.com/>

Arkessa. 2019. Arkessa. Recuperado de <https://www.arkessa.com/>

PTC. 2019. Thingworx. Recuperado de

<https://www.ptc.com/en/products/iot/thingworx-platform>

Oracle. 2019. Oracle cloud / Internet of Things. Recuperado de

<https://cloud.oracle.com/iot>

Cisco Systems. 2019. Jasper. Recuperado de <https://www.jasper.com/>

Aeris. 2019. AerCloud. Recuperado de <https://www.aeris.com/>

Thingspeak. 2019. Thingspeak. Recuperado de <https://thingspeak.com/>

Plotly. 2019. Plotly. Recuperado de <https://plot.ly/>

IBM. 2019. IBM IoT. Recuperado de <https://internetofthings.ibmcloud.com/>