

DISEÑO DE UN SISTEMA EN CLOUD PARA CONTROLAR DISPOSITIVOS IOT VÍA INTERNET.

Autor: Diana Marcela Ruiz Martínez

UNIVERSIDAD MILITAR NUEVA GRANADA

FACULTAD DE INGENIERIA

INGENIERIA EN TELECOMUNICACIONES

BOGOTÁ

2016

DISEÑO DE UN SISTEMA EN CLOUD PARA CONTROLAR DISPOSITIVOS IOT VÍA INTERNET.

Autor: Diana Marcela Ruiz Martínez

Trabajo de grado para optar por el título de ingeniero en telecomunicaciones

Tutor: Ing. Edward Paul Guillén Pinto, MSc, PhD

UNIIVERSIDAD MILITAR NUEVA GRANADA

FACULTAD DE INGENIERIA

INGENIERIA EN TELECOMUNICACIONES

BOGOTÁ

2016

I. Tabla de contenido

I. Tabla de contenido.....	2
II. Lista de tablas.....	4
III. GLOSARIO	5
1. INTRODUCCIÓN	8
1.1. PROBLEMA	9
1.2 OBJETIVOS	9
1.2.1. Objetivo General	9
1.2.2. Objetivos específicos.....	9
2. RECOPIACIÓN DE LA INFORMACIÓN	10
3. ANÁLISIS PLATAFORMAS CLOUD IOT	21
4. ANÁLISIS PROTOCOLOS IOT	33
5. PRUEBAS.....	38
5.1.1 Escenario de Prueba	39
5.2.1 Escenario de Prueba	49
6. CONCLUSIONES	52
REFERENCIAS.....	53

Lista de figuras

<i>Figura 1</i> Estructura de IoT por la recomendación Y2060.	13
<i>Figura 2</i> Modelo de Referencia IoT presentado por la UIT en la recomendación Y206	24
<i>Figura 4</i> <i>Arquitectura de Servicio de Microsoft Azure.</i>	25
<i>Figura 5</i> <i>.Modelo de Servicio Nube.</i>	31
<i>Figura 6</i> <i>. Flujograma modelo de servicios</i>	32
<i>Figura 7</i> <i>Escenario de Prueba con AWS.</i>	40
<i>Figura 8</i> <i>Creación de Fuente AWS.</i>	41
<i>Figura 9</i> <i>Creación cosa AWS IoT</i>	41
<i>Figura 10</i> <i>Visualización cosa AWS IoT.</i>	42
<i>Figura 11</i> <i>Conexión de dispositivo a AWS IoT.</i>	42
<i>Figura 12</i> <i>Creación de certificados AWS IoT.</i>	43
<i>Figura 13</i> <i>Certificados y claves AWS IoT.</i>	43
<i>Figura 14</i> <i>Conexión SDK AWS IoT.</i>	44
<i>Figura 15</i> <i>Punto final REST API AWS IoT</i>	45
<i>Figura 16</i> <i>Compilación ejemplo AWS SDK C.</i>	46
<i>Figura 17</i> <i>Conexión Raspberry Pi2 con AWS.</i>	46
<i>Figura 18</i> <i>Conexión a Cliente MQTT AWS..</i>	47
<i>Figura 19</i> <i>Costos por Servicios usados AWS.</i>	48
<i>Figura 20</i> <i>Predicción de Costos AWS .</i>	48
<i>Figura 21</i> <i>Escenario de prueba con Microsoft Azure</i>	49
<i>Figura 22</i> <i>Creación IoT Hub Azure</i>	49
<i>Figura 23</i> <i>Servicios implementados IoT Suite Azure.</i>	50
<i>Figura 24</i> <i>Prueba Piloto a AWS..</i>	52

II. Lista de tablas

<i>Tabla 1. Plataformas IoT.</i>	18
<i>Tabla 2. Análisis Plataformas IoT.</i>	19
<i>Tabla 3. Dispositivos con conexión a Internet.</i>	20
<i>Tabla 4. Análisis de Dispositivos.</i>	21
<i>Tabla 5. Comparación entre AWS IoT y Microsoft Azure IoT.</i>	22
<i>Tabla 6. Paralelo entre servicios de AWS IoT y Microsoft Azure IoT.</i>	26
<i>Tabla 7. Paralelo de costos entre AWS IoT y Microsoft Azure IoT.</i>	27
<i>Tabla 8. Protocolos IoT].</i>	35
<i>Tabla 9. Análisis de Protocolos.</i>	36

III. GLOSARIO

Cloud Computing	“Es un modelo para habilitar acceso conveniente por demanda a un conjunto compartido de recursos computacionales configurables (Servicios a través de internet), que pueden ser rápidamente aprovisionados y liberados con un esfuerzo mínimo de administración o de interacción con el proveedor de servicios” [1].
Internet of Things (IoT)	“La infraestructura mundial para la sociedad de la información que propicia la prestación de servicios avanzados mediante la interconexión de objetos (físicos y virtuales) gracias a la interoperabilidad de tecnologías de la información y la comunicación presentes y futuras” [2].
Microsoft Azure	Plataforma de Cloud de Microsoft que permite la ejecución y el despliegue de aplicaciones y servicios en la nube [3].
IoT Hub	Concentrador de dispositivos IoT. Gestiona y conecta los dispositivos IoT a Cloud. [4]
Amazon Web Services	Es una plataforma de Cloud administrada, que permite conectar dispositivos a la nube y a otros dispositivos. Siendo capaz de procesar y enrutar mensajes a puntos de enlace de AWS [5].

Gateway IoT	Conecta sistemas heredados y nuevos, permitiendo el flujo de datos seguros y sin problemas entre los dispositivos de última generación y la nube [6].
Things (Cosas)	“Un objeto del mundo físico (objetos físicos) o del mundo de la información (objetos virtuales) que se puede identificar e integrar en las redes de comunicaciones” [2].
SDK	Son kits de desarrollo de software que facilitan la conexión de un dispositivo hardware con la nube. Estos incluyen bibliotecas cliente, guía del desarrollador y guía de puertos para fabricantes [7].
IaaS	Infraestructura como un Servicio. Modelo de Negocio Cloud donde el proveedor proporciona la infraestructura, como los recursos de red, almacenamiento, o de cómputo y el cliente gestiona su propio software en la parte superior del hardware del proveedor [8].
PaaS	Plataformas como un Servicio. Modelo de negocio cloud, donde el proveedor proporciona la infraestructura y una plataforma de desarrollo de aplicaciones, mientras los clientes administran solo sus propias aplicaciones [8].

SaaS	Software como un Servicio. Modelo de negocio cloud, donde las aplicaciones están alojadas por un proveedor y puestos a disposición de los clientes en una base de suscripción [8].
MQTT	(Message Queue Telemetry Transport). Es un protocolo usado para comunicación maquina a máquina (M2M), basado en el modelo Publicación / Suscripción y con arquitectura cliente/servidor donde el cliente se conecta a un Broker para interactuar en la red [9].
XMPP	(Stands for Extensible Messaging and Presence Protocol). Es un protocolo para comunicación en tiempo real, con arquitectura cliente/servidor y con formato de texto XML [9].
DDP	(Data Distribution Service). Es un protocolo usado para comunicación M2M y dispositivos a dispositivo y modelo Publicación/Suscripción [9].
AMQP	(Advanced Message Queuing Protocol). Es un protocolo que soporta el modelo de comunicaciones punto a punto y Publicación/Suscripción [9].
HTTP	Protocolo de Transferencia de Hipertexto [9].
CoAP	Protocolo de aplicación restringida [9].

1. INTRODUCCIÓN

En este documento se presenta el diseño de un modelo en cloud que permite conectar y controlar dispositivos IoT desde internet. Este modelo se establece a partir del análisis de costos de IoT, para identificar la viabilidad de implementación de IoT, teniendo en cuenta cuales son las plataformas IoT existentes y sus protocolos de comunicación.

Para la elaboración de este proyecto se establecieron varias etapas.

- I. La primera etapa consistió en la recopilación de la información, sobre plataformas Cloud que brindan el servicio de IoT, protocolos de IoT y dispositivos compatibles con las plataformas IoT.
- II. La segunda etapa consistió en el análisis sobre plataformas Cloud para IoT existentes.
- III. En la tercera etapa se realizó el análisis de Protocolos IoT existentes.
- IV. En la cuarta etapa se realizaron las pruebas pertinentes, teniendo en cuenta los resultados obtenidos de las etapas anteriores.
- V. Por último, en la quinta etapa se elaboró el presente documento.

Este Trabajo de Grado se encuentra asociado al proyecto de investigación INV-ING- 2114 de la Universidad Militar Nueva Granada.

1.1. PROBLEMA

Actualmente, existen muchos dispositivos, sensores y cosas con conectividad a internet; esta evolución en la tecnología generó un nuevo concepto, “Internet of Things”, que permite la interacción entre los objetos físicos he internet, para la recopilación, manejo y análisis de datos. Pero en este momento, no hay como desplegar IoT de forma fácil y aunque existen concentradores de dispositivos individuales, se requiere uno unificado. Es decir, un servidor en la nube, de bajo costo.

Por esta razón, se necesita de un modelo, donde se puedan distribuir sensores fácilmente, que sea accesible para cualquier persona. Este modelo permitirá cuantificar el despliegue de IoT evaluando los costos de implementación.

1.2 OBJETIVOS

En esta sección se presentan los objetivos aprobados por el Comité de Opción de grado del programa de Ingeniería en Telecomunicaciones de la Universidad Militar Nueva Granada el día 23 de noviembre de 2015.

1.2.1. Objetivo General

Diseñar un sistema en Cloud, que permita controlar dispositivos IoT a través Internet en una plataforma unificada.

1.2.2. Objetivos específicos

1. Analizar los protocolos de IoT existentes, con el fin de determinar cuál es el protocolo más adecuado para el diseño del sistema en cloud.
2. Trabajar al menos sobre dos plataformas que ofrecen control de dispositivos IoT, con el fin de desarrollar el sistema sobre alguna de ellas.
3. Analizar los tipos de servicios que se pueden ofrecer en el sistema Cloud para controlar dispositivos IoT.
4. Montar un Servicio piloto sobre el sistema.
5. Realizar el análisis de costos por servicios y por dispositivos.

2. RECOPILOCIÓN DE LA INFORMACIÓN

2.1. Estado del Arte.

Dave Evans, Director y Jefe de Tecnología de Cisco en el 2011, habla en su artículo: “Internet of Things La próxima evolución de Internet lo está cambiando todo”, de Internet of Things (IoT) como la tecnología que lo cambiaría todo, afirmación basada en el impacto que Internet ha tenido en los sectores de la educación, la comunicación, los negocios, la ciencia, el gobierno y la humanidad. Según el, el Internet es una de las creaciones más importantes y potentes de la historia de la humanidad, siendo IoT la evolución del internet, que trae consigo, avances en la capacidad para recopilar, analizar y distribuir datos [10].

En este mismo artículo, se menciona el estudio realizado por IBSG de Cisco en abril del 2011 en el que se relaciona el crecimiento de IoT, teniendo como variables la cantidad de población mundial y la cantidad de dispositivos conectados a internet. Este estudio revela que realmente IoT es un concepto muy reciente, que tuvo sus inicios a mediados del 2008 y 2009 cuando se encontraron conectados más dispositivos que personas a internet. Lo que llevo a una estimación de 25.000 millones de dispositivos conectados a Internet en 2015 y 50.000 millones en 2020 [10].

Por otro lado, para James Manyika director y Michael Chui Partner de McKinsey Global Institute (MGI), el internet de las cosas es la nueva forma para hacer negocios, donde los sensores que se emplean pueden rastrear y controlar los objetos físicos, que, en empresas de equipos, puede reducir los costos de mantenimiento hasta en un 25%, reducir las interrupciones no planificadas hasta en un 50%, y extender la vida de las máquinas por año. Además, ellos estiman que hacer uso de productos que reportan la forma en la que están siendo utilizados, puede generar una mayor penetración en los clientes y causar un impacto económico total para el 2025 de alrededor de \$3,9 billones a \$ 11.1 billones [11].

El Internet de las Cosas es la muestra de la relación que existe entre el mundo real y el mundo virtual, donde los dispositivos de uso cotidiano pueden ser manejados remotamente y además pueden actuar como puntos de acceso a los servicios de Internet y/o cloud que se define como una “técnica de intercambio de recursos en servidores, almacenamiento y otras infraestructuras de computación en múltiples lugares que están conectados por redes” [12] mediante plataformas en las que se destacan Microsoft Azure, Amazon, Google Apps, Wyliodrin, entre otras, las cuales tienen la capacidad de integrar aplicaciones con la nube y de esta forma crear entornos inteligentes. Hoy estas tecnologías permiten que cualquier persona pueda acceder a un

servicio de internet desde un computador en casa, un pc en el trabajo o un dispositivo móvil desde cualquier parte del mundo.

Teniendo en cuenta la acogida de IoT, su importancia y crecimiento. En el año 2012, la Unión Internacional de las Telecomunicaciones (UIT), a través de su recomendación UIT-T Y.2060 [13]: Descripción general de Internet de los objetos, definió Internet de los objetos (IoT) como “La infraestructura mundial para la sociedad de la información que propicia la prestación de servicios avanzados mediante la interconexión de objetos (físicos y virtuales) gracias a la interoperabilidad de tecnologías de la información y la comunicación presentes y futuras” y un Objeto como “Un objeto del mundo físico (objetos físicos) o del mundo de la información (objetos virtuales) que se puede identificar e integrar en las redes de comunicaciones”. Más adelante, la UIT en el año 2015, con el fin de acelerar la colaboración mundial para el desarrollo de la IoT, a través del grupo GANT, **creó una nueva Comisión de Estudio del UIT-T**, la Comisión de Estudio 20: Que se basa en IoT y sus aplicaciones, para permitir el desarrollo coordinado de las tecnologías de IoT, incluidos la normalización de arquitecturas de extremo a extremo para la IoT y los mecanismos para la interoperabilidad de las aplicaciones y de los conjuntos de datos de la IoT [13].

En otras palabras, IoT es la conexión entre el mundo real o físico e internet, donde existe comunicación con cualquier objeto, en cualquier instante y en cualquier lugar [13].

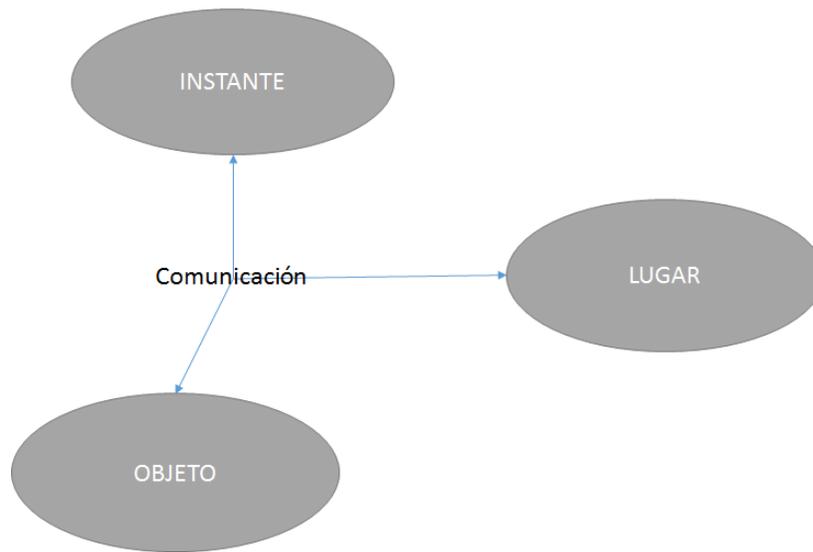


Figura 1. Estructura de IoT por la recomendación Y2060 [13].

Por otro lado, según el modelo de referencia de IoT que presentan la UIT. Internet of Things, consta de dos capacidades, una de gestión y una de seguridad relacionadas con cuatro capas adicionales las cuales son: Capa de aplicación donde se encuentran las aplicaciones IoT, Capa de apoyo a servicios y aplicaciones genéricas y específicas, Capa de red con las capacidades de red y transporte y, por último, la Capa de dispositivo y pasarela [13].



Figura 2. Modelo de Referencia IoT presentado por la UIT en la recomendación Y2060 [13].

El informe presentado en Ginebra el 19 de enero del 2016 entre la UIT y Cisco System: *“Harnessing the Internet of Things for Global Development”* (Aprovechar la Internet de las cosas en pro del desarrollo mundial), se presenta la IoT como una oportunidad de desarrollo mundial que podría mejorar la calidad de vida de millones de personas y alcanzar grandes avances en el desarrollo sostenible. En informe define IoT como el crecimiento de dispositivos capaces de conectarse a internet y que pueden comunicarse con otros dispositivos sin intervención humana [14].

Además, el informe presentan tres enfoques fundamentales para el desarrollo de IoT [15]:
Disponibilidad: Garantiza el acceso a dispositivos IoT, de forma fácil y a bajos costos.

Asequibilidad: Garantiza la disponibilidad de la IoT básica

Adaptabilidad: Garantiza la adaptación de los dispositivos al medio y la interoperabilidad entre ellos.

Teniéndose en cuenta estos tres enfoques, se puede mencionar que según el último estudio realizado por Gartner, Inc., empresa consultora y de investigación de las tecnologías de la información, se estima que para el año 2016, estarán conectados a internet alrededor de 6.4 mil millones de cosas, superando de esta forma en un 30% los dispositivos conectados en el año 2015 [16].

Gartner, también estima que IoT un gasto de servicios total de \$ 235 millones de dólares en 2016. Servicios clasificados como categoría profesional (donde se emplean proveedores externos con el fin de diseñar, instalar y operar Los sistemas de IoT), dejando abierta la posibilidad de crecimiento de los servicios de conectividad y de consumo de forma rápida [16].

Según Jim Tull, vicepresidente de Gartner, los servicios de IoT son el conductor del valor de la IoT, y la atención se está orientando a los nuevos servicios por parte de organizaciones de usuarios finales y proveedores," [16].

Por otro lado, dentro de los estudios que realiza Gartner, Inc., se encuentran los llamados Cuadrantes Mágicos de Gartner, que permiten identificar cuáles son los líderes en los principales mercados de tecnología. Estos cuadros investigan un mercado específico y establece las posiciones de los competidores del mercado, clasificándolos en cuatro categorías: Líderes, visionarios, jugadores de nicho y retadores [17].

En sus más recientes versiones, se encuentran: El Cuadrante Mágico de Gartner para la plataforma de aplicaciones empresariales como servicio, por todo el mundo, publicado en marzo del 2016, el Cuadrante Mágico de Gartner para servicio IaaS, publicado en mayo del 2015 y el Cuadrante Mágico de Gartner para servicios de almacenamiento en la nube pública, publicado en junio del 2015; donde los más destacados del Mercado son las plataformas de Amazon Web Services y Microsoft Azure, como proveedores de servicios en la nube [18].

2.2. Plataformas Cloud de IoT

Amazon Web Services (AWS) y Microsoft Azure son considerados líderes en Cloud, siendo AWS aproximadamente el doble de grande que Microsoft Azure quien es su rival más cercano, y 10 veces más grande que sus otros competidores juntos, aunque Azure ha presentado un rápido desarrollo en los últimos años [19][20].

Pero no son los únicos proveedores de servicios en la nube. Actualmente son muchas las plataformas de cloud computing y muchas de ellas ya están incluyendo en su portafolio de servicios IoT. Dentro de las más conocidas se encuentran las relacionadas en la Tabla 1. En esta tabla se pueden visualizar las características más esenciales de cada una de ellas.

Plataforma	SDK/Lenguaje	Protocolos de Comunicación	Hardware	API	Broker	Servicios
 Amazon Web Services	Embedded C, JavaScript, Arduino Yún.	HTTPS, WebSockets y MQTT	Broadcom, Marvell, Renesas, Texas Instruments, Microchip, Intel, Mediatek, Qualcomm, Seeed, BeagleBoard, Raspberry Pi, Intel.	API RESTful	AWS IoT	Computo. Almacenamiento. Distribución de Contenido. Bases de Datos. Servicios de Aplicaciones. Soporte. Redes. Análisis de Datos.
 Microsoft Azure	.Net y UWP, Java, C, Nodejs	HTTP, AMQP, MQTT y protocolos personalizados.	Intel, Raspberry Pi 2, Freescale, Texas Instruments, Minnow Board, BeagleBoard, Seeed, resin.io	REST API. API Apps.	IoT Hub	Computo. Almacenamiento. Distribución de Contenido. Bases de Datos. Servicios de Aplicaciones. Soporte. Redes. Análisis de Datos.
 CloudMQTT	Heroku Ruby Python Node.js Java	MQTT y WebSockets.	ESP8266 Raspberry Pi Arduino	HTTP API.	Servidor Mosquitto (Broker MQTT)	Conexión de Dispositivos IoT y Análisis de Datos con AWS Kinesis.

	Go .NET NodeMCU PHP					
<p>Google Cloud Platform</p> 	Johnny-Five JavaScript framework, MRAA, EMBD Go library, Arduino- wiring, Firmata.	HTTP	Beaglebone , Raspberry Pi, Intel Edison, Arduino series, Adafruit Feather.	HTTPS REST APIs	Cloud Pub/sub	Computo. almacenamient o y bases de datos. Redes. Manejo de Datos. Análisis de Datos.
<p>Wyliodrin</p> 	C / C + +, Java, Pascal, Shell Script, Perl, PHP, Objective-C, C #, Python, Javascript.	XMPP	Intel Galileo, Beaglebone Black, Intel Edison, Raspberry PI, ZedBoard.	Board API. Semantics .	No	Conexión de Dispositivos. Programación Visual. Visualización de datos. Procesamiento de Datos.
<p>Thingspeak</p> 	Lenguaje M(Matlab), HTML, CSS y JavaScript	HTTP	Arduino, Spark, Raspberry Pi, Electronic Imp	ThingSpea k API	No	Recolección, análisis y manejo de datos y dispositivos.
<p>Carriots</p> 	Java, Python, Ruby y Smalltalk	MQTT	Arduino Raspberry Pi 2, BeagleBoar d, FEZ cerbuino, Cubie Board, TST Gate, Nanode.	REST API	No	Recolección, análisis y manejo de datos y dispositivos,
<p>Eclipse IoT</p> 	Java, OSGi y Lua	MQTT, ETSI SmartM2M, CoAP y Lightweight M2M.	Arduino, Raspberry Pi, ARM mbed.	API REST	Mosquitt o Broker.	Conexión, configuración y Administración

						de Dispositivos desde la nube.
Oracle Cloud 	C/C++, Java, Android, Javascript e iOS.	HTTPS	Arduino, Raspberry Pi.	RESTful API	No	Conexión de dispositivos, Almacenamiento o procesamiento de datos, análisis empresarial, integración.

Tabla 1. Plataformas IoT.

Todos estos proveedores, y los demás que brindar servicios de Cloud, tienen su punto de diferencia que el tipo de solución que brindan a cada cliente según su necesidad y en el flujo de trabajo que ejecutan. Pero, aun así, existen unos factores importantes que visualizan mejor el enfoque que tienen cada una de ellas. Estos factores son: Capacidad de computo, almacenamiento y redes, servicios cloud pública y privada y ahora, servicios dirigidos a IoT, etc. [21].

Para diferenciar cada una de estas plataformas, se evaluaron 6 variables como se muestra en la Tabla 2: IaaS, PaaS, Servicios Cloud, compatibilidad con protocolos IoT, Compatibilidad con Hardware y Documentación, que hace alusión a que tanta información de soporte se encuentra de cada una de ellas.

El valor asignado para la evaluación fue:

4: Cumple todo.

3: Cumple con al menos con 2 características.

2: Cumple al menos 1 características.

1: No cumple con ninguna característica.

Plataforma Variable	Amazon Web Services	Microsoft Azure	CloudMQTT	Wylidrin	Google Cloud Platform	Thingspeak	Carriots	Eclipse IoT	Oracle Cloud
IaaS	4	4	4	1	4	1	1	4	1
PaaS	4	4	2	4	2	4	4	3	3
Servicios Cloud	4	4	3	2	3	2	2	2	3
Compatibilidad con Protocolos IoT	4	4	4	2	3	1	1	4	1
Compatibilidad con Hardware	4	4	4	4	3	3	3	4	3
Documentación	4	3	4	4	2	2	2	2	2
Total	24	23	21	17	17	13	13	19	13

Tabla 2. Análisis Plataformas IoT.

La evaluación de la Tabla 2, comprueba los estudios y supuestos de la penetración que tienen las plataformas Amazon Web Services y Microsoft Azure, dado que se visualiza que, aunque las otras plataformas cumplen con algunas de las variables evaluadas, las más completas son estas dos, llevándose el primer lugar Amazon Web Services.

2.2. Dispositivos IoT.

Eventualmente, así como existen varias plataformas de IoT Cloud, también existen múltiples dispositivos capaces de interactuar con dichas plataformas. El número de dispositivos también es considerable, pero los relacionados en la Tabla 3, son los más conocidos. En la Tabla 3, se muestran las características más importantes de los dispositivos para poder después evaluarlos e identificar cual es el mejor y con más aplicación.

	T1	T2	T3	T4	T5
Nombre	Intel Galileo 	Raspberry PI 2 	Raspberry Pi 3 	Intel Edison 	FRDM-K64F Freedom 
Procesador	Intel Quark SoC X1000	Broadcom BCM2835	Broadcom BCM2837 a 1,2 GHz	Intel Atom	ARM Cortex-M4
Arquitectura	Clase Pentium de Intel	ARM [®] ™ ARM1176	ARM Cortex-A53 de 64 bits y cuatro núcleos	x86	ARM
Frecuencia del Procesador	400 MHz	700MHz	900 MHz	500 MHz	120 MHz
RAM	512 KB SRAM y 256 MB DRAM	512MB SDRAM	1GB LPDDR2	1 GB LPDDR3	256 KB
Memoria Flash	8 MB Flash NOR	Conectada al procesador.	Conectada al procesador.	4 GB de eMMC	1 MB
Red	Ethernet 10/100 Mbps	Ethernet 10/100 Mbps	Ethernet 10/100 Wi-Fi Bluetooth 4.1. Bluetooth (BLE).	Bluetooth 4.0 y Wi-Fi	Ethernet 10/100 Mbps
Puertos USB	3	4	4	2	2
Puerto Serial	1	GPIO14(Tx) y GPIO15(Rx)	GPIO14(Tx) y GPIO15(Rx)	N/A	N/A
Soporte de Video	N/A	HDMI – 1080p	Full HDMI	N/A	N/A

Tabla 3. Dispositivos con conexión a Internet.

De igual forma que con las plataformas cloud IoT, a las tarjetas también se le evalúan 5 variables como se muestra en la Tabla 4: Compatibilidad con plataformas, compatibilidad con librerías, capacidad de almacenamiento externo, conexión a la red y puertos disponibles.

El valor asignado para la evaluación fue:

4: Cumple todo.

3: Cumple con al menos con 2 características.

2: Cumple al menos 1 características.

1: No cumple con ninguna característica.

Variable	Dispositivo	Intel Galileo	Raspberry PI 2	Raspberry PI 3	Intel Edison	FRDM-K64F Freedom
Compatibilidad con Plataformas		4	4	4	4	2
Compatibilidad con Librerías		3	4	4	3	2
Capacidad Almacenamiento Externo		4	4	4	4	3
Conexión a la Red		4	4	4	4	4
Puertos Disponibles		3	4	4	3	2
Total		18	20	20	18	13

Tabla 4. Análisis de Dispositivos.

La Tabla 4, evidencia como las tarjetas Raspberry Pi 2 y Pi 3 son los dispositivos con más aplicación para integración con Cloud e IoT por sus características y desarrollos, y que además cuenta con las información de soporte.

3. ANÁLISIS PLATAFORMAS CLOUD IOT

Amazon Web Services (AWS) y Microsoft Azure son considerados líderes en Cloud, siendo AWS aproximadamente el doble de grande que Microsoft Azure quien es su rival más cercano, y 10 veces más grande que sus otros competidores juntos, aunque Azure ha presentado un rápido desarrollo en los últimos años [19][20].

Estos dos proveedores, y los demás que brindar servicios de Cloud, tienen su punto de diferencia que el tipo de solución que brindan a cada cliente según su necesidad y en el flujo de trabajo que ejecutan. Pero, aun así, existen unos factores importantes que visualizan mejor el

enfoque que tienen cada una de ellas. Estos factores son: Capacidad de computo, almacenamiento y redes, servicios cloud pública y privada y ahora, servicios dirigidos a IoT [21].

3.1 CARACTERÍSTICAS DE AMAZON WEB SERVICES Y MICROSOFT AZURE

Los factores determinan el enfoque de cada una de las plataformas que brindan servicios en cloud [21]. AWS y Azure ofrecen rangos similares en capacidad de computo, almacenamiento y redes, y servicios IoT para control de dispositivos y manejo de datos [22] [23]. Este servicio en ambas plataformas se integra con los demás servicios que estas ya ofrecen con el fin de alcanzar una mejor solución.

3.2 Comparación entre Amazon Web Services IoT y Microsoft Azure IoT.

Para entender de una manera más clara el funcionamiento de IoT en estas dos plataformas, se presenta la Tabla 5:

	Amazon Web Service IoT	Microsoft Azure IoT
Protocolos	Control: HTTPS. Datos: HTTPS, WebSockets y MQTT	HTTP, AMQP, MQTT y protocolos personalizados.
Patrones de Comunicación	Telemetría (Control) y Mando	Telemetría (Control) y Mando
Plataformas de dispositivos certificadas	Broadcom, Marvell, Renesas, Texas Instruments, Microchip, Intel, Mediatek, Qualcomm, Seeed, BeagleBoard, Raspberry Pi, Intel.	Intel, Raspberry Pi, Freescale, Texas Instruments, MinnowBoard, BeagleBoard, Seeed, resin.io
SDK/Lenguaje	Embedded C, JavaScript, Arduino Yún.	.Net y UWP, Java, C, Nodejs
Seguridad	TLS (Mutua autenticación)	TLS (Solo autenticación de servidor)
Autenticación	AWS IAM o AWS Cognito para solicitudes HTTPS y WebSockets. SigV4 y Certificado X.509 para conexiones con HTTP y autenticación basado en certificados para conexiones MQTT.	Por dispositivo con el token de SAS

Tabla 5. Comparación entre AWS IoT y Microsoft Azure IoT [21].

Las dos (2) plataformas integran en su estructura la arquitectura de IoT, empleando protocolos de comunicación de IoT, entre los que se destacan HTTP y MQTT, con patrones de comunicación de telemetría para lograr el control y mando de los dispositivos, ambas se basan en lenguajes de programación tales como Java y C y permiten la conexión de los mismos dispositivos [21].

La comunicación de cada plataforma cifrada a través del protocolo TLS (Transport Layer Security) enviando certificados X.509 para mantener una comunicación segura. La autenticación difiere entre cada plataforma, AWS emplea dos tipos de autenticación que depende del protocolo empleado: AWS IAM e Incognito o SigV4 y Certificado X.509. Microsoft Azure emplea un Token SAS para control de acceso y credenciales que se encarga de dar permiso de acceso a los puntos finales [21].

3.3 Arquitectura de Servicio de AWS IoT

La arquitectura de AWS IoT, está basada en el protocolo MQTT. Con esto las “cosas” informan su estado a través de mensajes enviados a un bróker con referencia a un tópico específico, y quienes reciben los mensajes son los que están suscritos a ese tópico [24].

La estructura AWS IoT, Tiene un SDK para dispositivos con AWS IoT. Es un kit de desarrollo de software que permite la conexión, autenticación e intercambio de mensajes de dispositivos de hardware o aplicaciones móviles, realizando la autenticación y autorización según el protocolo. La puerta de enlace del dispositivo es el punto de comunicación entre los dispositivos conectados y cloud (Motor de reglas, Sombras de dispositivos, Servicios AWS y aplicaciones). Por ultimo según cuenta con un motor de reglas que permite el procesamiento de datos entrantes provenientes de los dispositivos y una sombra de dispositivos que almacena los estados enviados desde los

dispositivos y que a través de alguna aplicación pueden ser modificados, como se muestra en la Figura 3 [24][25].

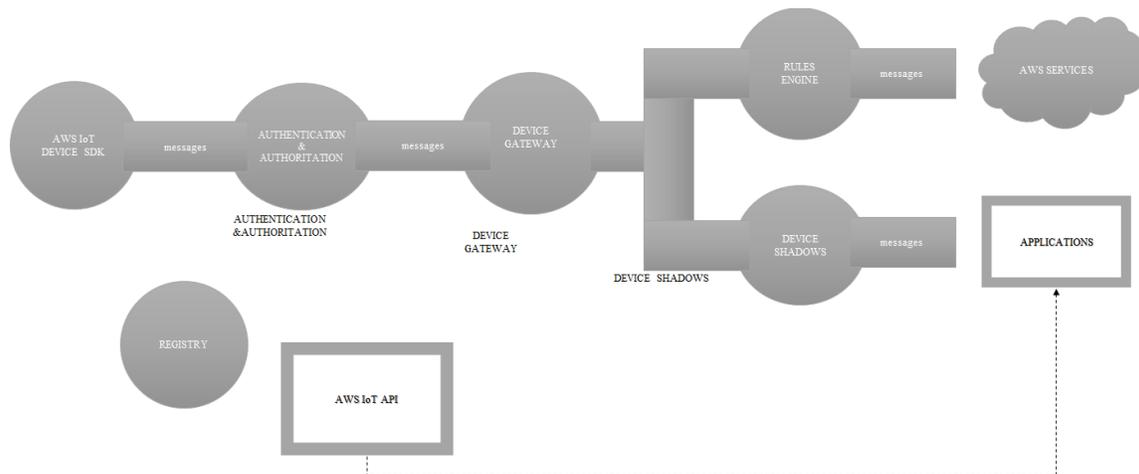


Figura 3. Arquitectura de Servicio de AWS IoT [25].

3.4 Arquitectura de Servicio de Microsoft Azure IoT

La arquitectura que Microsoft presente se divide en tres grupos: Conectividad de Dispositivos, Procesamiento de datos y Análisis y presentación. Dentro de esta estructura, los dispositivos recolectan los datos que se envían a la puerta de enlace que permite que los datos estén disponibles para su procesamiento a través de otros servicios denominados back-end, desde donde se entran los datos a otras aplicaciones a través de un panel de control o un dispositivo de presentación, como se muestra en la Figura 4 [26].

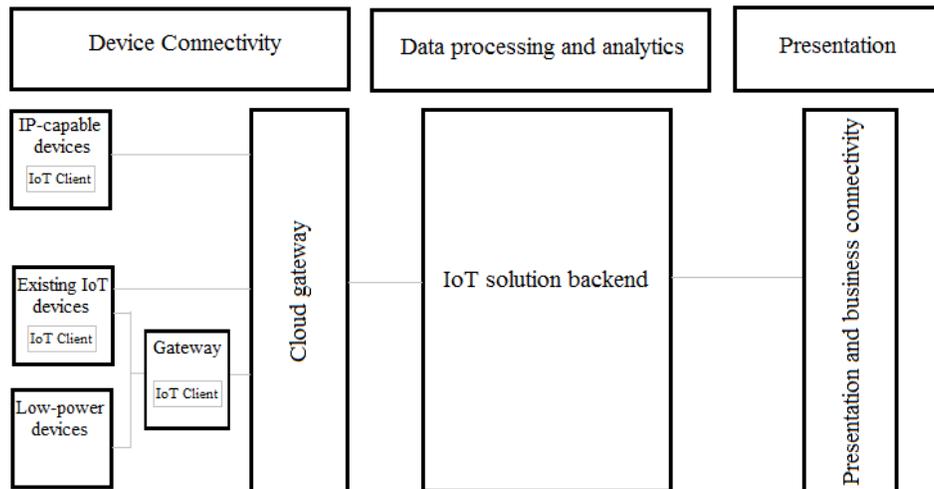


Figura 4. Arquitectura de Servicio de Microsoft Azure [26].

3.5 Servicios y Costos de AWS y Microsoft Azure relacionados con IoT.

Las plataformas se analizan en cuestión de servicios ofrecidos y costos, que pueden, además, integrarse con las soluciones IoT [27 - 29] como se presenta en la Tabla 6 y Tabla 7.

Servicio	AWS	Microsoft Azure
Proceso		
Servidores Virtuales	EC2	Máquinas Virtuales
Lógica de procesos de back-end	Lambda	Functions Trabajos web Aplicaciones lógicas
Almacenamiento y entrega de contenido		
Almacenamiento de objetos	S3	Almacenamiento de blobs
Redes	Virtual Private Cloud	Red Virtual
Base de datos relacional	RDS	Base de datos SQL
Base de datos NoSQL	DynamoDB	DocumentDB
Almacenamiento de tablas	DynamoDB SimpleDB	Almacenamiento de tablas
Análisis y macrodatos		
Análisis	Kinesis Analytics	Análisis de transmisiones
Aprendizaje automático	Aprendizaje automático	Aprendizaje automático
Internet de las Cosas		
Transmisión de datos	Kinesis Firehose Kinesis Streams	Event Hubs
Internet of Things	AWS IoT	IoT Hub
Servicios móviles		

Notificación	Simple Notification Service	Bases de datos centrales de notificaciones
Servicios de aplicación		
Mensajería	Simple Queue Service	Service Bus
Seguridad e identidad		
Directorio	Directorio de Servicio	Directorio Activo Azure

Tabla 6. Paralelo entre servicios de AWS IoT y Microsoft Azure IoT [27].

Un sistema IoT puede integrar múltiples servicios como los relacionados en la Tabla 6, para abarcar la totalidad de una necesidad específica, donde muchas de esas soluciones podrían traer en una sola integrados los servicios tales como: almacenamiento, bases de datos, análisis de datos, control de notificaciones, entre otros servicios [30]. En casos particulares, las plataformas cobran de acuerdo al volumen de información y la capacidad de almacenamiento según sea el caso, pagando por lo que usan. Cuando se trata de envío de mensajes en AWS, se debe tener en cuenta que un mensaje es un bloque de datos de 512 bytes. Es decir, un mensaje de 1024 bytes se facturará como dos mensajes, y los mensajes de tamaño menor a 512 bytes, se cobra como un mensaje, mientras que Microsoft Azure establece para mensajes de dispositivo a la nube un tamaño de 256 KB y de nube a dispositivo de 64 KB[31] [32].

Servicio	Costo AWS	Costo Microsoft Azure
Proceso		
Servidores Virtuales	\$0,018/h	\$0,018/h
Lógica de procesos de back-end	0,00001667 USD por cada GB-segundo utilizado.	No Disponible
Almacenamiento y entrega de contenido		
Almacenamiento de objetos	\$0.0300/GB	\$0.0240/ GB
Redes	\$ 0,05 USD por hora de conexión de VPN. Gateway: \$0,045/h	Conexión Gratuita con suscripción de hasta 50 redes virtuales. Gateway: \$0.036 Gateway/hora
Base de datos relacional	\$0.115/GB mensuales	\$5/GB mensuales.
Base de datos NoSQL	25GB/mes Gratis, a partir de ahí \$0.25 GB/mes	\$ 0,25 GB / mes.

Almacenamiento de tablas	Primer GB gratis, a partir de ahí \$0.250/GB mensual	\$0,07/GB
Análisis y Macrodatos		
Análisis	\$0.015/horas.	\$0,031/horas.
Aprendizaje automático	Análisis de datos y creación de modelos \$0,4/hora.	Seats: \$ 9,99/mes Uso de estudio: \$1,00 seats/mes
Internet de las Cosas		
Transmisión de datos	\$0.035/GB por volumen de datos recibidos.	\$0,028 por millón de eventos. Unidad de rendimiento: \$0,015/hora.
Internet of Things	Dispositivos ilimitados. 5 USD/1 millón de mensaje al mes.	Dispositivos Ilimitados. \$50/ 400000 mensajes al mes.
Servicios móviles		
Notificación	Primer millón de solicitudes gratis, a partir de ahí \$0.50 por millón de solicitudes.	De 10 a 100 millones de inserciones \$1/mes
Servicios de aplicación		
Mensajería	0,50 USD por 1 millón de solicitudes	\$ 0,05 por millón de operaciones
Seguridad e identidad		
Directorio	500 Usuarios \$0,05/ mes.	\$ 1 usuario / mes

Tabla 7. Paralelo de costos entre AWS IoT y Microsoft Azure IoT.

Para la deducción de costos entre las dos plataformas, se hizo necesario establecer unas variables que representen los valores evaluados, así:

Servidores Virtuales: Se evalúa la variable **MR** (Memoria RAM)

EC2 AWS: Ofrece más Memoria RAM (1,07GB), pero el almacenamiento es aparte.

MV Azure: Ofrece menor Memoria RAM (0.75GB), pero incluye 20GB de disco duro.

Lógica de Procesos: Se evalúa la variable **MD** (Memoria asignada a la función por el tiempo de la realización de esa función)

Almacenamiento de Objetos: Se evalúa la variable **CA** (Capacidad de almacenamiento). Para ambos se tiene en cuenta el primer TB/mes.

Redes: Se evalúa la variable **TC** (Tiempo de Conexión) y la variable **G** (Gateway).

Base de datos relacional: Se evalúa la variable **ALG** (Almacenamiento uso General)

Base de datos NoSQL: Se evalúa la variable **AL** (Almacenamiento)

Almacenamiento de tablas: Se evalúa la variable **ALM** (Almacenamiento)

Análisis: Se evalúa la variable **UT** (Unidad de Transmisión).

Los costos relacionados en la Tabla 7, son evaluados para las mismas variables.

Dentro del análisis de costos se pudo evidenciar que AWS cuenta con una capa gratuita de un año para los clientes nuevos que supera en costo/beneficio a Microsoft Azure. Además, los costos que presenta AWS IoT en comparación con IoT también son superados toda vez que para IoT, AWS ofrece de forma gratuita los servicios: Amazon S3 (Para almacenamiento escalable), Amazon DynamoDB (Para bases de datos NoSQL), AWS Lambda (Corre su código en los servidores virtuales de Amazon EC2 en respuesta a los eventos), Amazon Kinesis (Para procesamiento en tiempo real de transmisión de datos a una escala masiva), Amazon SNS (Para el envío o recibo de notificaciones) y Amazon SQS (Para cola de servicio de datos).

3.5.1 Análisis de Costo Caso de Uso AWS

Se desean conectar 20 dispositivo a un concentrador IoT (AWS IoT), cada dispositivo envía un mensaje de 512 bytes por minuto al concentrador. Posteriormente, el concentrador envía cada uno de los para almacenamiento (Amazon S3) y además envía todos los mensajes a otro dispositivo y a una base de datos NoSQL (DynamoDB).

Costo de publicación en AWS IoT: $1 \text{ mensajes} * 60 \text{ minutos} * 24 \text{ horas} * 30 \text{ días} = 43200 \text{ mensajes al mes por dispositivo} * 20 \text{ dispositivos} = 864000 \text{ mensajes al mes}$. Es decir, el costo de AWS IoT es igual a: $864000 * (5 \text{ USD}/1 \text{ millón de mensajes}) = 4,32 \text{ USD al mes por la publicación de mensajes}$.

Costo del envío a S3: No se incurre en ningún costo de AWS IoT por el envío de mensajes. Sin embargo, se sigue incurriendo en los costos estándar de Amazon S3.

Costo del envío a los dispositivos: AWS IoT igual a: 20 dispositivo * 1 mensajes * 60 minutos * 24 horas * 30 días = 864000 millones de mensajes al mes, lo que incurriría en 864000 * (5 USD/1 millón de mensajes) = 4,32 USD al mes por el envío de mensajes en AWS IoT.

Costo del envío a DynamoDB: No se incurre en ningún costo de AWS IoT por el envío de mensajes a DynamoDB. Sin embargo, se sigue incurriendo en los costos estándar de Amazon DynamoDB.

Costo total:

3.5.2 Análisis de Costo Caso de Uso Microsoft Azure.

Se desean conectar 20 dispositivo a un concentrador IoT (IoT Hub), cada dispositivo envía un mensaje de 1024 bytes por minuto al concentrador. Posteriormente, el concentrador envía cada uno de los para almacenamiento (Almacenamiento de Blog) y además envía todos los mensajes a otro dispositivo y a una base de datos NoSQL (DocumentDB).

Costos IoT Hub: mensajes * 60 minutos * 24 horas * 30 días = 43200 mensajes al mes por dispositivo * 20 dispositivos = 864000 mensajes al mes. Por este servicio cobraría \$108 dolares.

Costo Almacenamiento de Blog: Son 864000 mensajes al mes. Eso implica un almacenamiento de 0.823975. Por ese servicio se cobraría. \$0.019754 mensuales

Costo NoSQL: Son 864000 mensajes al mes. Eso implica un almacenamiento de 0.823975. Por ese servicio se cobraría. \$ 0,20599375 dólares mensuales.

3.6 Modelo de Negocio Cloud

Para comprender a detalle la arquitectura de servicios de ambas plataformas es necesario hablar de sus formas de negocio, las cual se estructura en tres modelos [33]:

IaaS (Infraestructura como un servicio): Se cuenta con el equipo, pero la administración la realiza el cliente. Es decir, el proveedor proporciona la máquina, pero la instalación y el resto lo realiza el cliente. (Máquinas virtuales, Servidores, Almacenamiento).

PaaS (Plataforma como servicio): El proveedor brinda todas las herramientas y el cliente no tiene que administrarlas. Es decir, el cliente no instala, no hace programas, la plataforma brinda todo y el cliente lo usa.

SaaS (Software como servicio): Se trata del software que corre sobre la plataforma y el cliente emplea habitualmente. (APIs: CRM, Email, Hangouts, etc.)

Amazon Web Services es conocido principalmente como una plataforma de infraestructura como un servicio, pero muchos de sus servicios ofrecidos, también son comparables con plataforma como un servicio [33]. La comparación entre IaaS y PaaS está en el tipo de servicio ofrecido y ambas plataformas tienen mezcla de ambos [34]. todo esto se resume en la Figura 5.

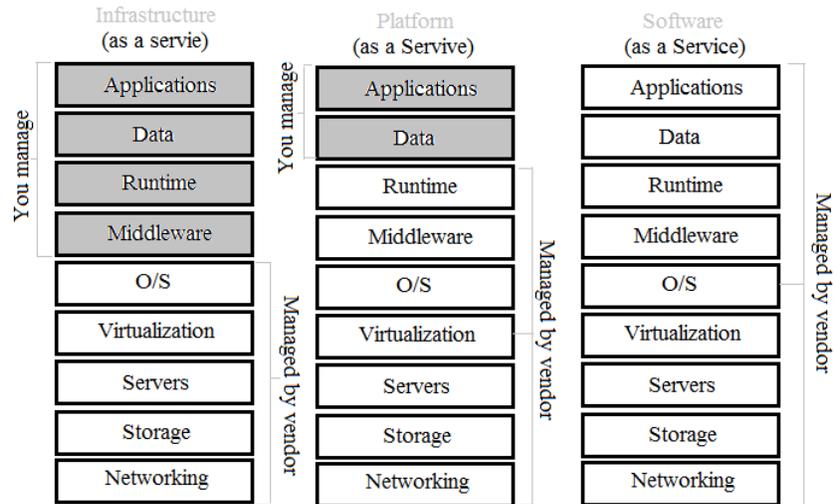


Figura 5. Modelo de Servicio Nube [33].

Ahora, conociendo cuales son los servicios de las plataformas AWS y Microsoft (Tabla 6), los costos de cada una de ellas por servicios (Tabla 7), y los modelos de negocio de plataformas en la nube (Figura 5), se puede proceder a identificar modelos y servicios de negocios más adecuados para la aplicación de la IoT sistemas (Figura 6).

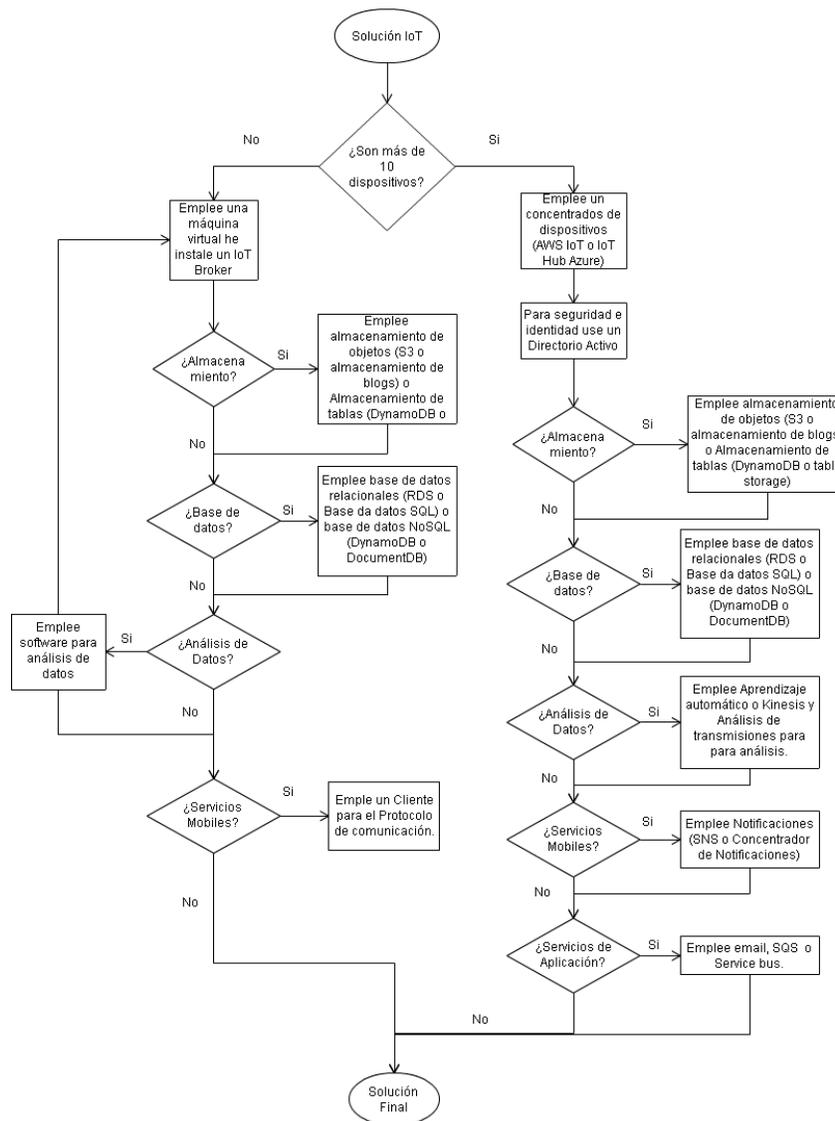


Figura 6. Flujograma modelo de servicios.

El diagrama de flujo, permite visualizar el modelo de negocio para la IO cuando se trata de IaaS y PaaS. Como se puede ver, los distintos modelo de negocio De acuerdo a la necesidad del cliente, de acuerdo con los dispositivos que desea conectar. Por esta razón, no se puede definir cuál de los dos modelos es el mejor. Si el número de los dispositivos que desea conectar es menor de 10, el modelo de negocio más adecuado es el IaaS, donde los recursos

son gestionados por el cliente, pero si el número de dispositivos es mayor que 10, el modelo de negocio más adecuado es el PaaS , donde el cliente utiliza los recursos que ofrece el proveedor de llevarlo a cabo implementaciones y también la gestión de sus aplicaciones.

4. ANÁLISIS PROTOCOLOS IOT

Inicialmente el protocolo oficial adoptado por Microsoft Azure para los servicios de IoT fue AQMP, dado que era el que mejor se adaptaba a sus servicios: Service Bus, topics/subscriptions, Event Hub e IoT Hub, mientras que el protocolo base empleado por AWS IoT es MQTT, aunque ambas plataformas soportan HTTP [21].

Pero este año, Azure incorporo el soporte de forma nativa para MQTT, permitiendo de esta manera omitir la traducción que se hacía de campos MQTT a AMQP para comunicarse con el bróker [35].

De esta forma, no solo estas dos plataformas están compitiendo en oferta, si no que están resaltan la importancia y utilidad del protocolo MQTT en IoT.

Pero estos son los únicos protocolos, además de estos se encuentran: **XMPP** (Mensajería extensible y Protocolo de presencia), **DDS** (Servicio de Distribución de Datos), **JMS** (Servicio de mensajería Java), **CoAP** (Protocolo de aplicación restringida), **REST** (Transferencia de estado representacional), entre otros [36], los cuales se encuentran relacionados y con sus principales características en la Tabla 8.

	MQTT	XMPP	DDS	AMQP	JMS	CoAP	REST
	Message Queue Telemetry Transport	Extensible Messaging and Presence Protocol	Data Distribution Service	Advanced Message Queuing Protocol	Java Message Service	Constrained Application Protocol	Representational State Transfer

Modelo de Comunicación	Publicación /Suscripción y Petición/Respuesta	Publicación /Suscripción y Petición/Respuesta	Publicación /Suscripción	Publicación /Suscripción	Publicación /Suscripción	Petición/Respuesta	Petición/Respuesta
Arquitectura	Cliente/Servidor - cada sensor de cliente conectado a un Broker	Cliente/Servidor empujando Gateway – Roosters	Espacio de datos globales	P2P o Broked	Brokered	Cliente/Servidor – M2M	P2P
Tiempo Real	No	Si	Si	No	No	No	No
Transporte	TCP RFC5246 RFC6455	TCP	UDP y TCP	TCP	TCP	UDP	TCP
QoS	3 Niveles *Fire and forget *delivered at least once *delivered exactly once	3	22	3	3	Solicitudes y mensajes: Confiable o no confiable	Proporcionada por transporte e.g TCP
Estándar	MQTT-v3.1.1. OASIS STANDARD	IETF RFC 6122	OMG's RTPS y DDSI	OASIS AMQP	JCP JMS	IETF: RFC 7252	Es estilo de arquitectura, no un estándar.
Control de suscripciones	Temas con emparejamiento jerárquico	N/A	Temas particiones, con el filtrado de mensajes	Intercambios, las colas y los enlaces en v0.9.1 estándar, definido en el último v1.0	Los temas y las colas con el filtrado de mensajes	Proporciona soporte para direccionamiento multicast	N/A
Serialización de datos	Indefinido	JSON y XML	CDR	Tipo AMQP	Indefinido	Configurable	No
Modelo de Licencia	Abierto y comercialmente con licencia	Abierto, público y sin ningún costo.	Abierto y comercialmente con licencia	Abierto y comercialmente con licencia	Abierto y comercialmente con licencia	Abierto y comercialmente	HTTP disponible de forma

						Con licencia	gratuita en la mayoría de las plataform as
Seguri dad	Autenticaci ón de clientes, cifrado de Informació n con SSL/TLS	TLS/SASL	Proveedor especifico, generalme nte SSL, TLS o control de acceso exclusivo	Autenticaci ón SASL, TLS para encriptaci ón de datos.	Proveedor especifico, generalme nte SSL o TLS usado con JAAS API	DTLS: datagram as de Transport Layer Security	SSL o TLS

Tabla 8. Protocolos IoT [36].

Con las características de cada una de los protocolos, se procese a la evaluación de cada uno de ellos, con el fin de identificar cual protocolo es más viable emplear en una solución de IoT.

Para este caso, las variables evaluadas fueron 5 como se muestra en la Tabla 9: Disponibilidad en Plataforma, disponibilidad en Broker, disponibilidad de clientes, Funcionamiento y calidad de servicio.

El valor asignado para la evaluación fue:

4: Cumple todo.

3: Cumple con al menos con 2 características.

2: Cumple al menos 1 características.

1: No cumple con ninguna característica.

	MQTT Message Queue Telemetr y Transpor)	XMPP Extensibl e Messagin g and Presence Protocol	DDS Data Distributio n Service	AMQP Advance d Message Queuing Protocol	JMS Java Messag e Service	CoAP Constraine d Applicatio n Protocol	REST Representati on al State Transfer
--	---	--	--	---	--	--	---

Disponibilidad en Plataformas	4	2	1	3	1	2	3
Disponibilidad en Broker	4	1	1	2	1	1	1
Disponibilidad de Clientes	4	4	1	3	1	1	1
Funcionamiento	4	3	3	4	3	4	3
Calidad de Servicio	4	4	4	4	4	4	4
Total	20	14	10	16	10	12	12

Tabla 9. Análisis de Protocolos.

El protocolo que Microsoft Azure usa para los servicios de IoT es Advanced Message Queuing Protocol - AMQP, debido a su adaptación a los servicios: Service Bus, topics/subscriptions, Event Hub e IoT Hub. Pero ahora incorporo el soporte de forma nativa para MQTT, omitiendo la traducción de campos MQTT a AMQP en la comunicación con el bróker [35]. El protocolo base para AWS IoT es MQTT, pero las plataformas soportan el Hypertext Transfer Protocol - HTTP.

De la Tabla 9, se puede deducir que el protocolo más empleado y sobre el que más se ha trabajado es en protocolo MQTT, por esta razón se profundiza en ese protocolo.

El protocolo MQTT fue inventado por IBM pero posteriormente fue adoptado por la OASIS, sociedad en estándares abiertos, quienes lo mostraron como versión 3.1.1. Es un protocolo de conectividad para IoT, M2M (Machine-to-machine) que emplea el modelo publicación / suscripción y necesita de un Bróker para administrar y direccionar mensajes entre los nodos de una red [37][38].

--*Modelo Publicación /Suscripción:* El bróker y nodos publican información y otros se suscriben de acuerdo a un tema. Generalmente el bróker está suscrito a todos los mensajes y después gestiona el flujo a los nodos.

--*Desacoplamiento de espacio*: Mientras que el nodo y el bróker necesita tener la dirección IP de uno al otro, los nodos pueden publicar información y suscribirse a la información publicada otros nodos. Esto reduce la sobrecarga que puede acompañar a las sesiones TCP y puertos, y permite que los nodos finales que operan independientemente uno de otro [38].

--*Tiempo de desacoplamiento*: Un nodo puede publicar su información independientemente de los estados de otros nodos. Otros nodos pueden recibir la información publicada por el broker cuando están activos. Esto permite que los nodos permanecen en estados con sueño, incluso cuando otros nodos están publicando mensajes directamente relevantes para ellos [38].

--*Desacoplamiento de sincronización*: Un nodo que está en medio de una operación no se interrumpe para recibir un mensaje publicado a la que está suscrito. El mensaje se pone en cola por el corredor hasta que el nodo receptor está terminado con su operación existente. Se evita así la corriente de funcionamiento y reduce las operaciones repetidas, evitando las interrupciones de las operaciones en curso o estados somnolientos [38].

--*Seguridad de MQTT*, utiliza TCP sin cifrar, pero como TCP usa TLS / SSL de seguridad de Internet, este es un método muy seguro para el cifrado del tráfico.

--*MQTT Calidad de los niveles de servicio (QoS)*: Este protocolo cuenta con tres niveles de servicio: niveles 0, 1, y 2, donde el aumento del nivel garantiza la entrega de mensajes.

Nivel 0 (Dispara y olvida), es una sola ráfaga de transmisión sin la garantía de la llegada del mensaje.

Nivel 1, garantiza que un mensaje es recibido al menos una vez por el destinatario. Cuando el mensaje es recibido y entendido, se responde con un acuse de recibo (PUBACK) dirigido al nodo de publicación.

Nivel 2, este nivel intenta garantizar que el mensaje es recibido y decodificado por el receptor. Es el nivel MQTT más seguro y fiable de calidad de servicio [38].

--*Última voluntad y testamento*: Este protocolo cuenta con un mensaje "(GLN) testamento", que puede ser almacenado en el bróker en caso de que un nodo se desconecte inesperadamente de la red. Este LWT conserva el estado y el propósito del nodo, incluyendo los tipos de comandos que publicó y sus suscripciones, y si el nodo desaparece, el bróker notifica a todos los abonados de LWT del nodo [38].

5. PRUEBAS

Para la realización de las pruebas se tuvieron en cuenta los análisis obtenidos del estudio de las plataformas, los protocolos y los dispositivos de IoT. Esos análisis concluyeron que actualmente, las dos plataformas con más despliegue de IoT con Amazon Web Services y Microsoft Azure, que el protocolo más empleado por las características que tiene es el protocolo MQTT y que los dispositivos que más se adaptan en las integraciones de IoT son las Raspberry Pi2 y Raspberr Pi3.

Por esta razón, se realizaron pruebas sobre ambas plataformas, empleando el protocolo MQTT y la tarjeta Raspberry Pi3.

Las pruebas realizadas sobre AWS constaron de la implementación de un escenario para conectar la tarjeta Raspberry Pi2 a internet y comunicarla con el concentrado de dispositivos de AWS (AWS IoT) y, por último, integrarla con los demás servicios que ofrece AWS para almacenamiento, manejo de datos, análisis de datos, servicios móviles, notificaciones, entre otros.

Antes de comenzar con las pruebas, se debe configurar la Raspberry Pi así:

Para poder emplear la Raspberry Pi2, primero es necesario instalarle el sistema operativo que se desea emplear, en una memoria SD. Los sistemas que se pueden instalar son varios según las necesidades de cada uno. Para este caso, se instala el sistema operativo de Rasbian Jessie así:

- a. Se descarga el archivo con el sistema operativo directamente se la página oficial de Rasbian.
- b. El archivo es un archivo comprimido ZIP, así que se debe descomprimir el archivo en una carpeta.
- c. Al descomprimir el archivo, se observa que este contiene una imagen de instalación, pero esta imagen no se puede almacenar en la SD así, porque no permitiría la lectura de su contenido. Por esta razón, se emplea el programa Win32 Disk Imager, que llama la imagen y lo descomprime en el lugar que se indique, en este caso la tarjeta SD.
- d. Con el sistema operativo en el SD, esta, se inserta en la Raspberry, se conecta a la fuente de poder, a Internet y se enciende.
- e. Después de encendida, esta inicia el sistema operativo y través de SSH desde un computador se realizan las configuraciones iniciales del sistema operativo: País, idioma, configuración a Internet, etc.
- f. Listo, la tarjeta está disponible para empezar a usarse.

5.1 Pruebas sobre la Plataformas Amazon Web Services

5.1.1 Escenario de Prueba

El escenario de pruebas se encuentra ilustrado en la Figura 7. Este consta de una tarjeta Raspberry Pi2 que se conecta vía Ethernet a un Router que permite la salida a Internet para poder conectarse con AWS IoT, y la cual está controlada desde un computador vía SSH, a través del

programa VNC Server, que además de permitir el acceso por SSH, permite la visualización de la interfaz gráfica del sistema operativo de la Raspberry sin emplear otros periféricos. Después, directamente desde AWS se integran los demás servicios.

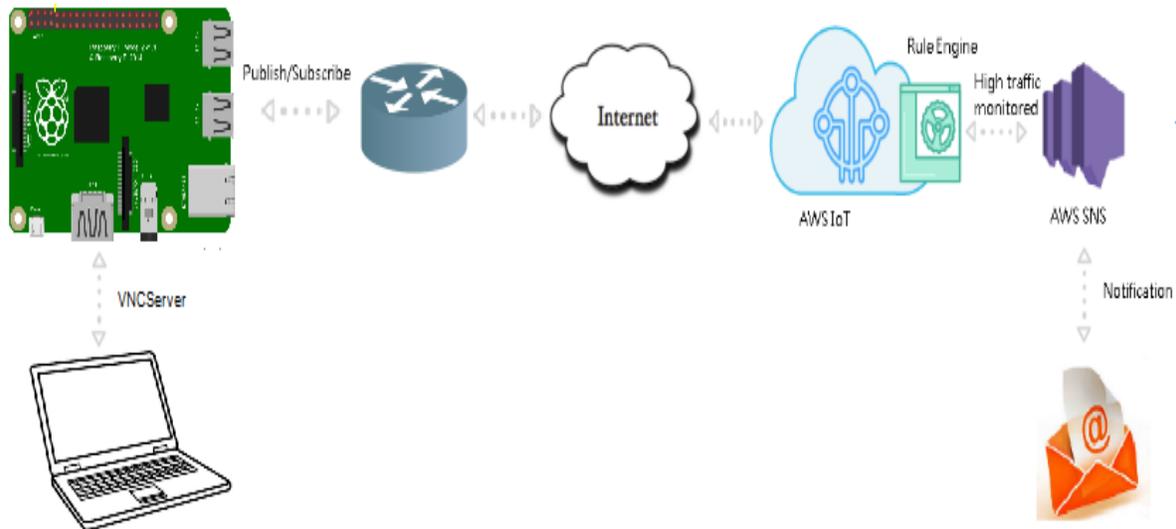


Figura 7. Escenario de Prueba con AWS.

5.1.2 Conexión de la Raspberry Pi a AWS IoT.

El procedimiento para la conexión de la tarjeta al concentrador de dispositivos de AWS, inicia directamente de la consola de configuración de AWS y posteriormente a configuraciones en la tarjeta así:

- a. Se inicia sesión en la consola de AWS IoT en <https://aws.amazon.com/iot>.
- b. En la página de bienvenida, se selecciona Comience con AWS IO y después se selecciona la opción crear una fuente.



Figura 8. Creación de Fuente AWS.

- c. Se crea y se adjunta una cosa (dispositivo), dando click en crear una cosa y asignándole un nombre.

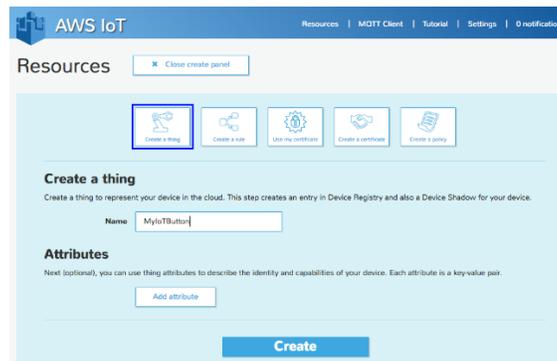


Figura 9. Creación cosa AWS IoT

- d. Después de creada la cosa, esta se visualiza más abajo. Dentro de la cosa creada, se da click en ver cosa para obtener la información sobre este.

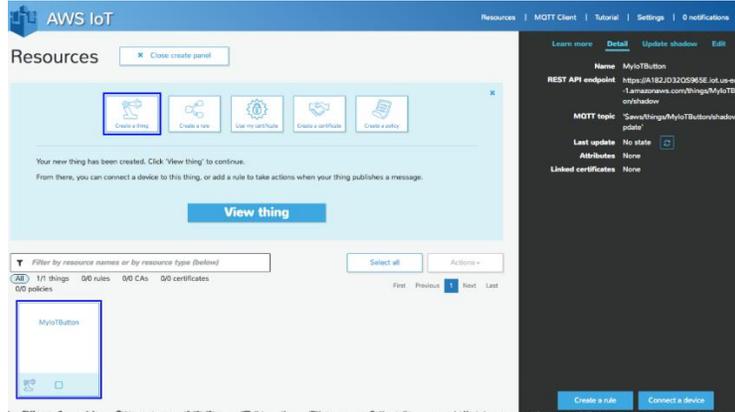


Figura 10. Visualización cosa AWS IoT.

- e. Con la cosa creada, se elige la opción conectar dispositivo dentro del menú que se encuentra dentro de la información de la cosa.

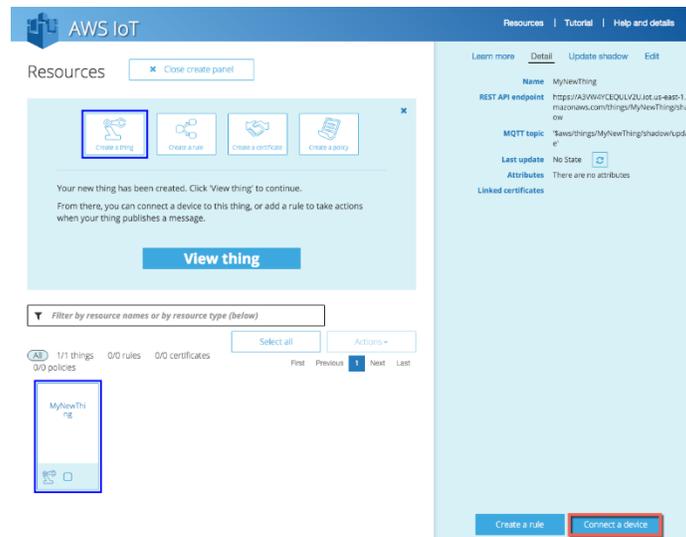


Figura 11. Conexión de dispositivo a AWS IoT.

- f. Se elige el SDK que se desea de usar, en este caso Embedded C, y se selecciona la opción Generar certificado y la política, los cuales crea las claves y un certificado para la conexión con AWS IoT.

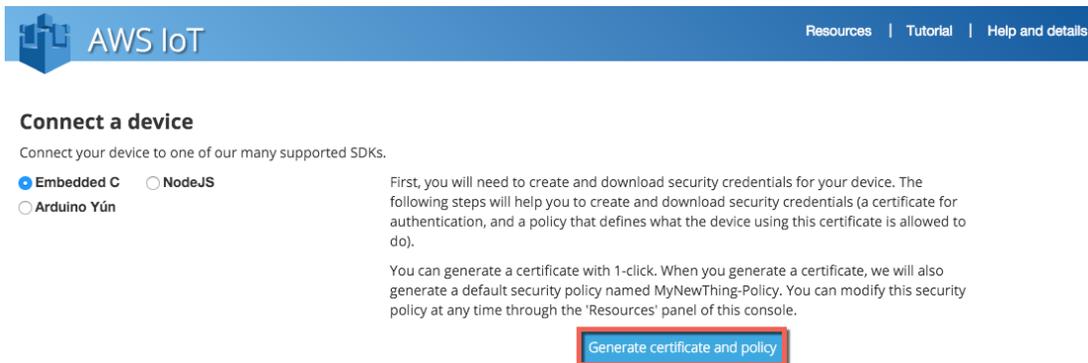


Figura 12. Creación de certificados AWS IoT.

- g. La creación de políticas, generará un certificado X.509 y unas claves pública y privada, que se deben almacenar en una carpeta dentro de la Raspberry Pi2. La carpeta en este caso se llamó deviceSDK. La descarga de estos archivos se realiza desde el PC, pero posteriormente a través de una USB se guardan en la tarjeta. Además de eso, se necesita un certificado, que se debe copiar desde la misma página de AWS y almacenar en la misma carpeta donde se almacenaron los anterior.

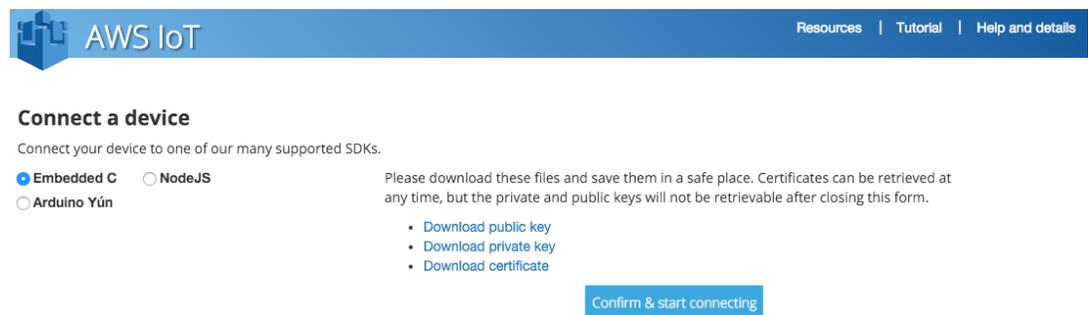


Figura 13. Certificados y claves AWS IoT.

- h. Se elige la opción confirmar y comenzar a conectar.

Connect a device

Connect your device to one of our many supported SDKs.

- Embedded C NodeJS
 Arduino Yún

AWS IoT C SDK

Download one of the AWS IoT C SDKs:

- [OpenSSL](#)
- [mbed-TLS](#)

Set up the SDK using the instructions in our [README](#) on GitHub.

Add in the following sample code based on your account, Thing, and new certificate:

```
// Get from console
// =====
#define AWS_IOT_MQTT_HOST      "A3VW4YCEQULV2U.iot.us-east-1.amaz
onaws.com"
#define AWS_IOT_MQTT_PORT      8883
#define AWS_IOT_MQTT_CLIENT_ID "MyNewThing"
#define AWS_IOT_MY_THING_NAME  "MyNewThing"
#define AWS_IOT_ROOT_CA_FILENAME "root-CA.crt"
#define AWS_IOT_CERTIFICATE_FILENAME "5f2934908b-certificate.pem.crt"
#define AWS_IOT_PRIVATE_KEY_FILENAME "5f2934908b-private.pem.key"
// =====
```

Start one of the sample applications found in the SDK. You can use the AWS IoT console to observe the state of your thing's shadow and interact with your device by updating the shadow. Only one device can use a clientID for connecting to the AWS IoT platform at the same time. If you want to connect multiple devices concurrently please create a separate thing (and client certificate) per device that you intend to connect.

[Return to Thing Detail](#)

Figura 14. Conexión SDK AWS IoT.

- Cuando se da click en iniciar a conectar, AWS muestra las dos versiones de SDK que hay para el lenguaje C. En este caso se elige la versión OpenSSL. Al elegir esta opción, se descarga el SDK de dispositivos AWS IoT en favor de C en un archivo comprimido (`linux_mqtt_openssl-latest.tar`). Este archivo también se guarda en la carpeta creada en la tarjeta `deviceSDK`.
- Se descomprime el archivo `.tar` en la tarjeta desde la terminal de la tarjeta, ingresando primero a la dirección del directorio `deviceSDK` y ejecutando el comando: ``Tar -xvf linux_mqtt_openssl-latest.tar``.
- El SDK ya se encuentra instalado en la tarjeta, ahora es necesario instalar la biblioteca OpenSSL en Raspberry Pi. Para esto, nuevamente desde la terminal, se ejecute el comando: `sudo apt-get install libssl-dev`.

- I. Se ejecuta el ejemplo de SDK de C: `subscribe_publish_sample`. Antes de la ejecución, se debe copiar los certificados y las claves en el directorio `certs` que creó el SDK, junto con el directorio de raíz descargado desde AWS, y además configurar desde el archivo, el punto final REST API, que aparece en la descripción de la cosa.

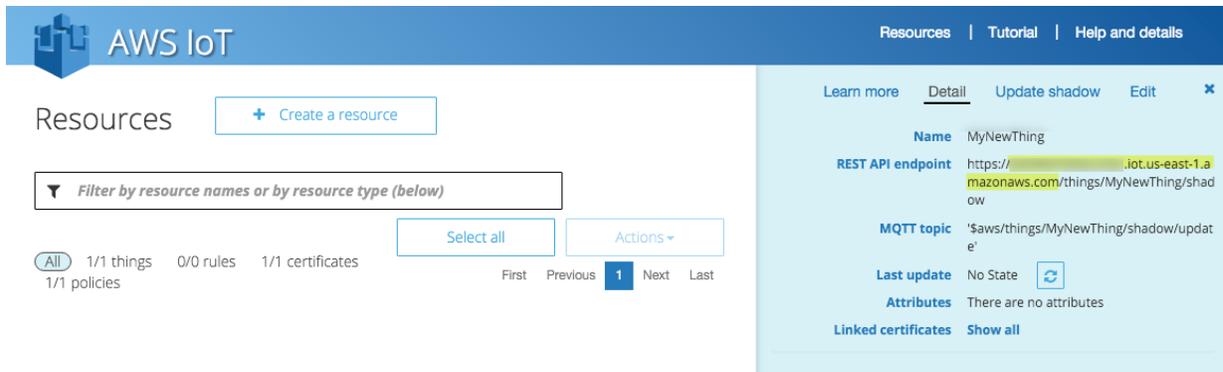


Figura 15. Punto final REST API AWS IoT.

- m. Se abre el archivo de configuración: `aws_iot_config.h` y se modifican los campos:
`AWS_IOT_MQTT_HOST`, `AWS_IOT_MY_THING_NAME`,
`AWS_IOT_ROOT_CA_FILENAME`, `AWS_IOT_CERTIFICATE_FILENAME`, `AWS_IOT_PRIVATE_KEY_FILENAME`, con la información necesaria y los nombres de los certificados y las claves.
- n. Se compila el `subscribe_publish_sample_app` utilizando el comando `makefile`. Este es el que genera el archivo ejecutable para correr el ejemplo.

```
pi@raspberrypi: ~/Downloads/linux_mqtt_openssl-1.0.1/sample_apps/subscribe_publish_sample
File Edit Tabs Help
pi@raspberrypi ~/Downloads/linux_mqtt_openssl-1.0.1/sample_apps/subscribe_publish_sample
h_sample $ ls
aws_iot_config.h Makefile subscribe_publish_sample.c
pi@raspberrypi ~/Downloads/linux_mqtt_openssl-1.0.1/sample_apps/subscribe_publish_sample
h_sample $ make
pi@raspberrypi ~/Downloads/linux_mqtt_openssl-1.0.1/sample_apps/subscribe_publish_sample
h_sample $ ls
aws_iot_config.h subscribe_publish_sample
Makefile subscribe_publish_sample.c
pi@raspberrypi ~/Downloads/linux_mqtt_openssl-1.0.1/sample_apps/subscribe_publish_sample
h_sample $
```

Figura 16. Compilación ejemplo AWS SDK C.

Con esto, termina el procedimiento para la conexión de la tarjeta al AWS IoT, cuando se ejecuta el ejemplo la tarjeta empieza a enviar mensajes a AWS IoT. En este caso el mensaje enviado al concentrador fue HOLA MUNDO.

```
pi's X desktop (raspberrypi:1): VNC Viewer
pi@raspberrypi: ~/Desktop...
pi@raspberrypi: ~/Desktop/deviceSDK/sample_apps/subscribe_publish_sample
File Edit Tabs Help
sdkTest/sub HOLA MUNDO : 26
Subscribe callback
sdkTest/sub HOLA MUNDO : 27
-->sleep
Subscribe callback
sdkTest/sub HOLA MUNDO : 28
-->sleep
Subscribe callback
sdkTest/sub HOLA MUNDO : 29
-->sleep
Subscribe callback
sdkTest/sub HOLA MUNDO : 30
-->sleep
Subscribe callback
sdkTest/sub HOLA MUNDO : 31
-->sleep
Subscribe callback
sdkTest/sub HOLA MUNDO : 32
-->sleep
Subscribe callback
sdkTest/sub HOLA MUNDO : 33
Subscribe callback
sdkTest/sub HOLA MUNDO : 34
-->sleep
Subscribe callback
sdkTest/sub HOLA MUNDO : 35
-->sleep
Subscribe callback
sdkTest/sub HOLA MUNDO : 36
-->sleep
```

Figura 17. Conexión Raspberry Pi2 con AWS.

5.1.3 Empleo de Reglas para conexión a AWS IoT

Para este caso se creó una regla para notificación SNS, así:

- a. Se ingresa a la consola de AWS IoT para SNS
- b. Se crea un nuevo Tema.
- c. Se realizan las configuraciones del tema.
- d. Se crea na regla desde la Consola de AWS IoT.
- e. Se configura la regla con los datos de nombre, descripción, atributo, etc.
- f. En la opción objetivo SNS, se llama el tema creado anteriormente.
- g. Se crea una nueva función, que es lo que se va a ejecutar.
- h. Se crea la acción y se ejecuta la regla.

5.1.4 Cliente MQTT.

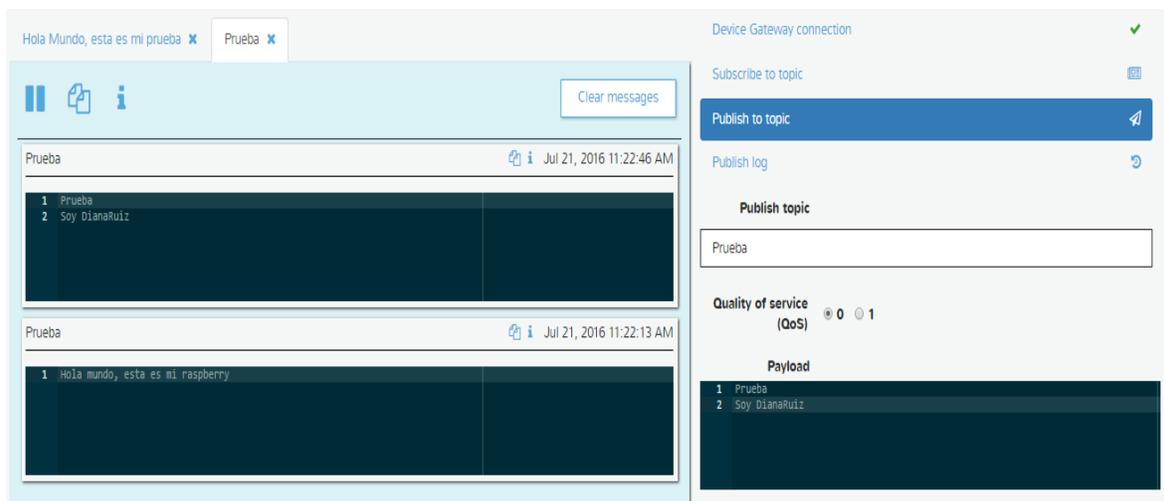


Figura 18. Conexión a Cliente MQTT AWS.

5.1.5 Costos sobre AWS y Proyección de costos.

Detalles		Total
Cargos por servicio de AWS		\$0.15
▼ Data Transfer		\$0.00
US East (Northern Virginia) Region		Uso
Bandwidth		
\$0.00 per GB - data transfer out under the monthly global free tier	0.000003 GB	\$0.00
Total:		\$0.00
Total de la región:		\$0.00
▼ DynamoDB		\$0.00
US East (Northern Virginia) Region		Uso
Amazon DynamoDB		
\$0.00 per hour for 25 units of read capacity for a month (free tier)	3,605 ReadCapacityUnit-Hrs	\$0.00
\$0.00 per hour for 25 units of write capacity for a month (free tier)	3,605 WriteCapacityUnit-Hrs	\$0.00
Total:		\$0.00
Total de la región:		\$0.00
▼ Elastic Compute Cloud		\$0.00
US East (Northern Virginia) Region		Uso
Amazon CloudWatch		
\$0.00 per alarm-month - first 10 alarms	0.971 Alarms	\$0.00
Total:		\$0.00
Total de la región:		\$0.00
▼ IoT		\$0.15
US East (Northern Virginia) Region		Uso
AWS IoT USE1-AWSIoT-MQTT		
\$5 per million messages	28,746 Messages	\$0.14
First 250k messages are free	250,000 Messages	\$0.00
Total:		\$0.14
AWS IoT USE1-AWSIoT-WEB_SOCKET-MQTT		
\$5 per million messages	278 messages	\$0.01
Total:		\$0.01
Total de la región:		\$0.15
▼ Simple Notification Service		\$0.00
US East (Northern Virginia) Region		Uso
Amazon Simple Notification Service DeliveryAttempts-SMTP		
First 1,000 Amazon SNS Email/Email-JSON Notifications per month are free	2 Notifications	\$0.00
Total:		\$0.00
Amazon Simple Notification Service Requests-Tier1		
First 1,000,000 Amazon SNS API Requests per month are free	23 Requests	\$0.00
Total:		\$0.00
Total de la región:		\$0.00

Figura 19. Costos por Servicios usados AWS.

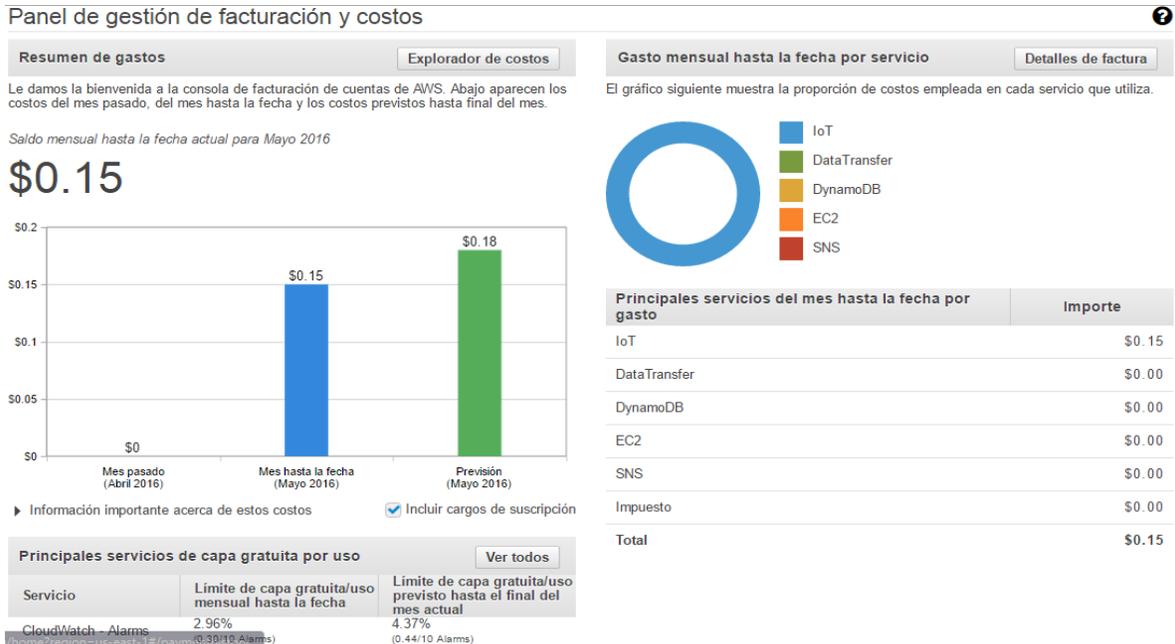


Figura 20. Predicción de Costos AWS.

5.2 Pruebas sobre la Plataforma Microsoft Azure

5.2.1 Escenario de Prueba

El escenario de pruebas se encuentra ilustrado en la Figura 21. Este consta de una tarjeta Raspberry Pi2 que se conecta vía Ethernet a un Router que permite la salida a Internet para poder conectarse con el IoT Hub de Microsoft Azure, y la cual está controlada desde un computador vía SSH, a través del programa VNC Server, que además de permitir el acceso por SSH, permite la visualización de la interfaz gráfica del sistema operativo de la Raspberry sin emplear otros periféricos. Después, directamente desde AWS se integran los demás servicios.

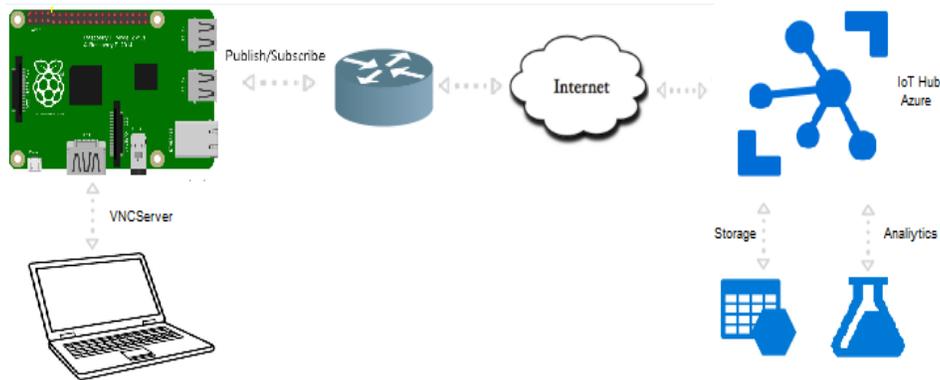


Figura 21. Escenario de prueba con Microsoft Azure.

5.2.2 Conexión a IoT Hub

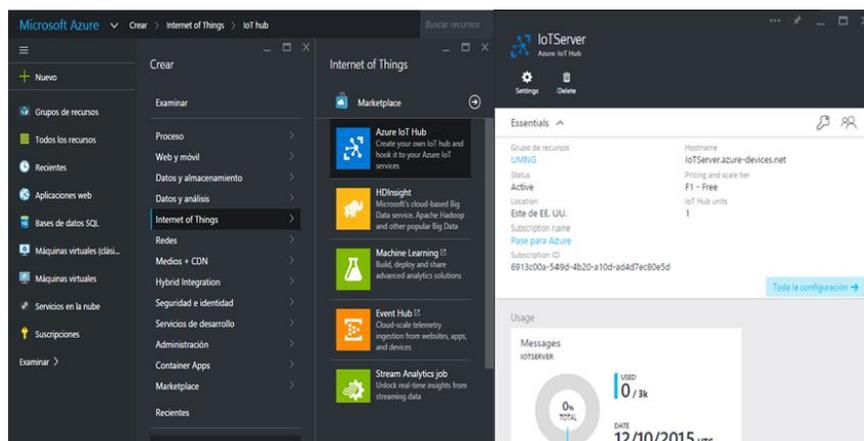


Figura 22. Creación IoT Hub Azure.

5.2.3 Creación servicios IoT.

NOMBRE	TIPO	GRUPO DE RECURSOS	UBICACIÓN	SUSCRIPCIÓN
 DianaMRuiz	IoT Hub	DianaMRuiz	Oeste de EE. UU.	Pase para Azure
 dianamruiz	Cuenta de Documen...	DianaMRuiz	Oeste de EE. UU.	Pase para Azure
 dianamruiz	Cuenta de almacena...	DianaMRuiz	Oeste de EE. UU.	Pase para Azure
 DianaMRuiz	Servicio de aplicacio...	DianaMRuiz	Oeste de EE. UU.	Pase para Azure
 DianaMRuiz-DevicelInfo	Trabajo de Análisis d...	DianaMRuiz	Oeste de EE. UU.	Pase para Azure
 DianaMRuiz-jobhost	Servicio de aplicacio...	DianaMRuiz	Oeste de EE. UU.	Pase para Azure
 DianaMRuiz-jobsplan	Plan del servicio de ...	DianaMRuiz	Oeste de EE. UU.	Pase para Azure
 DianaMRuiz-map	Bing Maps API for E...	DianaMRuiz	Oeste de EE. UU.	Pase para Azure
 DianaMRuiz-plan	Plan del servicio de ...	DianaMRuiz	Oeste de EE. UU.	Pase para Azure
 DianaMRuiz-Rules	Trabajo de Análisis d...	DianaMRuiz	Oeste de EE. UU.	Pase para Azure
 DianaMRuiz-Telemetry	Trabajo de Análisis d...	DianaMRuiz	Oeste de EE. UU.	Pase para Azure

Figura 23. Servicios implementados IoT Suite Azure.

5.3 Prueba Piloto

Pausa remota de Netflix con la Raspberry Pi a través de AWS IOT

Crear un recurso con la opción: "Crear una cosa"

Conectar un dispositivo.

Seleccionar "NodeJS" y seleccionar "Generar certificado y la política".

Conectar la Raspberry Pi al servicio AWS IOT

5.3. 1Conecta la Raspberry Pi a AWS IOT

a. Instalar el AWS CLI con el comando: sudo PIP instalas awscli.

b. Se abre la consola de IAM.

c. En el panel de navegación, se selecciona Usuarios.

- d. Se selecciona el nombre de usuario de IAM.
- e. se elige la ficha credencial de seguridad y luego seleccionar Crear clave de acceso.
- f. Seleccionar mostrar las credenciales de seguridad del usuario. Clave de acceso ID y Clave Secreta de acceso.
- g. Elegir credenciales de descarga, y almacenar las claves.

5.3.2 Ejecutar sudo AWS configure

- a. Rellenar credenciales
- b. instalar NodeJS, empleando el comandos: `sudo dpkg -i node_latest_armhf.deb`
`wget http://node-arm.herokuapp.com/node_latest_armhf.deb`
- c. Se instala el paquete de AWS mediante la ejecución de la IoT: `npm install aws-iot-device-sdk`
- d. Se instala el paquete shelljs con el comando: `npm install shelljs`
- e. Se configura el botón mediante los códigos de configuración, y estos se almacenan en la tarjeta Raspberry Pi2.

5.3.3 Configuración el Computador

- a. Se instala en el equipo el programado de Python
- b. Se instala la librería pip y después el cliente de aws con el comando: `sudo pip install awscli`
- c. Se descarga e instala el nodejs
- d. Se instala el paquete de AWS mediante el comando: `npm install aws-iot-device-sdk`
- e. Instalar los paquetes nmps con los comandos: `Npm install shelljs` y `npm install robotjs`
- f. Se ejecuta el programa que realiza la conexión con netflix.



Figura 24. Prueba Piloto a AWS.

6. CONCLUSIONES

Las plataformas tienen el mismo propósito al seguir la arquitectura IoT, la gran diferencia se encuentra en cómo cada una de ellas llega al cliente, es decir, la forma en que cada una ofrece sus servicios al cliente para que este pueda generar mayor entendimiento y aprovechamiento de los recursos y por supuesto, los costos.

Para IoT se pueden usar cualquiera de los modelos de negocio IaaS, PaaS o SaaS, dejando a modo de decisión del cliente la elección de algún modelo de acuerdo a las necesidades y al conocimiento y manejo de los recursos necesarios en una implementación.

Es el protocolo MQTT, el más usado para IoT, porque es un protocolo abierto, fácil de entender, funciona en redes inalámbricas y de ancho de banda bajo, presenta menor tiempo de respuesta que los otros protocolos y requiere de pocos recursos.

Con el fin de implementar soluciones de la IO, es posible emplear plataformas de computación en la nube, como Amazon Web Services o Microsoft Azure. Dentro de plataformas en la nube, también es posible implementar soluciones de infraestructura como servicio, plataforma como servicio o software como servicio. La decisión sobre cuál de esta se acerca a utilizar depende de los costos, ubicación geográfica o incluso parámetros de calidad de servicio. En este trabajo se muestra un modelo de decisión de acuerdo a las características relevantes para la IO implementaciones más de los sistemas de computación en nube.

REFERENCIAS

- [1] Peter Mell & Tim Grance. (2009). The NIST Definition of Cloud Computing. Versión 15. Obtenido de: <http://www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf>
- [2] Telecomunicación Standardization Sector of ITU. (2012). Series Y: Global Information Infrastructure, Internet Protocol Aspects and Next-Generation Networks. Recommendation ITU-T Y.2060.
- [3] Landa Ibon. Qué es Windows Azure. Obtenido de: <http://www.estoyenlanube.com/recursos/windows-azure/que-es-windows-azure/>
- [4] Damaggio Elio. (2015). What is Azure IoT Hub?. Obtenido de Microsoft Azure: <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-what-is-iot-hub/>
- [5] Amazon Web Services. Características de AWS IoT. Obtenido en: <https://aws.amazon.com/es/iot/how-it-works/>

[6] Intel, C. (2015). *Intel Corporation*. Obtenido de Transform Business with Intelligent Gateway Solutions for IoT: <http://www.intel.com/content/www/us/en/internet-of-things/gateway-solutions.html#>

[7] *Vangie Beal*. SDK - software development kit. Obtenido de: <http://www.webopedia.com/TERM/S/SDK.html>

[8] Bravent IT consulting Company. (2015). Migrar a la nube ¿Azure o AWS?. Obtenido de: <http://www.bravent.net/migrar-a-la-nube-azure-o-aws/>

[9] Arshdeep Bahga & Vijay Madiseti. (2014). *Internet of Things: A Hands-On Approach*. Obtenido de: <https://books.google.com.co/books?id=JPKGBAAAQBAJ&pg=PA28&dq=MQTT,+CoaP,+AMQP,+XMPP,+DDs+protocols&hl=es-419&sa=X&ved=0ahUKewiN1pWSrpPOAhXFNx4KHcRXBzAQ6AEIHDA#v=onepage&q=MQTT%2C%20CoaP%2C%20AMQP%2C%20XMPP%2C%20DDs%20protocols&f=false>

[10] D. Evans. (Abril 2011). *The Internet of Things La próxima evolución de Internet lo está cambiando todo*. Artículo Técnico. Obtenido de: http://www.cisco.com/c/dam/global/es_es/assets/executives/pdf/Inter

[11] James Manyika, M. C. (22 de Julio de 2015). McKinsey Global Institute. Obtenido de http://www.mckinsey.com/insights/mgi/in_the_news/by_2025_internet_of_things_applications_could_have_11_trillion_impact

[12] K. Chandrasekaran. (2012). *Essentials of Cloud Computing*. CRC Press. Taylor & Francis Group.

[13] Telecommunication Standardization Sector of ITU. (Junio 2012). Series Y: Global Information Infrastructure, Internet Protocol Aspects and Next-Generation Networks. Recommendation ITU-T Y.2060.

[14] International Telecommunication Union. (2016). Harnessing the Internet of Things for Global Development. Geneva. Obtenido de: <https://www.itu.int/en/action/broadband/Documents/Harnessing-IoT-Global-Development.pdf>.

[15] Sala de Prensa UIT. Internet de las cosas podría ser la “clave de conectividad” de bajo coste que transforme las vidas en los países en desarrollo. Obtenido de: http://www.itu.int/net/pressoffice/press_releases/2016/02-es.aspx#.V4udm_nhDIW

[16] Newsroom Gartner. (2015). Gartner Says 6.4 Billion Connected “Thing” Will Be in Use in 2016, Up 30 Percent from 2015. Stamford, Conn. Obtenido de: <http://www.gartner.com/newsroom/id/3165317>

[17]Gartner. Gartner Magic Quadrant. Obtenido de: http://www.gartner.com/technology/research/methodologies/research_mq.jsp.

[18] Microsoft. (Julio 2016). Follow an industry leader to the cloud. Microsoft Azure. Obtenido de: <https://azure.microsoft.com/en-us/campaigns/magic-quadrant/>.

[19] Finnegan Matthew. (2016). Microsoft Azure vs Amazon AWS public cloud comparison: Which cloud is best for the Enterprise? Obtenido de: <http://www.computerworlduk.com/it-vendors/microsoft-azure-vs-amazon-aws-public-cloud-comparison-which-cloud-is-best-for-enterprise-3624848/>

[20] Butler Brandon. (2015). The relative size of Amazon Web Service's cloud compared to its competitors remains staggering. Obtenido de: <http://www.networkworld.com/article/2925186/cloud-computing/gartner-amazon-s-cloud-is-10x-bigger-than-its-next-14-competitors-combined.html>

[21] Patierno Paolo. An IoT Platforms Match: Microsoft Azure IoT vs Amazon AWS IoT. Blog Devexpetience. Obtenido de: <https://paolopatierno.wordpress.com/2015/10/13/an-iot-platforms-match-microsoft-azure-iot-vs-amazon-aws-iot/>

[22] Amazon Web Services. (2016). AWS capa gratuita. Obtenido de: https://aws.amazon.com/free/?nc1=h_ls

[23] Microsoft Azure. (2016). Azure prueba gratis. Obtenido de: <https://azure.microsoft.com/en-us/offers/ms-azr-0044p/>

[24] Amazon Web Services. Preguntas frecuentes acerca de AWS IoT. Obtenido de: <https://aws.amazon.com/es/iot/faqs/>

[25] Amazon Web Services. Cómo funciona la plataforma AWS IoT. Disponible en: <https://aws.amazon.com/es/iot/how-it-works/>

[26] Betts Dominic. (2016). "Azure and Internet de las cosas". Obtenido de <https://azure.microsoft.com/enus/documentation/articles/iot-suite-what-is-azure-iot/>

[27] Microsoft Azure. (2016). Microsoft Azure and Amazon Web Services. Obtenido en: <https://azure.microsoft.com/en-us/campaigns/azure-vsaws/mapping/>

[28] Amazon Web Services.(2016). AWS Cloud Pricing. Obtenido en: https://aws.amazon.com/pricing/?nc1=h_ls

[29] Microsoft Azure.(2016).En línea. Obtenido en:
<https://azure.microsoft.com/es-es/pricing/calculator/>

[30] Centros de Investigación. (2016). Obtenido de:
<http://www.researchhubs.com/post/computing/cloudcomputing/what-is-cloud-computing.html>

[31] Amazon Web Services.(2016). AWS IoT Pricing. Obtenido de:
https://aws.amazon.com/iot/pricing/?nc1=h_ls

[32] Microsoft Azure. (2016). Obtenido en: <https://azure.microsoft.com/es-es/pricing/details/iot-hub/>

[33] appenda. IaaS, PaaS, SaaS (Explained and Compared). Obtenido en:
<https://appenda.com/library/paas/iaas-paas-saasexplained-compared/>

[34] Sullivan Dan. (2015). PaaS Provider Comparison Guide: Amazon AWS as a PaaS. Obtenido en: <http://www.tomsitpro.com/articles/amazon-aws-paas-iaas-cloudcomputing,2-608.html>

[35] Foster Andrew. (2013). Messaging Technologies for the Industrial Internet and the Internet of Things. Obtenido de:
http://www.prismtech.com/sites/default/files/documents/MessagingComparsionNov2013USROW_vfinal.pdf

[36] PRISMTECH. (2013). Messaging Technologies for the Industrial Internet and the Internet of Things. A Comparison Between DDS, AMQP, MQTT, JMS, REST and CoAP. Obtenido en: http://www.prismtech.com/sites/default/files/documents/MessagingComparsionNov2013USROW_vfinal.pdf

[37] Stansberry James. (2015). MQTT and CoAP: Underlying Protocols for the IoT. Obtenido en:
<http://electronicdesign.com/iot/mqtt-and-coap-underlyingprotocols-iot>

[38] Amazon Web Services. (2016). AWS Free Tier. Obtenido en:
https://aws.amazon.com/free/?nc1=h_ls