

**Proyecto Final:**  
**MEDIDOR DE CONSUMO**  
**DE AGUA REMOTO**  
**PARA HOGARES**

**Diego Eduardo Pineda Torres**  
Código 20171195004

**Prof. José Nelson Pérez**

Aplicaciones sobre Internet y la Nube  
Universidad Distrital Francisco José de Caldas  
Bogotá, Colombia  
Junio de 2018

# Índice

<b>1. INTRODUCCION</b>	<b>2</b>
<b>2. OBJETIVO GENERAL</b>	<b>2</b>
<b>3. COMPONENTES DEL SISTEMA IoT</b>	<b>3</b>
3.1. Sensor de flujo de líquido YF-S201 . . . . .	3
3.1.1. Efecto Hall . . . . .	4
3.2. Raspberry Pi . . . . .	5
3.2.1. Raspberry Pi 3 Modelo B . . . . .	5
3.3. AWS IoT Core . . . . .	6
3.4. AWS Elastic Search Service . . . . .	6
3.5. Protocolo MQTT . . . . .	7
<b>4. ARQUITECTURA DEL SISTEMA IoT</b>	<b>8</b>
<b>5. CONFIGURACION DEL SISTEMA IoT</b>	<b>9</b>
5.1. Configuración de Raspberry Pi . . . . .	9
5.1.1. Actualización de firmware y sistema operativo . . . . .	9
5.1.2. Instalación de Python . . . . .	9
5.1.3. Instalación de openssl . . . . .	10
5.1.4. Instalación de AWS IoT SDK . . . . .	10
5.1.5. Instalación de la librería de GPIO . . . . .	10
5.2. Configuración de AWS IoT Core . . . . .	11
5.2.1. Registro de dispositivo en AWS IoT . . . . .	12
5.3. Configuración de AWS Elastic Search (ES) . . . . .	19
5.4. Configuración de regla de operación en AWS IoT . . . . .	22
5.5. Programa de ejecución en Raspberry Pi . . . . .	26
<b>6. RESULTADOS</b>	<b>30</b>
<b>7. CONCLUSIONES</b>	<b>31</b>
<b>Bibliografía</b>	<b>31</b>

## **1. INTRODUCCION**

Una de las principales aplicaciones de los sistemas de IoT es el monitoreo de los diferentes servicios básicos que se proveen en el hogar como lo son el suministro de energía, de agua y de gas. Por medio de la implementación de sistemas inteligentes es posible tener un mayor control del consumo de dichos servicios a través de la adecuación de alarmas que indiquen al usuario de posibles fugas o consumos inesperados por encima del promedio, que permitan generar acciones de control y mitigación con el fin de disminuir estas pérdidas y hacer un eficiente uso de los recursos.

Del mismo modo, y particularmente en el territorio Colombiano, actualmente las mediciones de los servicios básicos se efectúa por medio de sistemas mecánicos sobre los cuales se realizan lecturas periódicas (por personal capacitado) con el fin de determinar el consumo. A razón de esto, teniendo en cuenta que este sistema es propenso a fallas humanas en la ejecución de la lectura, en ocasiones se presentan errores de facturación que perjudican al propietario de la vivienda. De tal manera, que la implementación de tecnologías de IoT se presenta como una opción favorable para la medición de dichos servicios, pues esta tarea se realizaría de forma automática, en tiempo real y de una forma precisa, así como también incluiría la prestación de funcionalidades adicionales como la generación de alarmas y el monitoreo remoto tanto por el propietario de la vivienda como de las empresas operadoras de los servicios.

En el presente documento se hace una descripción de la implementación de un prototipo de sistema IoT para la medición del consumo de agua en hogares por medio de un sensor de flujo para la medición de la variable, una tarjeta de procesamiento Raspberry Pi 3 para la captura de las señales y algunos componentes de la plataforma AWS (Amazon Web Services) para la ejecución de un servicio web que permita visualizar en tiempo real la variación del consumo de agua.

## **2. OBJETIVO GENERAL**

Realizar una descripción detallada de la implementación de un prototipo de sistema IoT para la medición, en tiempo real, del consumo de agua en hogares por medio de un sensor de flujo para la medición de la variable, una tarjeta de procesamiento Raspberry Pi 3 para la captura de las señales y comunicación con la plataforma AWS (Amazon Web Services), la cual dispondrá de un servicio web que permita visualizar en tiempo real la variación del consumo de agua.

### 3. COMPONENTES DEL SISTEMA IoT

Para la implementación del prototipo de sistema IoT para la medición del consumo de agua se propone utilizar los siguientes componentes:

- Sensor de flujo de líquido referencia YF-S201.
- Tarjeta Raspberry Pi 3 Modelo B.
- Router de comunicaciones con acceso a internet.
- Cuenta en AWS con acceso a los módulos de AWS IoT Core y AWS ElasticSearch Service.
- Protocolo MQTT para comunicaciones.

#### 3.1. Sensor de flujo de líquido YF-S201

Este sensor internamente contiene una rueda dentada que gira con el paso del flujo de agua. De igual manera, cuenta con un sensor de efecto hall integrado que genera un pulso eléctrico para cada revolución. La figura 1 gráficamente detalla los componentes internos de este sensor.

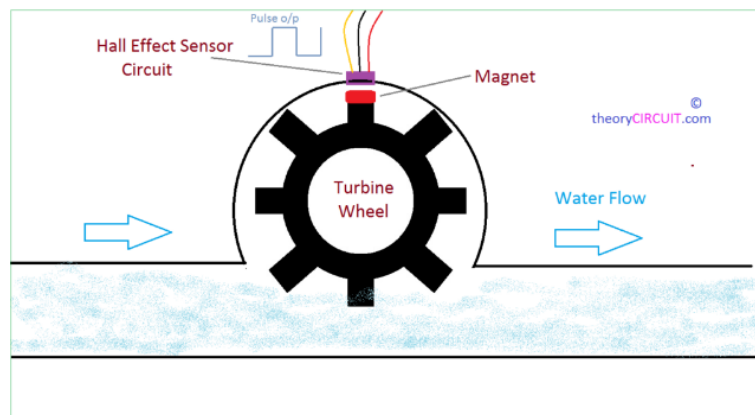


Figura 1: Vista interna del sensor de flujo de líquido.

Básicamente, cuando el flujo de agua pasa por el dispositivo, este hace que la turbina rote y por lo tanto, el componente magnético causa una interferencia en el sensor de efecto hall (la velocidad de interferencia depende de la velocidad del flujo de agua) produciendo un pulso eléctrico de salida que puede ser utilizado para calcular el volumen de agua. En teoría cada pulso es aproximadamente 2.25 mililitros. La figura 2 muestra una representación física del sensor y sus partes.



Figura 2: Vista externa del sensor de flujo de líquido.

### 3.1.1. Efecto Hall

El efecto hall se produce cuando se ejerce un campo magnético transversal sobre un cable por el que circula una corriente. Como la fuerza magnética ejercida sobre esta es perpendicular al campo magnético y a su velocidad, las cargas son impulsadas hacia un lado del conductor y se genera un potencial transversal o voltaje hall. La figura 3 muestra una ilustración del efecto hall.

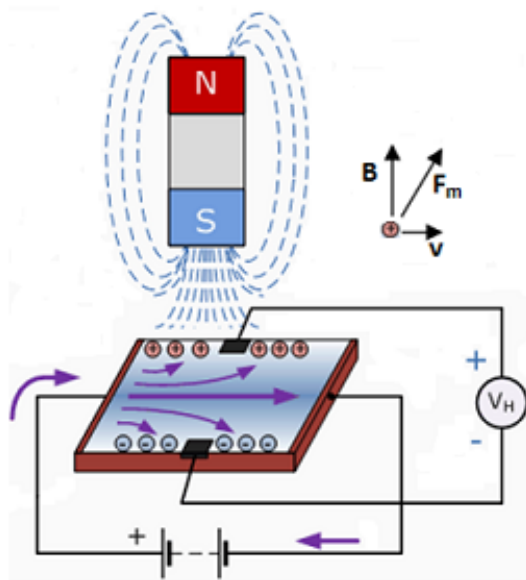


Figura 3: Representación física del efecto hall.

Como se observa, al interponer un componente magnético perpendicular sobre un elemento por el cual circula una corriente, se genera una diferencia de potencial transversal, debido al movimiento de las cargas, conocido como voltaje hall. Este principio es utilizado por el sensor de flujo detallado en la sección anterior.

### **3.2. Raspberry Pi**

La Raspberry Pi es una serie de pequeños computadores desarrollados en Inglaterra (UK) por la fundación “Raspberry Pi” para promover la enseñanza de las ciencias de la computación en escuelas y países en vía de desarrollo. Diversos modelos han sido elaborados, aunque todos utilizan un chip Broadcom con una CPU integrada ARM (Advanced RISC Machine) y una unidad de GPU (Graphics Processing Unit).

La velocidad de los procesadores varia en un rango de 700 MHz a 1.4 GHz para el modelo Pi 3 B+, con memoria RAM integrada desde 256 MB a 1 GB. Tarjetas SD (Secure Digital) son usadas para almacenar el sistema operativo y la memoria de programas en cualquier forma SDHC (SD High Capacity) o MicroSDHC. Las tarjetas cuentan con puertos USB, la salida de video es HDMI (High Definition Multimedia Interface) y un puerto de audio estándar 3.5 mm. De igual manera, cuenta con una tarjeta de puertos de entrada/salida GPIO (general Purpose Input/Output) que soportan protocolos comunes como I2C (Integrated-Integrated Circuit). Los modelos B tiene un puerto Ethernet y el Pi 3 y Pi Zero W cuentan con conexión Wifi 802.11n y Bluetooth.

#### **3.2.1. Raspberry Pi 3 Modelo B**

El modelo Raspberry Pi 3 B fue liberado en Febrero de 2016 con un procesador Quad Core de 64 bits, Wifi integrado, Bluetooth y capacidad de arranque vía USB. Fundamentalmente las características de este modelo de Raspberry son las siguientes:

- Procesador Quad Core de 1.2 GHz Broadcom BCM3827 64 bits.
- 1 GB de RAM.
- Módulo Wireless BCM43438 y Bluetooth de baja energía (BLE) integrado.
- Puerto Ethernet Base 100.
- 40 puertos GPIO.
- 4 puertos USB.

- Puerto HDMI.
- Puerto para conectar cámara para Raspberry Pi.
- Puerto para conectar pantalla Raspberry Pi.
- Puerto Micro SD para carga de sistema operativo y almacenamiento de datos.
- Puerto micro USB para alimentación de energía hasta 2.5 A.

La figura 4 muestra una vista interna/externa de la Raspberry 3 modelo B utilizada.

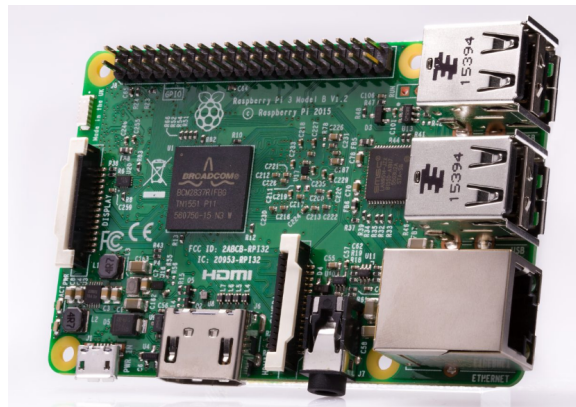


Figura 4: Raspberry Pi 3 Modelo B.

### 3.3. AWS IoT Core

AWS IoT Core es una plataforma en la nube administrada que permite conectar dispositivos fácilmente y de forma segura, interactuar con aplicaciones en la nube y otros dispositivos. AWS IoT Core puede soportar billones de dispositivos y mensajes, y puede procesar y enrutar esos mensajes a puntos finales AWS y otros dispositivos de forma confiable y segura. AWS IoT Core permite integrar servicios adicionales como AWS Lambda, Amazon Kinesis, Amazon S3, Amazon Machine Learning, Amazon DynamoDB, Amazon CloudWatch, AWS CloudTrail, y Amazon Elastic Search, para construir aplicaciones que juntan procesos, analizan y actúan sobre los datos generados por los dispositivos sin tener que administrar ninguna infraestructura de comunicaciones adicional.

### 3.4. AWS Elastic Search Service

Amazon ES es un servicio administrado que hace fácil desplegar, operar y escalar clusters de ElasticSearch en la nube de AWS. Elastic Search es un

servicio popular de código abierto utilizado como motor de búsqueda y análisis de datos, monitoreo de aplicaciones en tiempo real y análisis de stream de datos. Amazon ES cuenta con una herramienta denominada KIBANA, la cual permite generar análisis gráficos de datos de forma automática y en tiempo real.

### 3.5. Protocolo MQTT

MQTT es un protocolo de transporte de mensajería cliente - servidor basado en el principio de publish/subscribe. Es un protocolo liviano, abierto, simple y diseñado para ser de fácil implementación. Estas características lo hacen ideal para ambientes de recursos limitados como la comunicación máquina a máquina (M2M) y sistemas de IoT donde el ancho de banda es un recurso escaso y el consumo bajo de energía es importante. El protocolo fue desarrollado por Andy Stanford-Clark (IBM) y Arlen Nipper (Eurotech/Cirrus Link) en 1999.

El protocolo usa una arquitectura de “publish/subscribe” a diferencia de HTTP que utiliza un paradigma de solicitud/respuesta. Publish/subscribe es una acción manejada por eventos que habilita el envío de mensajes a los clientes. El punto central de comunicación se denomina MQTT “broker”, el cual está encargado de distribuir todos los mensajes entre los emisores y los receptores adecuados. Cada cliente que publica un mensaje al broker, incluye un “topic” (tema) en el mensaje. El topic es la información de enrutamiento para el broker. Cada cliente que quiere recibir mensajes se suscribe a un topic particular y el broker entrega todos los mensajes con el topic asociado hacia el cliente. Por lo tanto, los clientes no requieren conocerse entre ellos, ellos solo se comunican sobre el topic. La figura 5 muestra la arquitectura básica del protocolo MQTT.

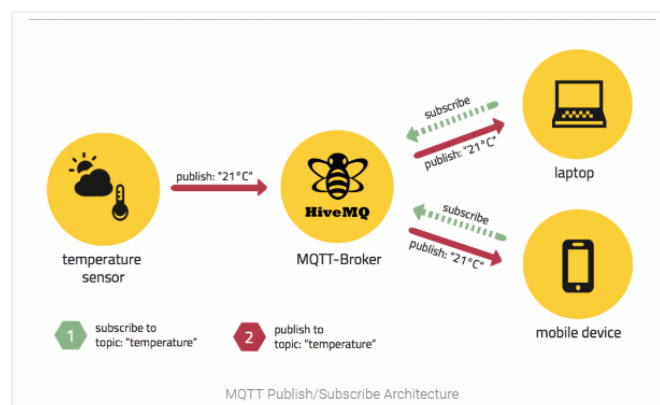


Figura 5: Protocolo MQTT.



La diferencia a HTTP es que el cliente no tiene que halar la información que este necesita, pero el broker envía la información al cliente, en el caso de que haya información nueva. De tal manera que el cliente MQTT tiene una conexión TCP abierta permanente hacia el broker. Si la conexión es interrumpida, el broker MQTT puede almacenar todos los mensajes y enviarlos cuando el cliente recupera conexión. Un topic o tema es un simple string que tiene niveles jerárquicos separados por un slash (/), por ejemplo: casa/cocina/temperatura.

El protocolo MQTT opera sobre TCP/IP y por tanto el cliente como el broker deben soportar el stack TCP/IP. La figura 6 muestra la ubicación del protocolo a nivel de las capas del modelo OSI.

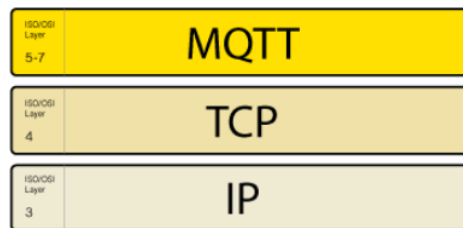


Figura 6: Ubicación de MQTT en las capas del modelo OSI.

## 4. ARQUITECTURA DEL SISTEMA IoT

Luego de definir los componentes utilizados en el sistema IoT propuesto, en la figura 7 se presenta un diagrama en general de la arquitectura de la solución planteada.

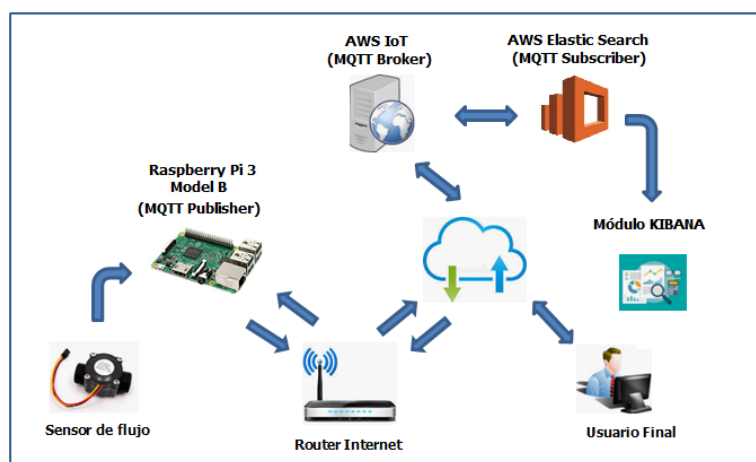


Figura 7: Arquitectura de la solución propuesta del sistema IoT.

## 5. CONFIGURACION DEL SISTEMA IoT

La implementación del sistema IoT para la medición remota del consumo de agua incluye la configuración de cada uno de los componentes mostrados en la arquitectura señalada en la figura 7. La descripción de cada una de estas etapas se puntualiza en cada una de las siguientes secciones.

### 5.1. Configuración de Raspberry Pi

De acuerdo a los ejemplos de configuración encontrados en la Web, a pesar de que AWS dispone de librerías para la programación de sistemas IoT en diferentes lenguajes como Java, C y Python, se decide trabajar con este último, pues ha sido ampliamente utilizado en este tipo de aplicaciones y los resultados han sido favorables.

La puesta en marcha de la Raspberry para garantizar la comunicación con la plataforma de AWS y la integración del sensor de flujo de agua requiere el despliegue de algunos componentes de software que junto con los comandos utilizados se listan a continuación. Cabe aclarar que el sistema operativo RASPBIAN se encuentra pre-instalado en una tarjeta Micro SD y operativo en la Raspberry.

#### 5.1.1. Actualización de firmware y sistema operativo

Para actualizar el firmware y el sistema operativo de la Raspberry Pi se deben ejecutar los siguientes comandos en la CLI de la tarjeta:

```
$ sudo apt-get install rpi-update
$ sudo rpi-update           #Actualiza el firmware
$ sudo reboot
$ sudo apt-get update       #Actualiza información de paquetes
$ sudo apt-get dist-upgrade #Actualiza el sistema en general
```

#### 5.1.2. Instalación de Python

Para instalar Python se deben ejecutar los siguientes comandos. En este caso se utilizó la versión 3.4.2:

```
$ wget https://www.python.org/ftp/python/3.4.2/Python-3.4.2.tgz
$ tar xvfz Python-3.4.2.tgz
$ cd Python-3.4.2/
$ ./configure --prefix=/opt/python3.4
$ make
$ sudo make install
```

Una vez completada la instalación, es posible verificar la versión utilizando el siguiente comando:

```
$ python3 -version
```

### 5.1.3. Instalación de openssl

Teniendo en cuenta que la autenticación del dispositivo en la plataforma de AWS se realiza por medio del uso de certificados digitales sobre el protocolo TLS (Transport Layer Security), se hace necesaria la instalación de openssl para poder establecer la comunicación segura desde el dispositivo. Para instalar openssl se debe ejecutar el siguiente comando:

```
$ sudo apt-get install openssl
```

Para verificar la versión instalada:

```
$ openssl version
```

### 5.1.4. Instalación de AWS IoT SDK

Para la comunicación del dispositivo, en este caso la Raspberry, con la plataforma AWS es posible desarrollar el código en diferentes lenguajes de programación como C, Java o Python. Para nuestro prototipo se ha decidido trabajar con este último dada su popularidad en este tipo de aplicaciones. De tal manera, que se hace necesario, aparte de instalar el lenguaje de programación Python como se mostró anteriormente, instalar el SDK específico para desplegar la conexión con AWS. EL SDK de AWS para Python se denomina AWSIoTPythonSDK. La forma más sencilla de instalar este paquete es ejecutando el siguiente comando:

```
$ pip install AWSIoTPythonSDK
```

PIP es un repositorio de Python (Python package Index).

Información adicional acerca de la instalación de este componente se puede consultar en el siguiente repositorio de gitHub: <https://github.com/aws/aws-iot-device-sdk-python>.

### 5.1.5. Instalación de la librería de GPIO

Con el fin de manipular los puertos GPIO (General Purpose Input/Output) de la Raspberry Pi y poder recibir las señales del sensor de flujo se hace necesario instalar la librería de GPIO utilizando la siguiente instrucción:

```
$ sudo pip install RPi.GPIO
```

## 5.2. Configuración de AWS IoT Core

AWS IoT core es un componente de la plataforma de AWS que permite establecer comunicación con diversos dispositivos para el intercambio de mensajes que permiten identificar el estado de diversos procesos y ejecutar acciones de control establecidas de acuerdo a las necesidades del sistema. Primordialmente, AWS establece la comunicación utilizando el protocolo MQTT explicado con anterioridad.

Como primer paso, para acceder al módulo de AWS IoT Core se debe acceder a la consola de AWS como se observa en la figura 8.

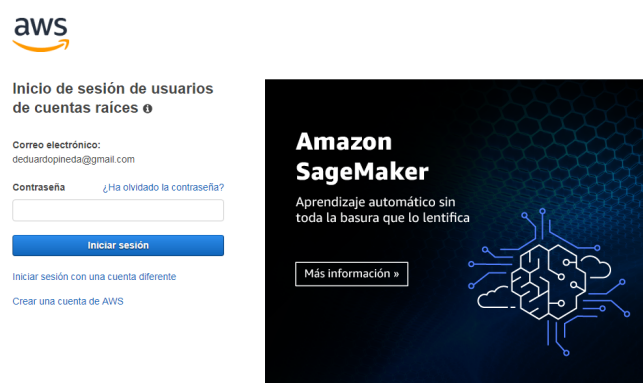


Figura 8: Página de inicio de sesión en AWS.

Una vez autenticado en la consola, basta con desplegar todos los servicios disponibles y en la parte inferior se encontrará el módulo IoT Core como se muestra en la figura 9.

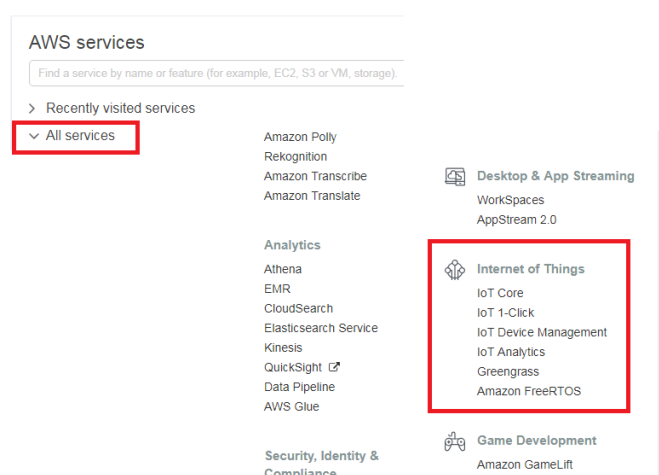


Figura 9: Módulo de IoT Core en AWS.

Luego acceder al módulo de IoT Core se muestra una interfaz como la indicada en la figura 10.

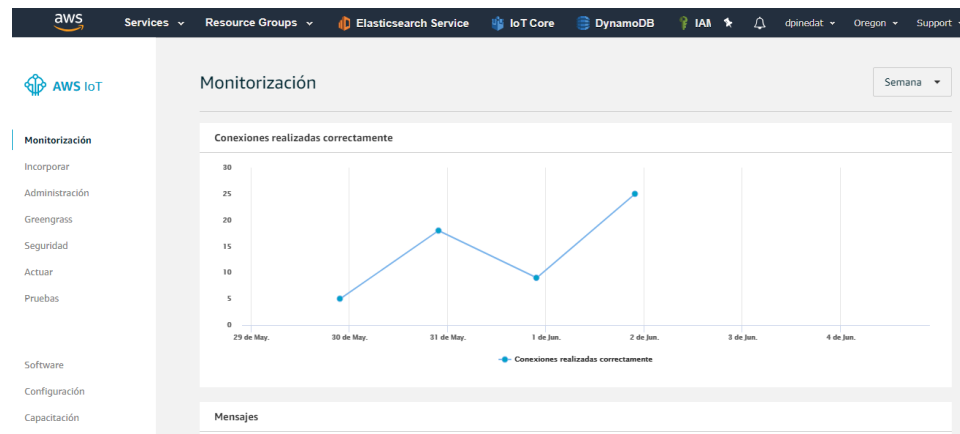


Figura 10: Módulo IoT Core.

### 5.2.1. Registro de dispositivo en AWS IoT

Para establecer la comunicación con el dispositivo es necesario configurar un registro en la plataforma. La primera vez que se accede a esta, se verá una interfaz como se muestra en la figura 11.

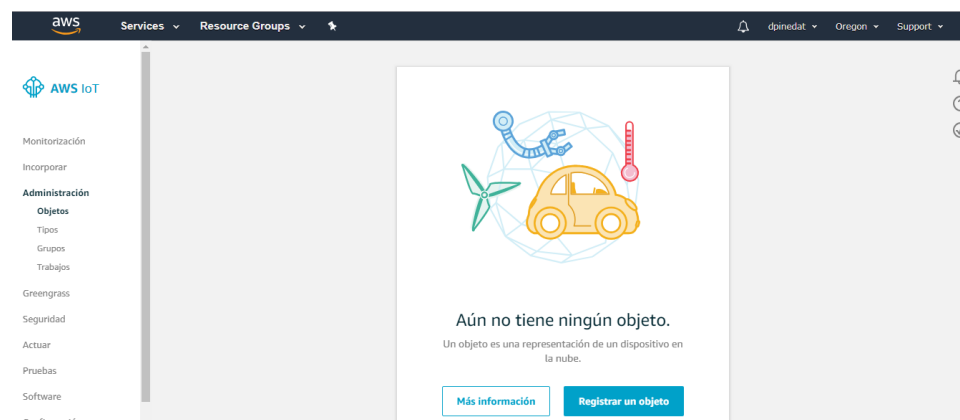


Figura 11: Paso 1. Registro de objeto.

Una vez dar click en la opción “Registrar un objeto” se desplegarán las opciones que se muestran en la figura 12.

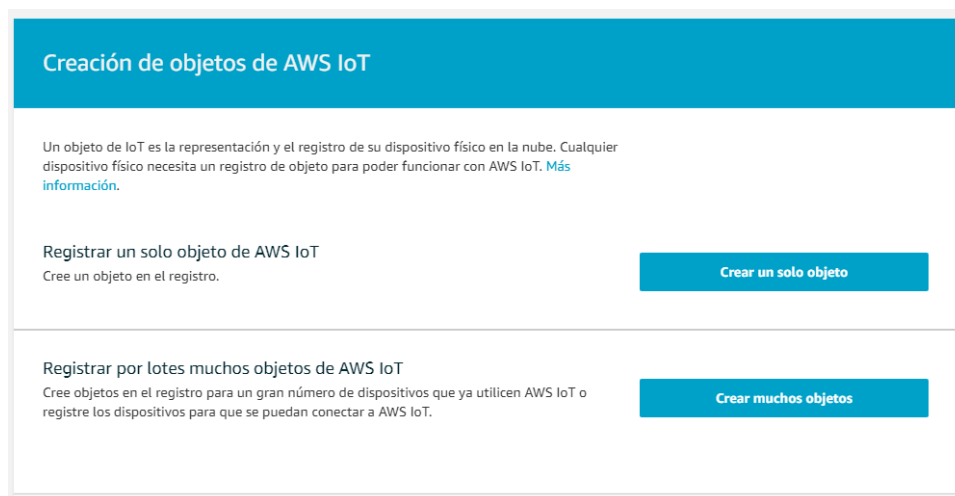


Figura 12: Paso 2. Creación de objeto.

Teniendo en cuenta que en este caso se requiere registrar un único objeto (la Raspberry Pi) se debe dar click en la opción “Crear un solo objeto”. La siguiente ventana que se mostrará se indica en la figura 13.

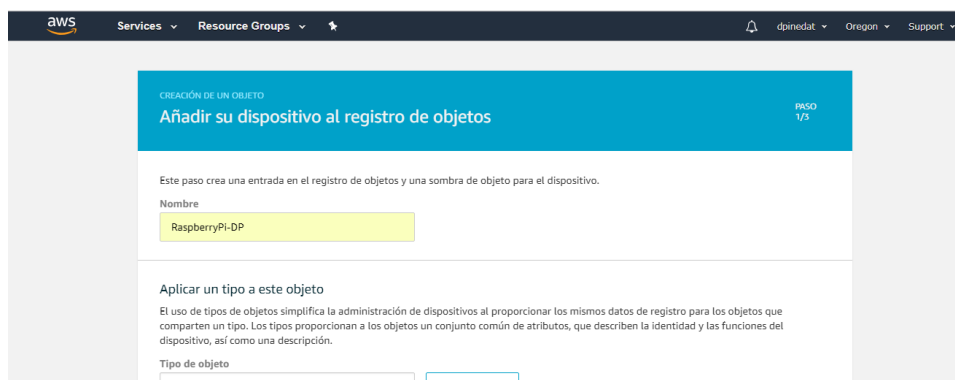


Figura 13: Paso 3. Asignación de nombre del objeto.

En esta ventana se debe primordialmente asignar un nombre al dispositivo. El siguiente paso es la generación de los certificados digitales para identificar el objeto o dispositivo, como se muestra en la figura 14.

CREACIÓN DE UN OBJETO

Añadir un certificado para el objeto PASO 2/3

Los certificados se utilizan para autenticar la conexión del dispositivo con AWS IoT.

**Creación de un certificado con un clic (recomendado)**  
Se generará un certificado, una clave pública y una clave privada mediante la entidad de certificación de AWS IoT.

**Crear con CSR**  
Cargue su propia solicitud de firma de certificado (CSR) basada en su propia clave privada.

**Usar mi certificado**  
Registre su certificado de CA y use sus propios certificados en todos los dispositivos que desee.

Crear certificado

Crear con CSR

Empezar

Figura 14: Paso 4. Creación de certificados.

Con tan solo dar click en la opción “Crear certificado” se crean los certificados digitales que identifican el dispositivo y serán necesarios para el proceso de autenticación. La figura 15 muestra el resultado de la creación de los certificados.

El certificado se ha creado.

Descargue estos archivos y guárdelos en un lugar seguro. Los certificados se pueden recuperar en cualquier momento, pero las claves privadas y públicas no se pueden recuperar después de cerrar esta página.

Para conectar un dispositivo, necesita descargar lo siguiente:

Un certificado para este objeto	5063d05e48.cert.pem	Descargar
Una clave pública	5063d05e48.public.key	Descargar
Una clave privada	5063d05e48.private.key	Descargar

Necesita descargar también una CA raíz para AWS IoT de Symantec:  
Una entidad de certificación raíz para AWS IoT Descargar

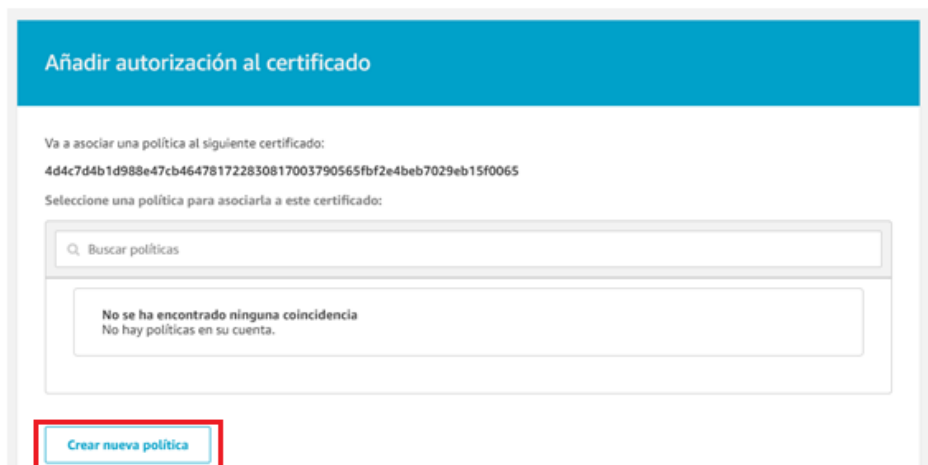
Activar

LANZAR LISTO Asociar una política

Figura 15: Paso 5. Descarga de certificados.

Como se observa en la figura, se deberán descargar 4 certificados, 3 que serán asociados al dispositivo y uno que identifica la CA que realiza la emisión de los mismos. Una vez descargados, se debe dar click en la opción

“Asociar una política”, luego de lo cual se desplegará la ventana mostrada en la figura 16.



Añadir autorización al certificado

Va a asociar una política al siguiente certificado:  
4d4c7d4b1d988e47cb464781722830817003790565fbf2e4beb7029eb15f0065

Seleccione una política para asociarla a este certificado:

Buscar políticas

No se ha encontrado ninguna coincidencia  
No hay políticas en su cuenta.

Crear nueva política

Figura 16: Paso 6. Creación de política.

Luego, se deberá crear una política dando click al botón “Crear nueva política” indicado en la figura. Una vez hecho esto se desplegará una nueva ventana como la mostrada en la figura 17.



Crear una política

Cree una política para definir un conjunto de acciones permitidas. Puede permitir acciones en uno o varios recursos (objetos, temas o filtros de temas). Para obtener más información sobre las políticas de IoT, consulte la [página de documentación de políticas de AWS IoT](#).

Nombre  
RaspberryPi-DP-Política

Añadir declaraciones  
Las declaraciones de política definen los tipos de acciones que puede realizar un recurso. Modo avanzado

Acción  
iot:\*

ARN de recurso  
\*

Efecto  
☒ Permitir ☐ Denegar

Quitar

Figura 17: Paso 7. Configuración de política.

Como se muestra, la acción se debe definir como “iot:\*” con el fin de



permitir diferentes mensajes de acción sin ninguna restricción. Modificaciones deberán tener lugar en ambientes de producción. El “ARN de recurso” deberá ser marcado con un (\*) y se debe escoger la opción “Permitir”. Una vez definidos estos valores, se debe dar click en el botón de “Crear”.

Una vez creada la política, esta debe ser asociada al certificado digital. Para esto se debe acceder al menú de “Seguridad - Certificados” como se muestra en la figura 18.

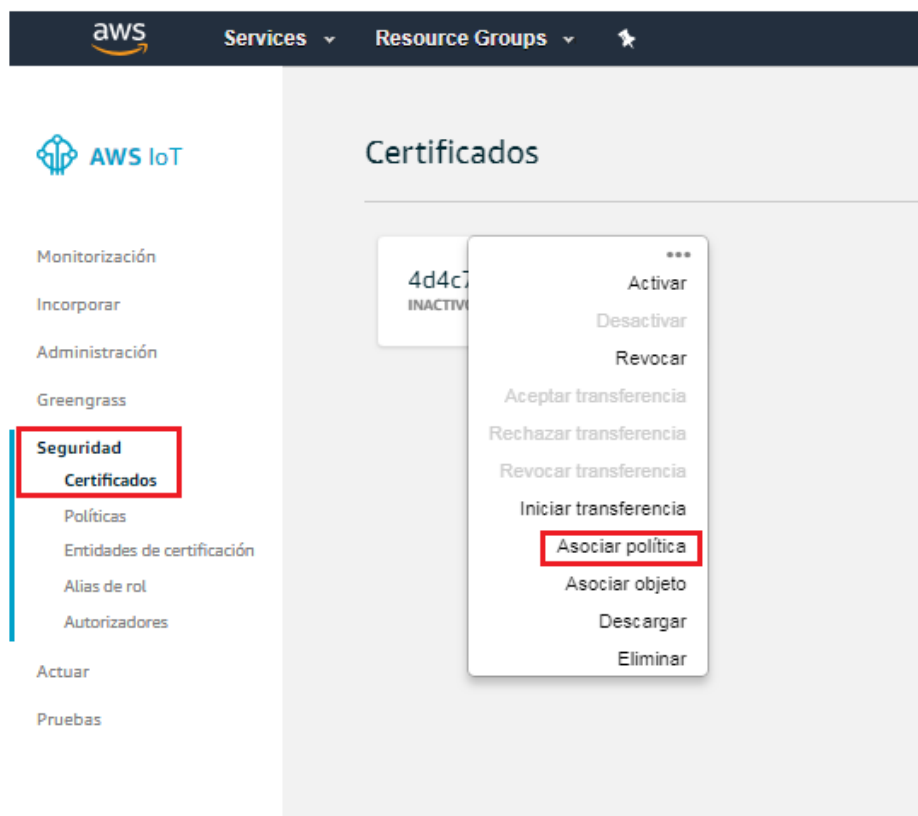


Figura 18: Paso 8. Asociación de política a certificado.

Luego se deberá asociar la política creada anteriormente como se muestra en la figura 19.



Figura 19: Paso 9. Asociación de política a certificado.

Una vez hecho esto, se debe asociar ahora el certificado digital al objeto creado inicialmente. Para esto nuevamente en el vínculo de Certificados, se selecciona el certificado y se da click en “Asociar objeto” como se muestra en la figura 20.

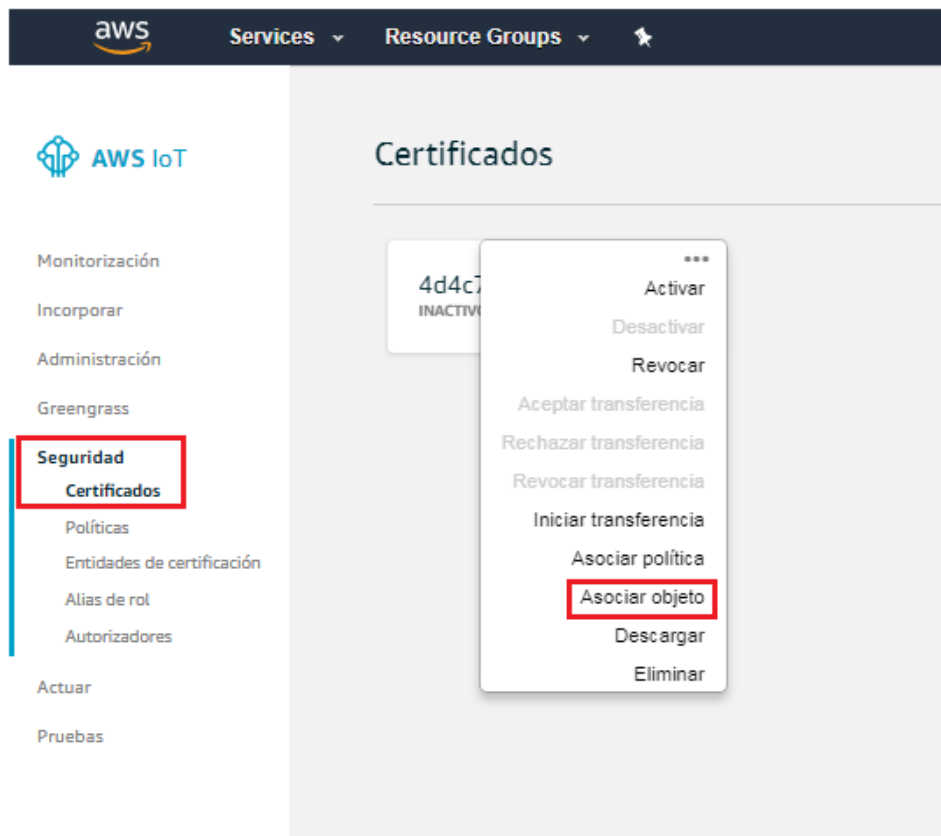


Figura 20: Paso 10. Asociación de objeto a certificado.

Luego se debe seleccionar el objeto creado como se indica en la figura 21.

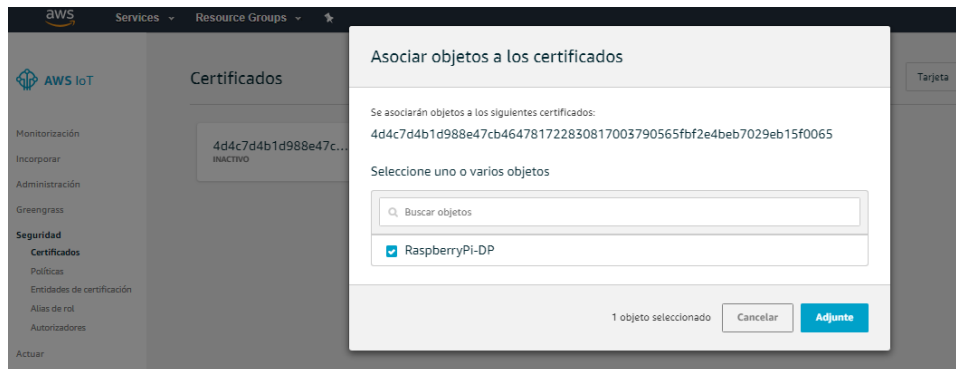


Figura 21: Paso 11. Asociación de objeto a certificado.

Por último es importante verificar que el certificado se encuentre activo, como se muestra en la figura 22.

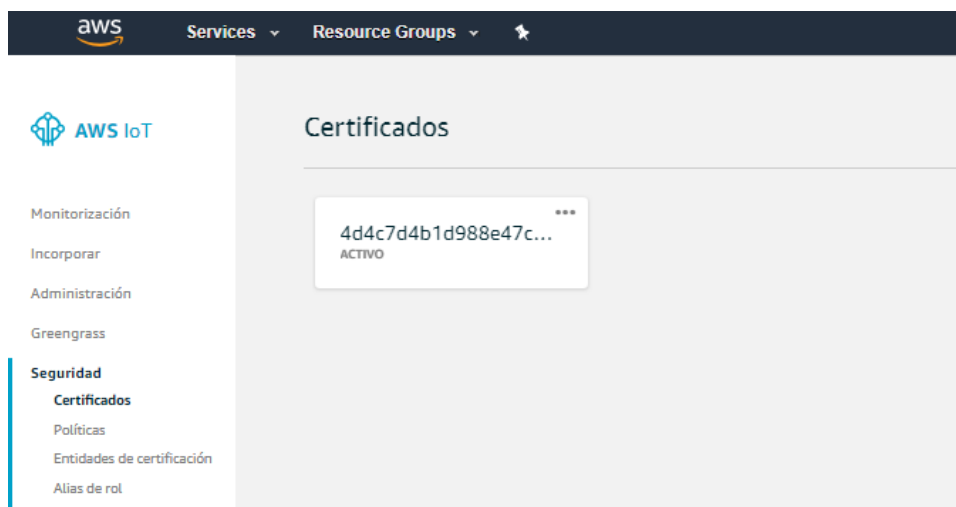


Figura 22: Paso 12. Activación de certificado.

Una vez realizados todos los pasos indicados se dispondrá del registro del objeto, asociado con el certificado digital que a su vez se encuentra ligado a una política de seguridad que permite la ejecución de diferentes acciones de comunicación del dispositivo con la plataforma de IoT de AWS.

Los certificados que fueron descargados deben ser almacenados en el dispositivo cliente, en nuestro caso, el Raspberry Pi.

### 5.3. Configuración de AWS Elastic Search (ES)

El componente de AWS Elastic Search Service es un servicio que permite ejecutar el análisis de datos de forma automática. Para la configuración de este módulo se debe crear un dominio como se muestra en la figura 23.

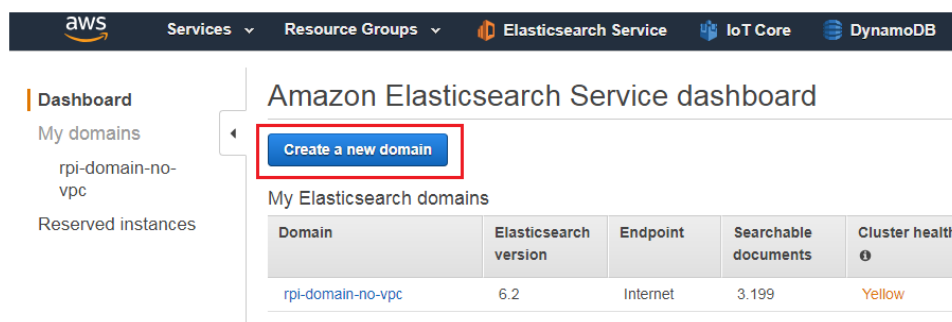


Figura 23: Creación de dominio en Amazon ES.

Luego se debe asignar un nombre al dominio y escoger una versión a desplegar como se muestra en la figura 24.

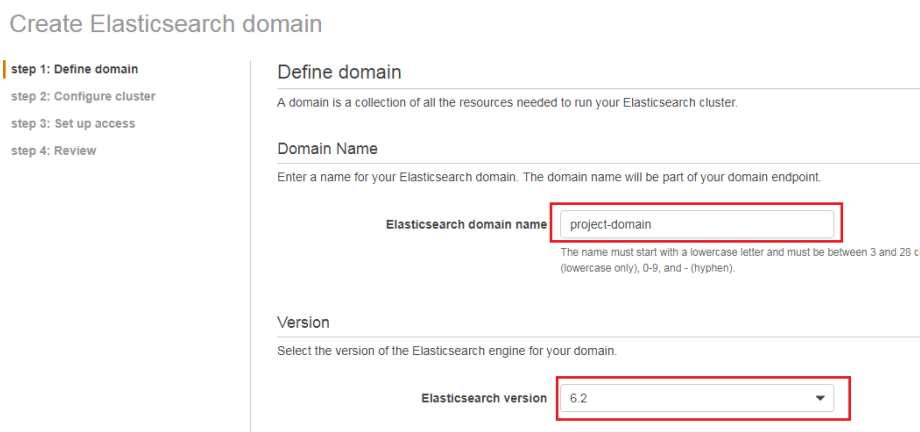


Figura 24: Configuración de dominio.

Posteriormente se deben asignar las opciones de configuración del cluster, las cuales se dejan en sus valores por defecto, pues resultan funcionales para el prototipo propuesto.

El siguiente paso es determinar el método de acceso al dominio, en este caso por medio de acceso público a través de internet como se muestra en la figura 25. De igual forma es necesario especificar las direcciones IP permitidas como se muestra en la figura 26.

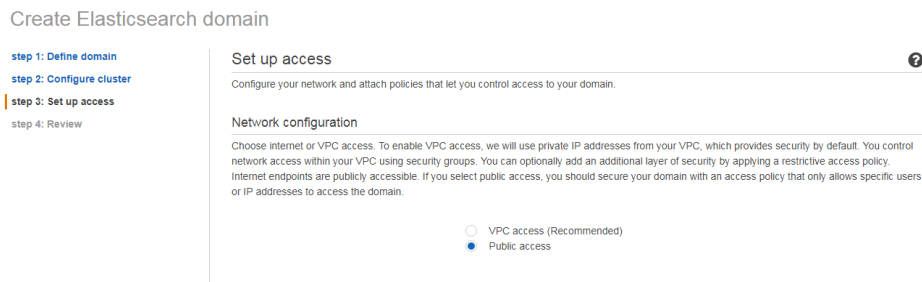


Figura 25: Configuración de acceso al dominio.



Figura 26: Configuración de IPs públicas permitidas.

Las direcciones IP definidas en la política de acceso son las únicas que tendrán acceso al dominio y a los componentes adicionales del mismo, como KIBANA, el cual es usado en el presente proyecto para la visualización en tiempo real de los datos del sensor de flujo de agua.

Por último se presenta un resumen de los parámetros de configuración definidos para el dominio antes de continuar con la creación, como se muestra en la figura 27.

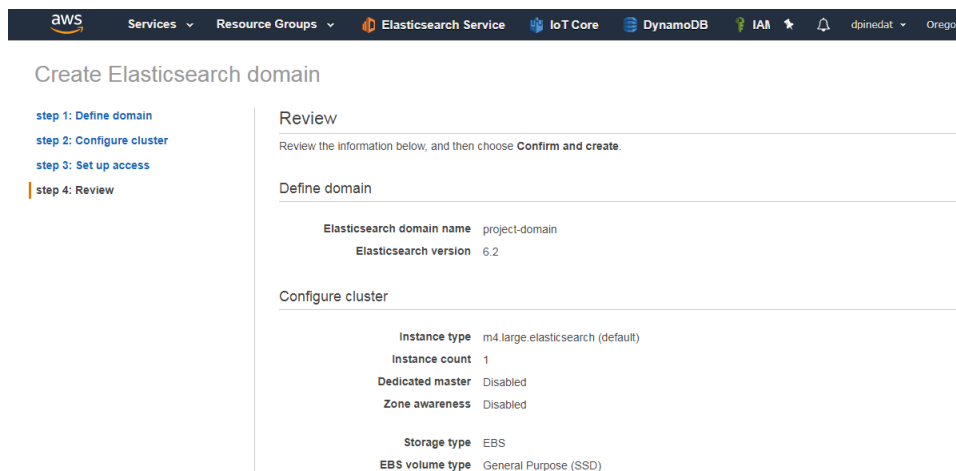


Figura 27: Revisión de parámetros del dominio.

Una vez creado el dominio de Elastic Search Service, se tiene una entrada en el dashboard como se muestra en la figura 28.

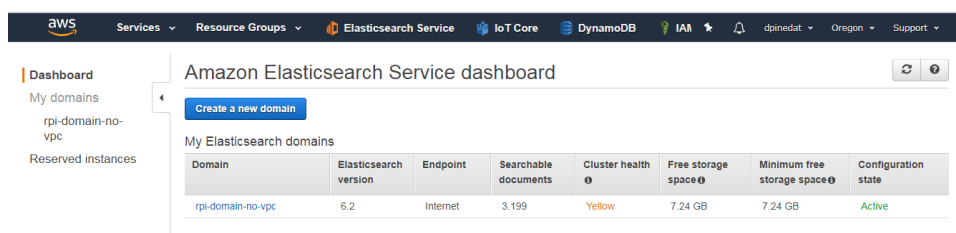


Figura 28: Verificación del dominio creado.

**NOTA:** Es importante tener en cuenta que este servicio de Elastic Search Service genera cargos de facturación en la cuenta de AWS, es decir es pago.

Una vez hecho esto se debe crear un índice en el servicio con el fin de contener los mensajes provenientes del módulo IoT Core que corresponden a las mediciones enviadas por el sensor de flujo. La creación del índice se puede realizar a partir de una función HTTP PUT generada por un cliente HTTP. EN este caso, se utilizó el cliente HTTP de google chrome para enviar el método PUT para la creación del índice en Amazon ES, como se muestra en la figura 29.

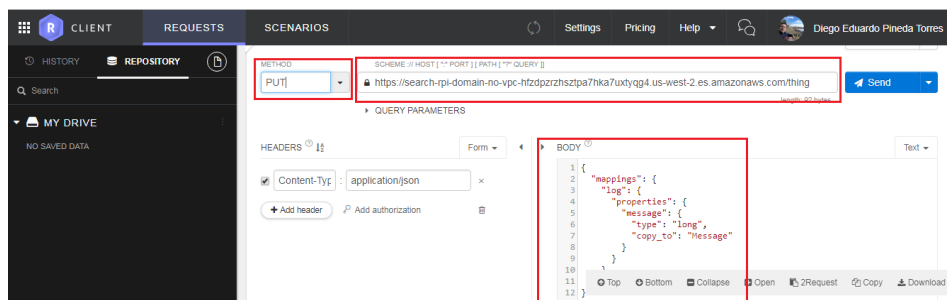


Figura 29: Método HTTP PUT para creación de índice en Amazon ES.

Básicamente, la URL del método PUT debe apuntar al identificador del dominio de Amazon ES, junto con el nombre del “topic” o tema que se utiliza como la identificación de los datos del sensor. Dentro del BODY del método se debe indicar el mapeo de los valores presente en el mensaje como se muestra en la figura. Una vez lanzado el comando desde el cliente HTTP, es posible verificar en Amazon ES que el índice ha sido creado, como se observa en la figura 30.

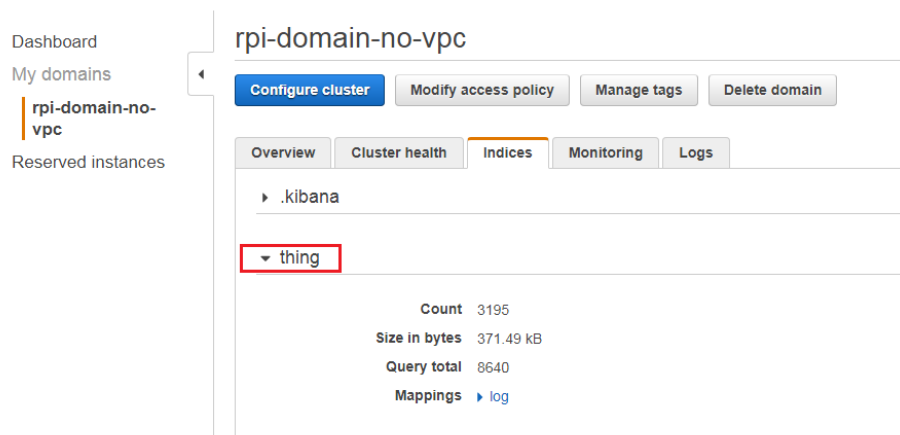


Figura 30: Verificación de índice en dominio de ES.

El valor resaltado en la figura, “thing”, corresponde al tópico o tema que debe concordar con la regla creada en AWS IoT para que el envío de la información se realice de forma adecuada.

#### 5.4. Configuración de regla de operación en AWS IoT

Una vez realizada la configuración del registro del dispositivo en la plataforma de AWS IoT y el despliegue del dominio en el componente de Amazon ES (Elastic Search), se hace necesaria la configuración de

una regla en el componente de IoT Core, con el fin de enrutar los datos que se reciben del sensor de forma automática hacia el componente de Amazon ES para su posterior visualización con el módulo de KIBANA. Para la creación de la regla, se debe acceder al vínculo de “Actuar” en AWS IoT como se muestra en la figura 31.

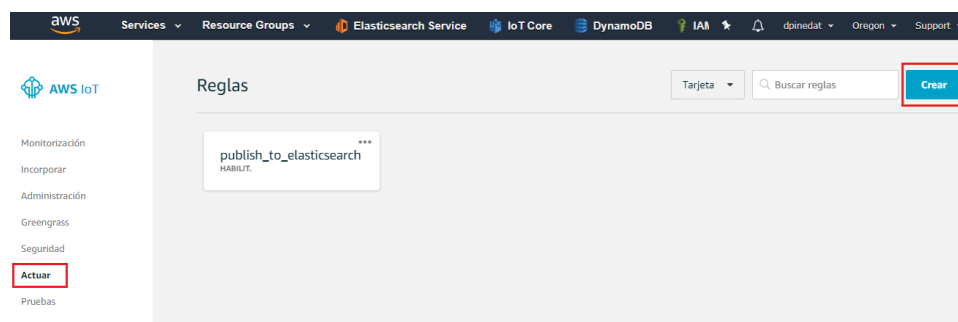


Figura 31: Paso 1. Creación de regla.

Luego de dar click en el botón crear, se deben ajustar los parámetros de la regla, comenzando por un nombre y una descripción de la misma, como se muestra en la figura 32.

Figura 32: Paso 2. Asignación de nombre/descripción.

De igual forma se deben ajustar los parámetros mostrados en la figura 33.



Origen del mensaje

Indique el origen de los mensajes que desea procesar con esta regla.

Uso de la versión de SQL [?](#)

2016-03-23 ▼

Instrucción de consulta de regla

```
SELECT * FROM 'thing/data'
```

Atributo [?](#)

\*

Filtro de temas [?](#)

thing/data

Condición [?](#)

p. ej., temperatura > 75

Figura 33: Paso 3. Configuración de parámetros.

Como se observa, dentro de los parámetros de configuración se debe definir la versión de SQL a utilizar, los atributos que se requieren filtrar del mensaje enviado por el sensor (en este caso se tomaran todos, por lo cual se utiliza la opción \*) y se debe señalar el string del “topic” o tema que identifica los mensajes, en este caso “thing/data”. Posteriormente, se debe añadir la acción que la regla deberá tomar con los mensajes recibidos. Para esto se debe dar click en la opción “Añadir acción” presentada en la figura 34.

Filtro de temas [?](#)

thing/data

Condición [?](#)

p. ej., temperatura > 75

---

Definir una o varias acciones

Seleccione una o varias de las acciones que se deben producir cuando la regla anterior coincida con un mensaje entrante. Las acciones definen actividades adicionales que se producen cuando llegan mensajes, como almacenarlos en una base de datos, invocar funciones de la nube o enviar notificaciones. (\*obligatorio)

[Añadir acción](#)

---

Acción de error

Si lo desea, defina la acción que se ejecutará cuando se produzca un error al procesar la regla.

[Añadir acción](#)

Figura 34: Paso 4. Añadir acción a la regla.

Una vez hecho esto, se despliega un conjunto de recursos o servicios hacia los cuales se pueden enrutar los mensajes. En este caso particular, nuestro objetivo es enviar los datos al componente de Amazon ES, como se muestra en la figura 35.



Figura 35: Paso 5. Selección de destino de acción.

Luego de esto se debe configurar la acción, definiendo los parámetros que se muestran en la figura 35.

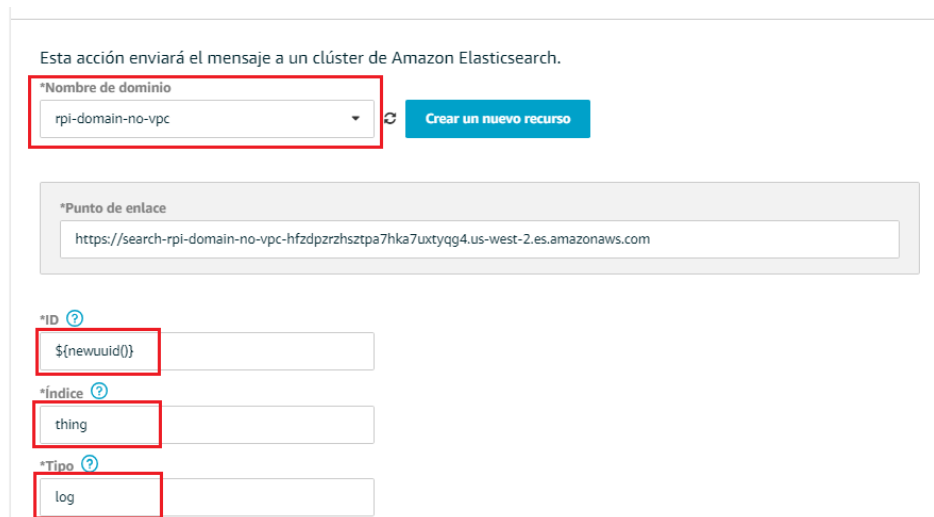


Figura 36: Paso 6. Configuración de acción.

En este paso es importante seleccionar el dominio de Amazon ES creado anteriormente, e indicar el índice que fue definido en el dominio, en este caso “thing”. Por último, se debe seleccionar un role que permita

el acceso al dominio como se muestra en la figura 37.

Elija o cree un rol para conceder acceso a AWS IoT al recurso Amazon Elasticsearch para que realice esta acción.

\*Nombre de rol de IAM  
elastic-role

Actualizar rol Crear un nuevo rol

Cancelar Añadir acción

Figura 37: Paso 7. Configuración de rol.

El rol debe permitir acceso al dominio creado en Amazon ES, como se muestra en la figura 38.

Search IAM

Dashboard Groups Users Roles Policies Identity providers Account settings Credential report Encryption keys

Maximum CLI/API session duration 1 hour (3,600 seconds) Edit

Permissions Trust relationships Access Advisor Revoke sessions

Attach policy Attached policies: 1

Policy name Policy type

aws-iot-role-es\_814265244 Managed policy

Policy summary {} JSON Edit policy Simulate policy

Filter

Service	Access level	Resource	Request condition
Elasticsearch Service	Limited: Write	DomainName   string like   rpi-domain-no-vpc*	None

Figura 38: Paso 8. Verificación del rol.

## 5.5. Programa de ejecución en Raspberry Pi

Para la captura y procesamiento de las señales del sensor de flujo de agua así como también establecer la conexión de la Raspberry Pi con la plataforma AWS, se hace necesario el desarrollo del código de ejecución de tareas del sistema. Como se mencionó en las etapas iniciales, se propone la definición del código utilizando el lenguaje de programación Python, pues es ampliamente referido en la literatura, junto con el SDK de AWS para este lenguaje, AWSIoTPythonSDK, que se encuentra ya desplegado en la Raspberry. El código propuesto se muestra en la siguiente página.

```

import os
import sys
import AWSIoTPythonSDK
#Importa la libreria de AWSIoTPythonSDK

sys.path.insert(0, os.path.dirname(AWSIoTPythonSDK.__file__))

from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient
#Importa la clase que implementa el cliente MQTT

from datetime import date, datetime
#Importa la clases de tiempo

import logging
import argparse
import json

import RPi.GPIO as GPIO
#Importa la libreria de GPIO para el control de puertos de I/O

import time, sys

FLOW_SENSOR = 27
#Define el puerto de entrada del sensor de flujo

GPIO.setmode(GPIO.BCM)
#Define numeracion BMC para los puertos

GPIO.setup(FLOW_SENSOR, GPIO.IN, pull_up_down = GPIO.PUD_UP)
#Configura el puerto 27 como de entrada

global count
#Vvariable count para el conteo de pulsos

count = 0
#Inicializa la variable en Cero (0)

# AWS IoT certificate based connection
myMQTTClient = AWSIoTMQTTClient("123afhlss456")
#Crear un nombre para el cliente MQTT

```

```

myMQTTClient.configureEndpoint
("a3cmxruztstldcp.iot.us-west-2.amazonaws.com", 8883)
#Punto de enlace de API REST

myMQTTClient.configureCredentials
("/home/pi/AWS_Certs/root_ca.pem",
"/home/pi/AWS_Certs/cloud-private.pem.key",
"/home/pi/AWS_Certs/cloud-certificate.pem.crt")
#Indica la ubicacion de los certificados digitales

myMQTTClient.configureOfflinePublishQueueing(-1)
# Define una cola de "Publish" infinita

myMQTTClient.configureDrainingFrequency(2) # 2 Hz
myMQTTClient.configureConnectDisconnectTimeout(10) # 10 sec
myMQTTClient.configureMQTTOperationTimeout(5) # 5 sec

#connect and publish
myMQTTClient.connect()
#Metodo de conexion del cliente MQTT hacia el Broker

myMQTTClient.publish("thing/info", "connected", 0)

def countPulse(channel):
#Metodo para el conteo de pulsos enviados por el sensor

    global count
    if start_counter == 1:
        count = count+1

GPIO.add_event_detect(FLOW_SENSOR, GPIO.FALLING,
callback=countPulse)
#Define la operacion del sensor

while True: #Metodo de jecucion infinito
    try:
        start_counter = 1
        time.sleep(1)
        now = datetime.utcnow()
        now_str = now.strftime('%Y-%m-%dT%H:%M:%SZ')
        #Metodo que establece el tiempo actual
        fluid = count * 2.6

```

```

#Calculo del volumen de liquido a partir del numero de pulsos
print "The flow is: %.3f ml" % (fluid)
#Imprime en pantalla el valor del volumen de liquido
payload = '{ "timestamp": ' + now_str +
', "flow": ' + str(fluid) + ' }'
#Se crea el payload o dato que con el tiempo actual
y el valor del volumen
myMQTTClient.publish("thing/data", payload, 0)
#Metodo que publica el dato en el topic o tema "thing/data"

except KeyboardInterrupt:
    #Se ejecuta si se present una interrupcion
    en la ejecucion del codigo
    print '\ncaught keyboard interrupt!, bye'
    GPIO.cleanup()
    sys.exit()          #Se cierra el programa

```

Del código mostrado anteriormente es importante resaltar que la línea que define el endpoint (`myMQTTClient.configureEndpoint`) se debe ajustar de acuerdo al valor de la sombra de objeto definida por HTTPS como una API REST como se muestra en la figura 39.



Figura 39: Endpoint de objeto en AWS.

## 6. RESULTADOS

Para ejecutar el código descrito en el punto anterior se debe ejecutar el siguiente comando en la terminal de la Raspberry:

```
$ python flowSensor.py
```

Una vez hecho esto, luego de realizar la demostración del paso de agua por el sensor de flujo (de acuerdo al sistema físico mostrado en la figura 40), los datos empiezan a ser enviados desde la Raspberry hacia la plataforma de AWS IoT (que actúa como el MQTT broker) y la cual a su vez enruta dichos mensajes al componente de Amazon ES, en donde por medio de su componente KIBANA es posible visualizar en tiempo real la variación del volumen de agua consumido. En la figura 41 se muestra el resultado de la medición luego de realizar la demostración.



Figura 40: Sistema físico de prueba.

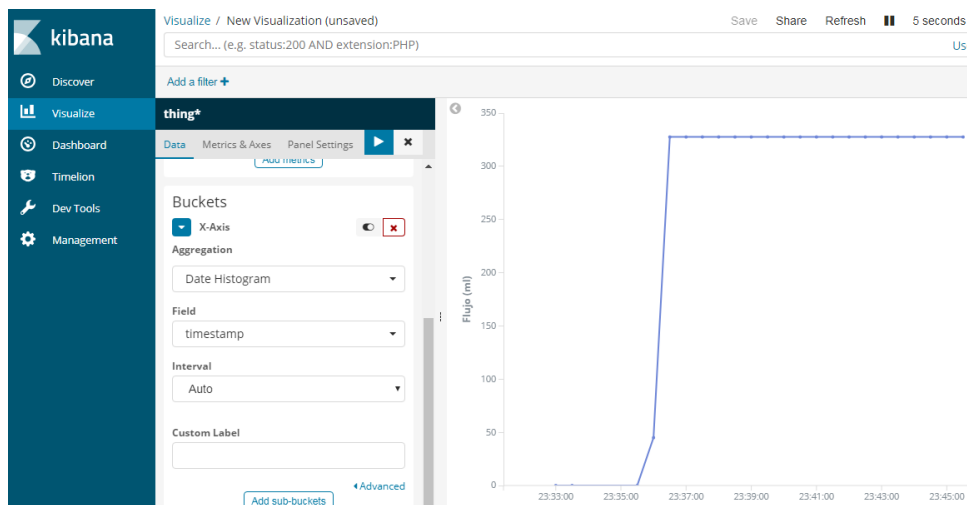


Figura 41: Resultados de medición en AWS KIBANA.

## 7. CONCLUSIONES

- De acuerdo al prototipo desarrollado, se corroboró que es posible aprovechar los componentes de AWS para conectar dispositivos a la nube y procesar datos de los mismos, teniendo en cuenta el paradigma de IoT.
- - Es importante conocer las características de hardware y la compatibilidad del dispositivo de IoT a conectar en la plataforma de AWS, pues no todos los desarrollos puede ser registrados. Un ejemplo es la tarjeta Arduino, cuyo modelo UNO no es compatible, debido a limitaciones de procesamiento que le impiden realizar el proceso de autenticación por medio de certificados. De acuerdo a consultas en la web, la tarjeta Arduino YUN cumple con los requerimientos que permiten la conexión a AWS. No obstante, el costo de dicha tarjeta es alto.
- - AWS ofrece un conjunto de módulos y elementos que poder ser utilizados para la implementación de soluciones de IoT, como es el caso del módulo de Elastic Search. No obstante, es importante, antes de utilizar dichas herramientas, considerar que muchas de estas generan costos de utilización que deben ser considerados.
- Con base en el prototipo desarrollado fue posible corroborar el funcionamiento del protocolo MQTT y su adecuada adaptabilidad a este tipo de soluciones.



## Referencias

- [1] Puertos GPIO - Raspberry Pi  
<https://www.raspberrypi.org/documentation/usage/gpio/README.md>
- [2] Raspberry Pi Based Liquid Flow Monitoring and Control  
[https://www.researchgate.net/publication/273297955\\_RASPBERRY\\_PI\\_BASED\\_LIQUID\\_FLOW\\_MONITORING\\_AND\\_CONTROL](https://www.researchgate.net/publication/273297955_RASPBERRY_PI_BASED_LIQUID_FLOW_MONITORING_AND_CONTROL)
- [3] Streaming Sensor Data (Raspberry Pi) to AWS IoT  
<http://techblog.calvinboey.com/raspberrypi-aws-iot-python/>
- [4] Analyze device-generated data with AWS IoT and Amazon Elasticsearch Service  
<https://aws.amazon.com/es/blogs/mobile>
- [5] Elasticsearch Service  
<https://www.elastic.co/guide/en/kibana/6.x/tutorial-define-index.html>
- [6] MQTT Essentials  
<https://www.hivemq.com/mqtt-essentials/>