Hub Of Things: Concentrador para Internet de las Cosas

Ricardo Brea[1], Daniel Skrie[1], Marisa Panizzi[1], Rodolfo Bertone[2]

[1] Escuela de Sistemas. Universidad Argentina John F. Kennedy.
Bartolomé Mitre 1411, Ciudad Autónoma de Buenos Aires (C1037ABA), Argentina
[2] Instituto de Investigaciones en Informática LIDI. Facultad de Informática. UNLP – CIC
brea.ricardo@gmail.com; dskrie@yahoo.com.ar; marisapanizzi@outlook.com;
rbertone@lidi.unlp.edu.ar

Resumen. Desde hace algunos años se observa un notorio incremento en la cantidad de dispositivos electrónicos conectados a internet, monitoreados y controlados en forma remota. La diversidad tecnológica, por un lado, y la cantidad de dispositivos, por otro, dificulta su integración o, al menos, una integración ordenada. Además, las plataformas basadas en microcontroladores o de procesamiento reducido, como Arduino, no brindan una conexión con niveles aceptables de seguridad. La privacidad constituye otra dificultad, ya que los usuarios desconocen si los distintos proveedores de soluciones IoT (Internet of Things) utilizan sus datos o los venden a terceros. La latencia también es otro factor determinante al diseñar una solución IoT, muchos de los proveedores de soluciones en la nube no tienen servidores locales, lo que degrada el tiempo de reacción ante determinado evento. El Hub Of Things (*HoT)* es una solución que permite integrar localmente los dispositivos *IoT*, brindando una alternativa para solucionar los problemas de diversidad de tecnologías, con una interface homogénea y segura. Facilitando la mediación entre los dispositivos y el usuario.

Palabras Clave: Internet de las Cosas, Fog computing, Seguridad, MQTT.

1 Introducción

El término “Internet de las Cosas” o IoT por sus siglas en inglés, es un concepto acuñado en 1999 por Kevin Ashton investigador de MIT [1]. Originalmente los datos disponibles en internet eran agregados o generados por humanos. Con la incorporación de sensores y conectividad más accesible, los dispositivos o “cosas” son capaces de generar y almacenar datos. Los dispositivos pueden ser diversos teléfonos inteligentes, bandas inteligentes (*SmartBands*), estaciones meteorológicas, automóviles, etc.

El IoT genera grandes posibilidades de crecimiento no solo en el ámbito hogareño sino también en el ámbito industrial, con lo que se denomina Industria 4.0; por ejemplo en la planificación urbana con el concepto de ciudad inteligente o Smart Cities.

En la Tabla 1, se representan las distintas proyecciones de crecimiento del IoT, son diversas, pero promedian 25,7 billones de dispositivos para 2020.

Tabla 1. Pronósticos de crecimiento del IoT

Fuente	Billones de dispositivos
Cisco [2]	**26.3**
Ericsson [3]	**28**
Gartner [4]	**20.8**
Goldman Sachs [5]	**28**

Con un pronóstico de crecimiento tan optimista, es importante detectar que problemas tiene el IoT en la actualidad con el propósito de mejorarlos. *Acquity Group* (División de *Accenture* dedicada a la estrategia digital y marketing) realizó en el año 2014 una encuesta a 2000 consumidores de los Estados Unidos, que conociendo los dispositivos *IoT*, no los adquirirían por los motivos que se presentan en la Figura 1.

Figura 1. Motivos de Rechazo del IoT

La falta de valor agregado es una falla que se puede deber a varios motivos, uno de esos motivos es la difícil integración con soluciones similares o existentes.

La preocupación por la privacidad es un elemento inherente a las soluciones en la nube. Los proveedores de servicios en la nube mitigan este problema con un contrato de privacidad.

La preocupación por los costos se presenta junto con varios elementos, desconocimiento de las tecnologías e incertidumbre en los costos de servicios necesarios para implementar las soluciones y que forman parte del costo fijo.

Hay aspectos técnicos que impiden la implementación del IoT, como la latencia o el tiempo de respuesta de la solución. Cuando se opta por tener una solución en la nube hay numerosos factores que inciden en la latencia desde que el dato es generado en el dispositivo hasta que se trasmite al proveedor del servicio. Estos factores pueden ser la calidad de la conexión a internet o la distancia entre el dispositivo y el centro de datos del proveedor. Para reducir la latencia inherente de las soluciones en la nube surge el concepto de *Fog Computing* [1] o *Edge Computing* que se refiere a:

- Analizar los datos más sensitivos en el *Edge*, el borde; donde se generan los datos, sin necesidad de enviar grandes volúmenes de datos a la red.
- Reaccionar a la información generada en el rango de milisegundos.
- Enviar la información a la nube para su análisis y almacenamiento a largo plazo.

2 Desarrollo

El objetivo del ***Hub Of Things*** es proveer una plataforma segura, homogénea, extensible y abierta, que opera en el *Fog Computing* y puede ser utilizada de base en nuevas soluciones *IoT* para el control y monitoreo de dispositivos conectados a internet. El ***Hub Of Things*** plantea una solución de bajo costo a los motivos de rechazo al IoT.

En su esencia el ***Hub Of Things*** es un IoT *Gateway*, una solución de software y hardware que sirve de intermediario entre los dispositivos y otros sistemas, ya sea que estén implementados localmente (*On-Premise*) o en la Nube. HoT incrementa la seguridad, la estabilidad y reduce la latencia Además, al brindar una capa de abstracción reduce el acoplamiento con servicios *IoT* en la nube y permite adaptar otros protocolos de comunicación como *LoRa*, *BLE*, *Modbus, etc.* a un único protocolo, en este caso *MQTT*. En la Figura 2**,** se presenta la arquitectura interna del sistema y la relación entre sus componentes.

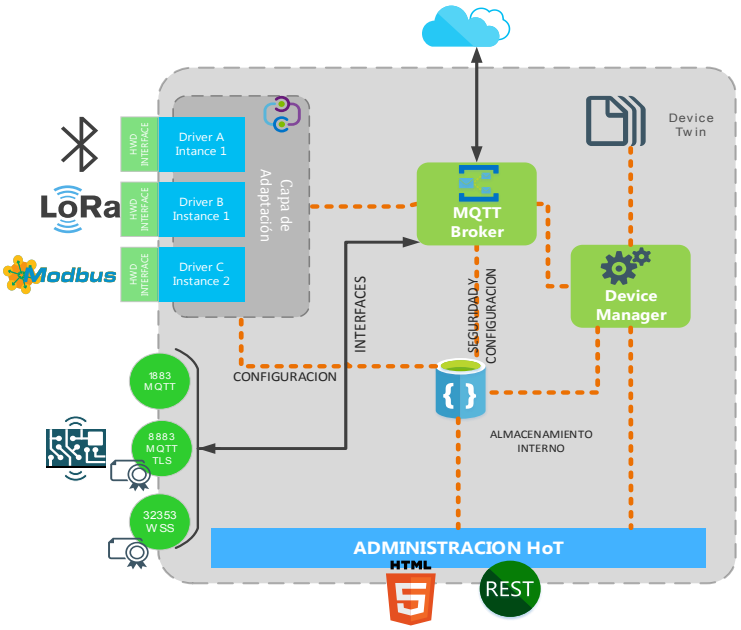

Figura 2. Arquitectura interna del HoT

- **Capa de adaptación**: el objetivo de esta capa es proveer una interface de programación, ***API***, que permita adaptar cualquier protocolo al sistema de mensajes similar a ***MQTT***, es decir en tuplas *<<TOPIC, PAYLOAD>>*.
- **MQTT Broker**: es el componente central de la solución permitiendo conectar dispositivos y clientes en forma homogénea. Implementa el protocolo MQTT sobre TCP y *WebSockets* lo que facilita la comunicación con clientes JavaScript.
- **Device Manager**: el objetivo del *Device Manager* es mantener el estado de los dispositivos utilizando el concepto de Gemelo Digital o *Digital Twins*.
- **API de dispositivos:** el HoT provee una API REST que permite administrar la definición de los dispositivos.
- **Interface HTML**: la capa de presentación se encuentra programada en *HTML5* y *JavaScript* utilizando el *framework Angular* para aplicaciones de una sola página o "*SPA*". El diseño de la página seguirá la directiva *Material Desing* de Google [2], que se asemeja a los controles del sistema operativo *Android*. Dichas directivas existen para diseñar interfaces de usuario intuitivas, homogéneas y fáciles de utilizar. Otras características de la interface HTML son; Interface *Responsive*, adaptable a varios tipos de dispositivos; Gráficos SVG, vectoriales que consumen menos espacio y son visibles en cualquier resolución [3].

2.1 MQTT.

(*Message Queue Telemetry Transport*) o MQTT un protocolo usado para la comunicación *machine-to-machine* (*M2M*) en *IoT*. Este protocolo está orientado a la comunicación de sensores y esta optimizado para no agregarle mucha información adicional a cada mensaje que envía y puede ser utilizado en la mayoría de los dispositivos embebidos y de pocos recursos. *MQTT* sigue una topología de estrella, con un nodo central que hace de servidor o *broker* y distribuye los mensajes utilizando el patrón *publiser-subscriber* [4].

Cada mensaje MQTT se compone de un tópico o *topic* y de una carga o *payload*. El tópico identifica el origen de los datos y la carga almacena el o los valores. El formato del tópico es de niveles separados por el carácter *'/'*. En la Figura 3**,** se muestra la interacción de los distintos actores dentro de la arquitectura del protocolo MQTT.

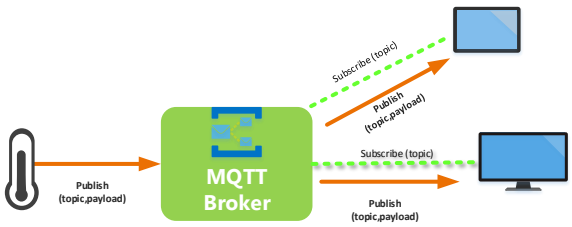

Figura 3. Arquitectura MQTT

2.2 Organización de los dispositivos

El HoT provee una estructura para organizar los dispositivos a fin de complementar al protocolo MQTT. El sistema almacena una definición por cada dispositivo conectado a fin de aplicar el concepto de *Digital Twin*, almacenando metadatos y estado actual del dispositivo. Este concepto es reciente y varias plataformas lo denominan de diferente manera. *AWS* lo denomina *Device Shadow*, *Azure* lo denomina *Device Twin*.

HoT utiliza la naturaleza jerárquica de los tópicos de MQTT para agrupar los dispositivos y sus sensores.

La definición de un dispositivo en HoT tiene los siguientes elementos:

- Información esencial:
  - Identificador de dispositivo: identificador único de dispositivo, dicho identificador se utiliza para autenticar los dispositivos en el sistema.
  - Secreto: se utiliza para autenticar al dispositivo
- Metadatos: información complementaria del dispositivo utilizada para su visualización como puede ser nombre, ícono y ubicación.
- Estado: utilizado por el *Device Manager* para monitorear los dispositivos:
  - Tópico testigo: tópico de MQTT al cual el *Device Manager* se subscribe para actualizar los demás atributos del estado
  - Fecha hora de último reporte
  - Tiempo máximo de inactividad
  - Activo
- Atributos: valores enviados desde los sensores conectados al dispositivo, dichos valores también pueden tener una estructura jerárquica.
- Comandos: valores enviados al dispositivo

2.3 Seguridad

La seguridad es un elemento fundamental del sistema y cubre los siguientes aspectos básicos:

- Recursos: se denomina recurso a los elementos resguardados por la seguridad. En este caso las interfaces REST, HTML y MQTT
- Autenticación: identificar los usuarios que se conectan al sistema
- Autorización: nivel de acceso al sistema de un usuario autenticado
- Intercambio seguro de datos: el sistema encripta los datos transmitidos desde y hacia los usuarios.

Autenticación: cualquier usuario que desee conectarse a alguna de las interfaces debe estar autenticado. Las contraseñas se validan contra la colección de usuarios.

Los dispositivos que se conecten mediante alguna de las interfaces MQTT también deben especificar credenciales. Dichas credenciales son el id de dispositivo y su secreto.

Autorización: la autorización se realiza utilizando roles en el caso de las interfaces HTML Y REST y una lista de control de acceso o ACL para la interface MQTT.

El sistema permite definir usuarios con un nivel de autorización *Administrador* o V*isualizador* para las interfaces HTML y REST. El nivel de acceso a la interface MQTT acceso utilizando una lista de control de acceso o ACL (*Access Control List*).

Cada registro de la ACL tendrá como atributos:

- Usuario: identificador de usuario
- Tópico: cadena representado el tópico al cual el usuario puede acceder, se puede hacer uso de los comodines admitidos por MQTT**.**
- Tipo de acceso: LECTURA | ESCRITURA | COMPLETO

En el caso de los dispositivos la autorización, se realiza de una forma más simple, sin necesidad de una ACL. Cada dispositivo puede publicar y subscribirse a cualquier tópico cuyo prefijo coincida con la definición del dispositivo.

Intercambio de datos: El sistema utiliza la tecnología TLS*, **T**ransport **L**ayer **S**ecurity*, seguridad en la capa de transporte, que consiste en encriptar los datos de la conexión segura por medio de un certificado emitido para tal fin. El intercambio de datos las interfaces (MQTT, HTML y REST) se encuentran asegurados por este método.

3 Caso de validación

A fin de comprobar el funcionamiento de la solución se han desarrollado dos dispositivos que son integrados a través del HoT. En esta implementación se abordan temas fundamentales de IoT como es la persistencia de los datos a largo plazo, la visualización de los datos almacenados y la arquitectura de implementación.

La implementación tiene tres elementos a destacar:

1. **Recolección de datos ambientales:** su función es la medición de variables ambientales de una planta (temperatura ambiente, humedad relativa del ambiente, intensidad lumínica y humedad de suelo).
2. **Cámara NDVI**: desarrollado sobre la plataforma Raspberry Pi Zero W y utiliza análisis de imagen para calcular el índice de vegetación diferencial normalizado o NDVI [5].
3. **HoT Host:** computadora que tendrá instalado todos los elementos de software correspondientes al HoT. En la Tabla 2, se presenta el detalle del hardware utilizado incluyendo los sensores conectados.

La solución se implementa utilizando la arquitectura de microservicios por la facilidad de actualización de cada uno de sus componentes; y con leves modificaciones puede ser implementado en diversas plataformas y sistemas operativos. *Docker* es una plataforma abierta para micro servicios elegida para esta implementación. Cabe destacar que la solución puede ser implementada prescindiendo de *Docker,* pero puede resultar más trabajoso dado que se debe configurar cada elemento manualmente.

Tabla 2. Detalle del hardware de la solución

Función	**Hardware**
RECOLECCION DE DATOS AMBIENTALES	ESP 32: Microcontrolador
	DHT22: Sensor de temperatura y humedad ambiente
	DS18B20: Sensor de temperatura de suelo
	BH1750: Sensor de intensidad lumínica
	Sensor de humedad de suelo capacitivo
CAMARA NDVI	Raspberry PI Zero W
	Pi NoIR: Cámara infrarroja
	Filtro azul para procesar el NIR (Near Infra red)
HoT Host	Rapsberry PI 3

La Figura 4, presenta un diagrama en bloques de los distintos componentes, incluidos los elementos de hardware.

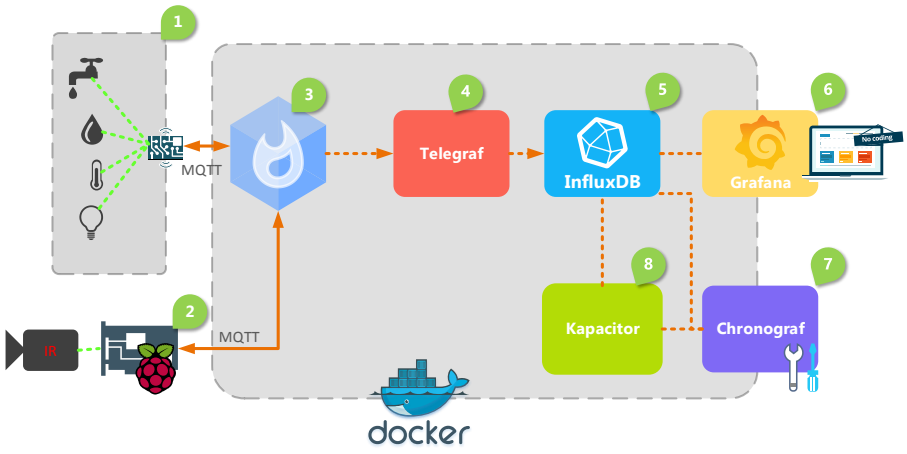

Figura 4. Arquitectura de la solución propuesta

La solución posee varios contenedores, que son instancias de una imagen. Una imagen es un conjunto de archivos inmutable que proporcionan un software pre-instalado[6]. La interrelación de los contenedores, representada en la figura 4, realizan el siguiente proceso:

1. Recolección de datos ambientales: sus sensores capturan las variables ambientales y el micro controlador las procesa y envían los valores vía MQTT al HoT.
2. Cámara *NDVI*: Captura imágenes en intervalos regulares, calcula el *NDVI* y envía el valor promedio vía *MQTT* al *HoT*.
3. *Hub Of Thing*: es un conjunto de componentes, como se describe en la **Figura 3.**

   a. Capa de adaptación: Adaptación de otros protocolos a formato de mensajes *MQTT*
   b. MQTT Bróker: Bróker *MQTT* conectado a la seguridad de HoT
   c. Interface HTM: Interface que permite la ver y crear dispositivos, monitorear sus valores y enviar comandos.
   d. REST API: interface que permite administrar la definición de dispositivos
   e. *Identity Server*: Servicio de validación de usuarios
   f. MongoDB: Base de persistencia de datos NoSQL
4. *Telegraf*: agente de recolección y persistencia de datos, se conecta al *MQTT* Bróker y persiste los valores numéricos en *InfluxDB*
5. *InfluxDB*: base de datos de series de tiempo para persistencia de datos a largo plazo.
6. *Grafana*: plataforma web para la visualización de datos, se conecta a InfluxDB y permite realizar gráficos personalizados con la información.
7. *Cronograf*: interface de administración de *InfluxDB* y *Kapacitor*, facilita la configuración de ambos elementos mediante una interface web.
8. *Kapacitor*: agente de procesamiento de datos en línea. Permite crear alerta de los valores transmitidos a *InlfuxDB*

3.1 Persistencia y visualización de datos

En los sistemas IoT los sensores, conectados a los dispositivos, generan valores en un momento determinado, es decir que la información generada tiene el formato [sensor, tiempo, valor]. Las bases de datos de serie de tiempo o *TSDB*, están diseñadas especialmente para manejar eventos y métricas combinadas con el tiempo.

Originalmente desarrolladas para analizar información financiera como la evolución de precios de las acciones, las bases de datos de serie de tiempo están encontrando un nuevo uso en el IoT. A diferencia de otro tipo de bases de datos, las *TSDB,* incluyen almacenamiento y compresión de datos con sello de tiempo, administración del ciclo de vida de los datos, funciones de agrupación, capacidad de manejar grandes escaneos dependientes de series de tiempo de muchos registros y consultas de series de tiempo.

3.2 Retención y reducción de muestras

Los sensores conectados a IoT pueden generar una gran cantidad de datos y con una implementación a mediano o largo plazo el espacio de almacenamiento es un punto de importancia a tener en cuenta. A fin de no causar una falla de sistema por falta de almacenamiento es necesario ajustar las políticas de retención y reducción de datos.

La política de retención permite poner un límite a la antigüedad de los datos. Una vez superado ese umbral de tiempo los datos son borrados.

Eliminar los datos "antiguos" no siempre es posible, en esos casos se puede optar por una técnica de reducción de muestras o *Downsampling*, Esta técnica consiste en agrupar todas las muestras en intervalos regulares (por ejemplo 30 minutos, una hora, o una instante de tiempo que se defina) y aplicar alguna función de agregación como puede ser suma, promedio, razón, etc., lo cual reduce ostensiblemente la cantidad de información almacenada.

El método de reducción de muestras puede ser aplicado en *InfluxDb* utilizando la política de retención de datos en la métrica donde impactan los valores de los sensores para que expire al transcurrir un cierto periodo de tiempo, por ejemplo, un día. Luego se programa una consulta continua (*Continuous Query*) que se ejecuta cada 24 horas y calcula la razón de las muestras de cada sensor agrupadas en intervalos de una hora y persistida.

4 Conclusiones y futuras líneas de investigación y desarrollo

El ***Hub Of Things*** es una herramienta que propone una base o punto de inicio a otras soluciones *IoT* a problemas particulares o como integrador para soluciones protocolos no *TCP/IP*.

La experiencia de usuario se extiende más allá del software y el tratamiento de los datos. El desarrollo de dispositivos de borde no es trivial ni sencillo. Si bien escapa al alcance de este trabajo, el dispositivo de borde es la "cosa" en la internet de las cosas.

La carencia de valor agregado o un funcionamiento deficiente no se puede subsanar con funcionalidad adicional.

La utilización de estadística para la reducción de datos es esencial para que la información importante no se pierda en el proceso.

El uso de herramientas de ciencia de datos también es un elemento que se debe considerar a fin para expandir el análisis de datos de la solución.

Docker es una gran herramienta a tener en cuenta, ideal para orquestar una solución que puede ser implementada tanto en la nube como localmente y que facilita la actualización.

Las futuras líneas de investigación y desarrollo se verán orientadas a la recopilación de datos y la integración de sistemas. Cada una de ellas, puede ser un trabajo independiente, pero se mencionan por que utilizarán el ***Hub Of Things*** como elemento central. Estas pueden ser:

- Medición de consumo eléctrico hogareño.
- Monitoreo de variables ambientales urbanas como la concentración de CO_2 o el nivel de ruido ambiente.
- Integración de sistemas inalámbricos *IoT* de largo alcance con *LoRa*.
- Integración de sistema cableados *Modbus*.

5 Referencias

[1] Cisco, «Fog Computing and the Internet of Things,» 2015. [En línea]. Available: https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf. [Último acceso: FEB 2016].

[2] Google, «Material Design - Guidelines,» 2014. [En línea]. Available: https://material.io/guidelines/. [Último acceso: NOV 2016].

[3] D. Strazzullo, D. D. Dailey and J. Frost, Building Web Applications with SVG, Sebastopol, California: O'Reilly Media, Inc., 2012.

[4] OASIS, «MQTT Version 3.1.1,» OCT 29 2014. [En línea]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html. [Último acceso: ENE 2016].

[5] J. Weier y D. Herring, «NASA-Earthobservatory-Measuring Vegetation (NDVI & EVI),» 30 AUG 2000. [En línea]. Available: https://earthobservatory.nasa.gov/Features/MeasuringVegetation/measuring_vegetation_1.php. [Último acceso: JUN 2017].

[6] Docker Inc., «Docker glossary,» Docker Inc., [En línea]. Available: https://docs.docker.com/glossary/. [Último acceso: JUN 2018].

[7] InfluxData, Inc, «Open Source Time Series Platform,» InfluxData, Inc, 2018. [En línea]. Available: https://www.influxdata.com/time-series-platform/. [Último acceso: MAY 2018].

[8] Grafana Labs, «Grafana: testimonials,» Grafana Labs, 2018. [En línea]. Available: https://grafana.com/grafana/testimonials. [Último acceso: 25 MAY 2018].

[9] Cisco, «Cisco Visual Networking Index Predicts Near-Tripling of IP Traffic by 2020,» 07 JUN 2016. [En línea]. Available: https://newsroom.cisco.com/press-release-content?type=press-release&articleId=1771211. [Último acceso: FEB 2018].

[10] Ericsson, «Ericsson Mobility Report November 2015,» NOV 2015. [En línea]. Available: https://www.ericsson.com/assets/local/news/2016/03/ericsson-mobility-report-nov-2015.pdf. [Último acceso: FEB 2018].

[11] Gartner, «Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016,» 10 NOV 2015. [En línea]. Available: https://www.gartner.com/newsroom/id/3165317. [Último acceso: FEB 2018].

[12] S. Jankows, J. Covello, H. Bellini, J. Ritchie y D. Costa, «The Internet of Things: Making sense of the next mega-trend,» 3 SEP 2014. [En línea]. Available: http://www.goldmansachs.com/our-thinking/outlook/internet-of-things/iot-report.pdf. [Último acceso: FEB 2018].