Demostrador de Internet of Things con la tecnología IEEE 802.15.4e utilizando la plataforma OpenMote y el sistema operativo OpenWSN y theThings.io

Francisco Mudarra Capdepont

Máster Universitario En Ingeniería de Telecomunicación (UOC-URL)

Junio 2018

Índice

Objetivos y motivaciones

- Desarrollar un demostrador de una red IoT basado en la plataforma OpenWSN, utilizando el estándar IEEE 802.15.4e, sobre una Raspberry Pi que permita la gestión doméstica de una vivienda, almacenando los datos en local para su posterior subida a la plataforma theThings.iO en la nube y su representación.

- Evaluación de los resultados obtenidos para comprar los resultados de aplicar Clould Computing o Edge Computing.

Objetivos y motivaciones

- Estudiar los diferentes estándares para sistemas IoT.
- Comprender y utilizar la herramienta OpenWSN.
- Desarrollo de una aplicación en Python para el control de la red.
- Subida de los datos resumidos a la nube y su representación en la plataforma theThings.iO

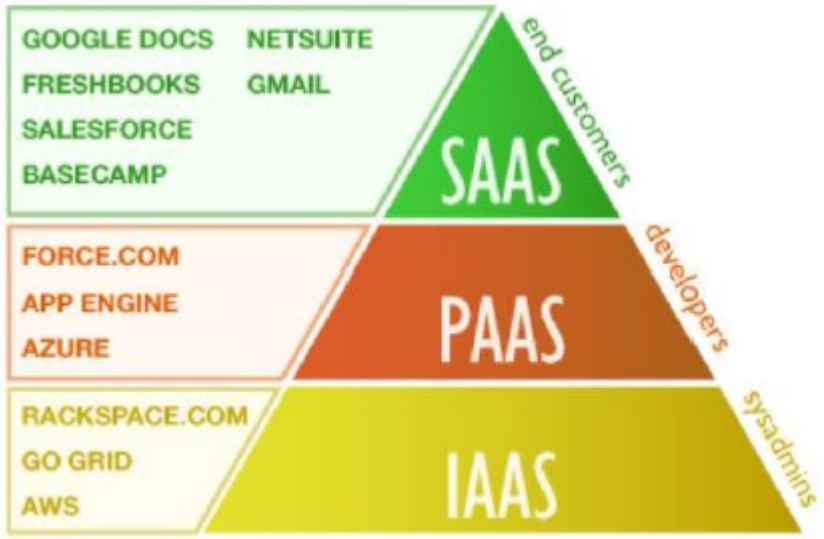
Internet of Things (IoT)

- IoT nace de la necesidad de conectar cualquier dispositivo a internet .
- Se puede definir como: La interconexión digital de objetos cotidianos a internet.
- En los últimos años se ha convertido en una de las tendencias mas relevantes del sector.

Cloud Computing

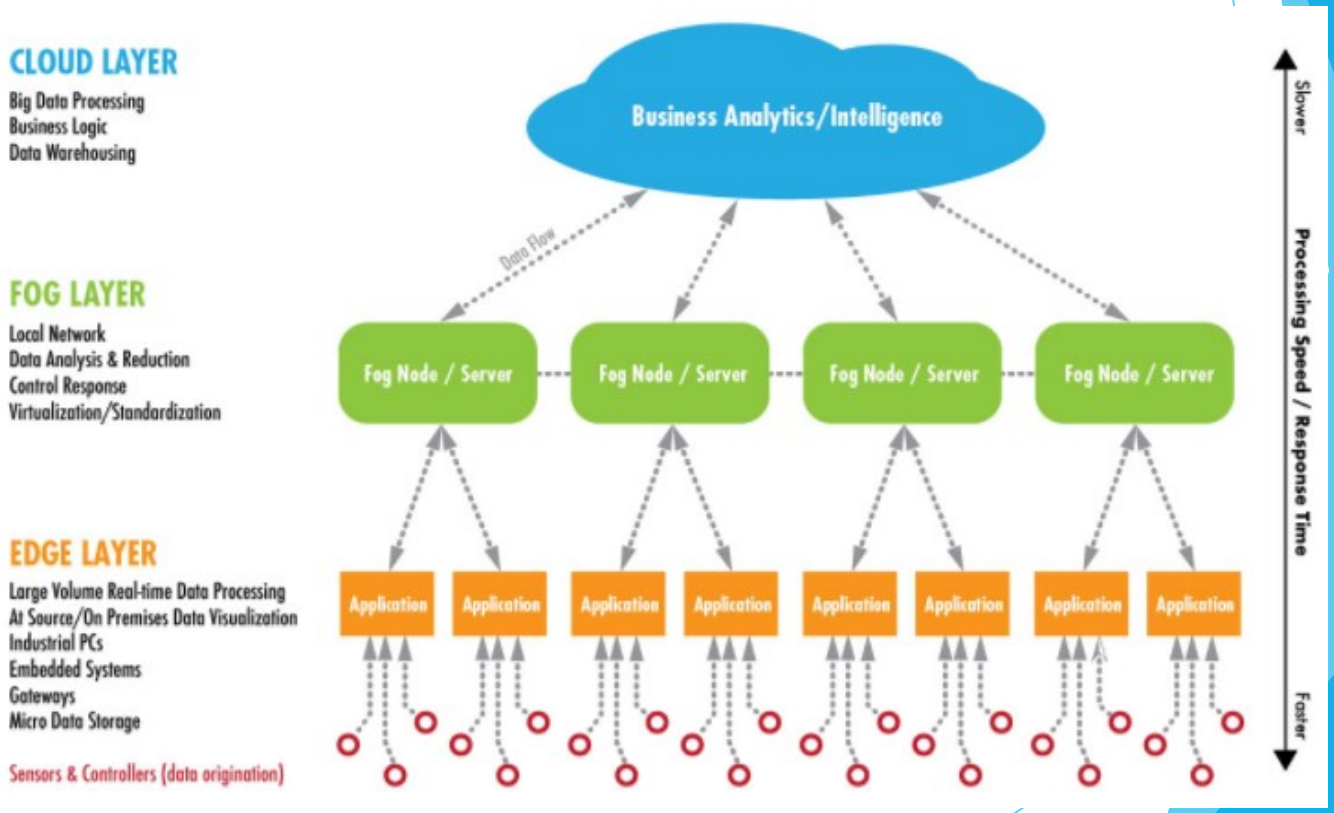
- Cloud computing nace de la necesidad de ofrecer servicios a través de Internet. La computación en la nube ofrece el acceso a recursos de software bajo demanda. Abstrayendo al usuario de esta gestión.

- Uso bajo demanda
- Escalabilidad
- Seguridad
- Sin mantenimiento
- Reducción de costes



Edge y Fog Computing

- Edge computing: Son los dispositivos que se encuentra en el borde de la red.
- Fog Computing: Es una infraestructura intermedia para la administración de los dispositivos para su uso mas eficiente.



Edge y Fog Computing

- Objetivo:
  - Reducir el retardo por procesamiento
  - Optimizar el uso del ancho de banda
  - Mejorar la seguridad
- Todo esto se consigue llevando el parte del procesado mas cerca de los dispositivos

Edge y Fog Computing

	Cloud Computing	Fog Computing
Latencia	Alta	Baja
Retardo Jitter	Alta	Muy bajo
Localización del servicio	Dentro de internet	En el borde de las redes locales
Distancia entre el cliente y el servidor	Múltiples saltos	Un salto
Seguridad	Indefinida	Se puede definir
Posibilidad de ataques	Alta	Baja
Localización definida	No	Si
Geo distribución	Centralizada	Distribuida
Número de nodos	Pocos	Muchos
Soporte para la movilidad	Limitada	Soportada
Iteraciones en tiempo real	Soportada	Soportada
Conexión última milla	Línea alquilada	Wireless

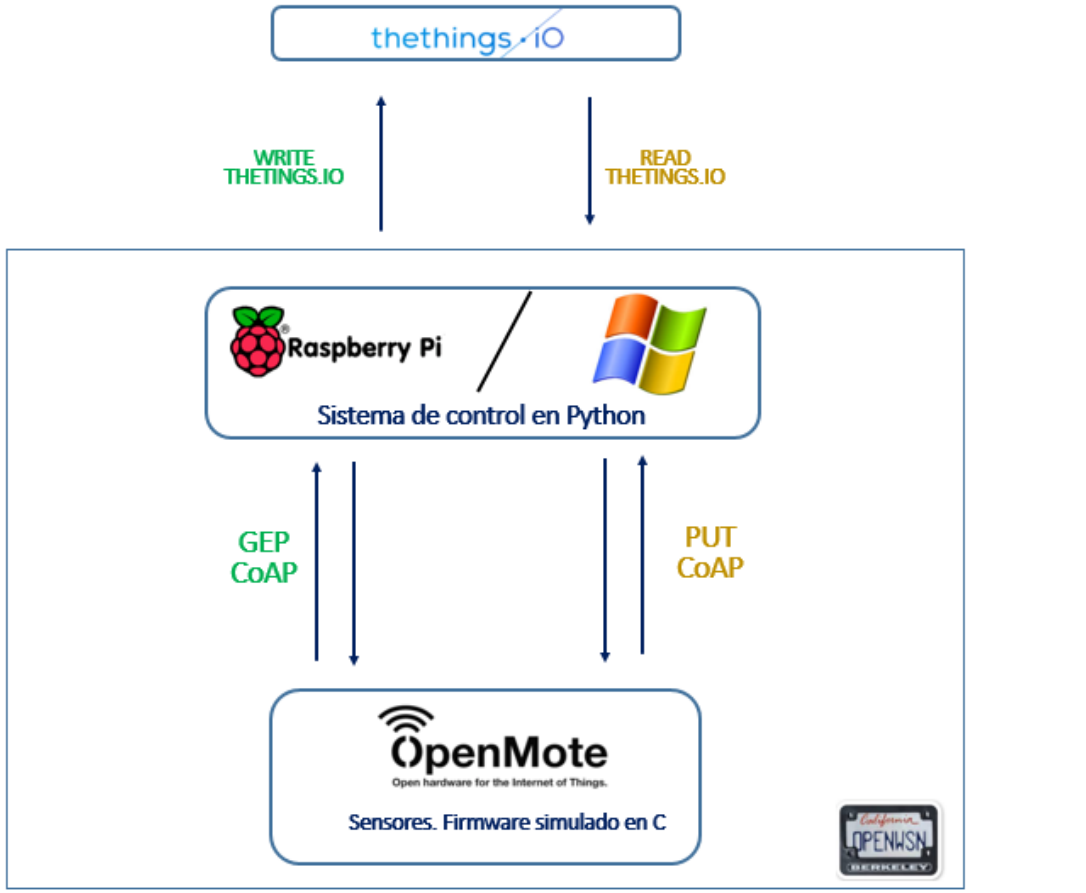
OpenWSN

- OpenWSN es una plataforma de código abierto que proporciona una serie de protocolos basados en el Internet de las Cosas (IoT).
- Es versátil, intuitiva y fácil de manejar e implementar.
- Pila de protocolos de OpenWSN:

Application	CoAP, HTTP
Transport	UDP, TCP
IP/routing	IETF RPL
Adaptation	IETF 6LoWPAN
Médium Access	IEEE802.15.4e
Phy	IEEE802.15.4-2006

Desarrollo software

- El desarrollo software consta de 3 partes:
  - Plataforma Cloud: thethings.iO
  - Programa principal en Python
  - Emulador sensores en C



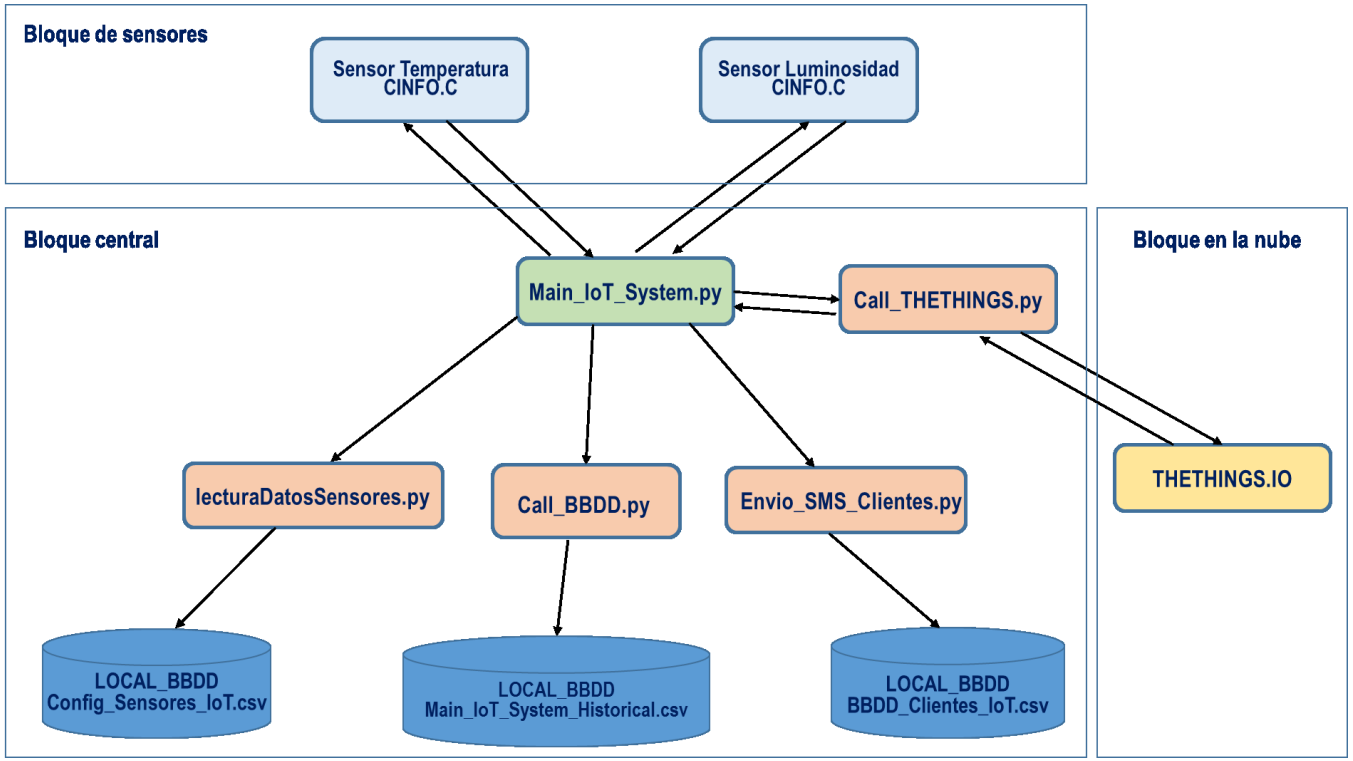
Emulador de sensores

- Los sensores han sido generados de forma emulada, modificando el firmware de OpenWSN.
- La comunicación se realiza por medio del protocolo CoAP
- CoAP: c.get(), c.put()
- Generación de datos pseudoaleatorios

Temperatura	cambio lectura									
Hora	0	1	2	3	4	5	6	7	8	9
0:00	16	17	18	19	20	21	22	23	24	25
1:00	14	15	16	17	18	19	20	21	22	23
2:00	13	14	15	16	17	18	19	20	21	22
3:00	12	13	14	15	16	17	18	19	20	21
4:00	12	13	14	15	16	17	18	19	20	21
5:00	11	12	13	14	15	16	17	18	19	20
6:00	11	12	13	14	15	16	17	18	19	20
7:00	11	12	13	14	15	16	17	18	19	20
8:00	11	12	13	14	15	16	17	18	19	20
9:00	12	13	14	15	16	17	18	19	20	21
10:00	13	14	15	16	17	18	19	20	21	22
11:00	14	15	16	17	18	19	20	21	22	23
12:00	15	16	17	18	19	20	21	22	23	24
13:00	16	17	18	19	20	21	22	23	24	25
14:00	17	18	19	20	21	22	23	24	25	26
15:00	18	19	20	21	22	23	24	25	26	27
16:00	19	20	21	22	23	24	25	26	27	28
17:00	20	21	22	23	24	25	26	27	28	29
18:00	19	20	21	22	23	24	25	26	27	28
19:00	18	19	20	21	22	23	24	25	26	27
20:00	17	18	19	20	21	22	23	24	25	26
21:00	16	17	18	19	20	21	22	23	24	25
22:00	15	16	17	18	19	20	21	22	23	24
23:00	15	16	17	18	19	20	21	22	23	24

Programa principal

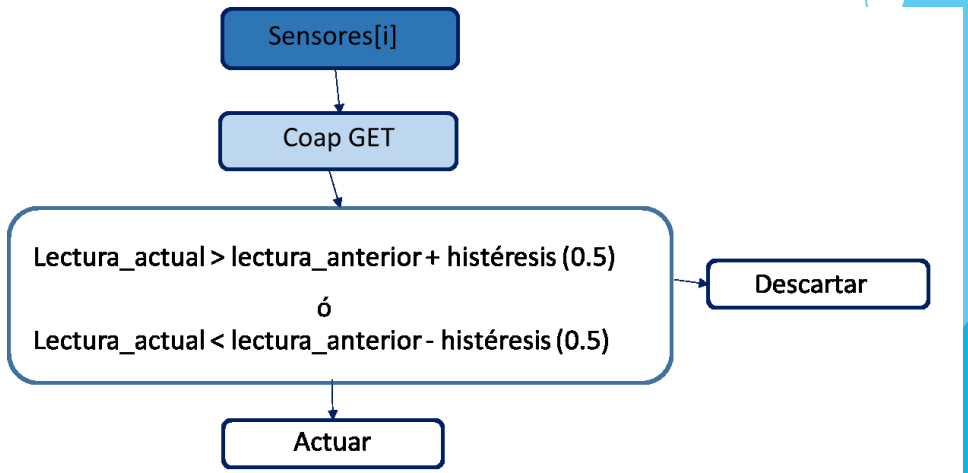
- Bloque central desarrolla en Python se en carga de la comunicación con los diferentes módulo y del procesado de información.



Programa principal

- Comunicación con el bloque de sensores mediante el protocolo CoAP.
- Se determinará si la lectura realizada es una lectura relevante o no.
- Para ello se comparará la lectura actual con la última lectura.

- Se aplicará un histéresis predeterminada en función de las necesidades de la aplicación.
- Con lo que se consigue una reducción del tráfico generado.

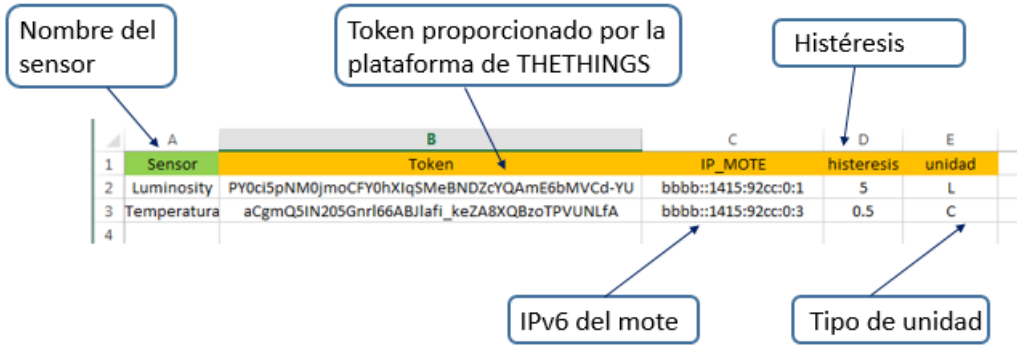


Programa principal

- Generación de una base de datos para la gestión de la información

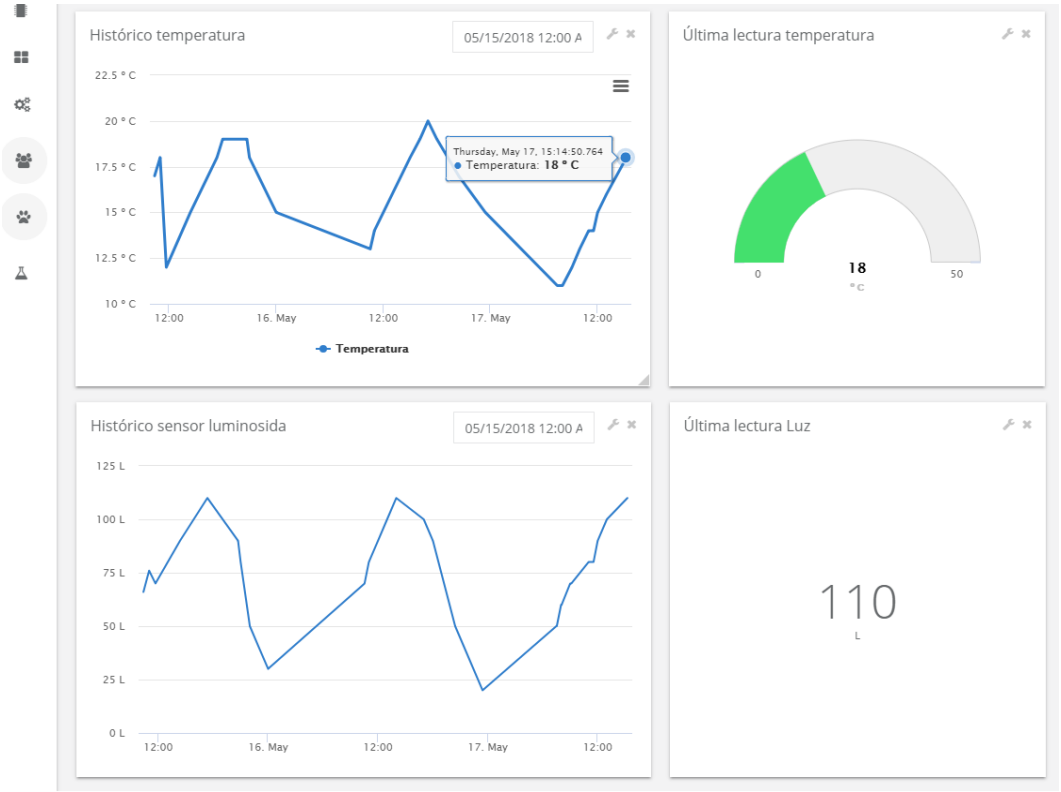
	A	B	C	D	E
1	Fecha	*	Sensor	*	Lectur
2	01/05/2018 9:34	El sensor:	Luminosity	registra:	70
3	01/05/2018 9:34	El sensor:	Temperatura	registra:	12
4	01/05/2018 9:35	El sensor:	Luminosity	registra:	77
5	01/05/2018 9:35	El sensor:	Temperatura	registra:	19
6	01/05/2018 9:37	El sensor:	Luminosity	registra:	77
7	01/05/2018 9:40	El sensor:	Luminosity	registra:	77
8	01/05/2018 9:40	El sensor:	Temperatura	registra:	19
9	01/05/2018 9:45	El sensor:	Luminosity	registra:	77
10	01/05/2018 9:45	El sensor:	Temperatura	registra:	19
11	01/05/2018 9:53	El sensor:	Luminosity	registra:	77
12	01/05/2018 10:00	El sensor:	Luminosity	registra:	77
13	01/05/2018 10:00	El sensor:	Temperatura	registra:	19
14	01/05/2018 10:01	El sensor:	Luminosity	registra:	77
15	01/05/2018 10:01	El sensor:	Temperatura	registra:	20
16	01/05/2018 10:10	El sensor:	Luminosity	registra:	77
17	01/05/2018 10:10	El sensor:	Temperatura	registra:	20
18	01/05/2018 10:15	El sensor:	Luminosity	registra:	77

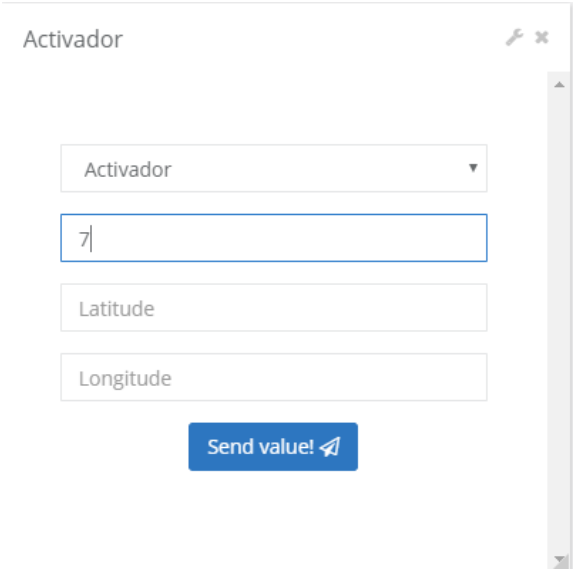
- Sistema versátil que permite añadir o eliminar sensores sin necesidad de modificar el código fuente.



	A	B	C	D	E
1	Sensor	Token	IP_MOTE	histeresis	unidad
2	Luminosity	PY0ci5pNM0jmoCFY0hXIqSMeBNDZcYQAmE6bMVCd-YU	bbbb::1415:92cc:0:1	5	L
3	Temperatura	aCgmQ5IN205Gnrl66ABJlafi_keZA8XQBzoTPVUNLfA	bbbb::1415:92cc:0:3	0.5	C
4					

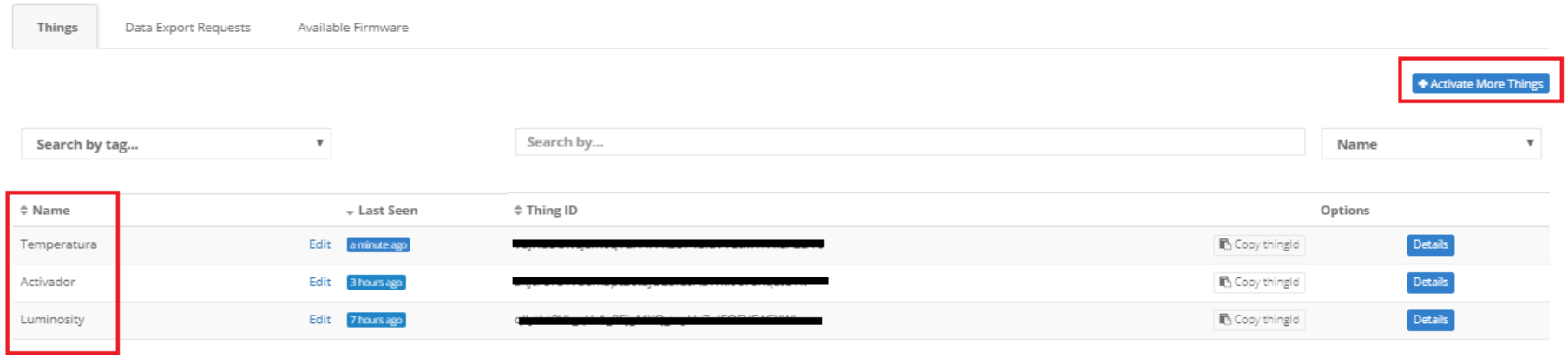
thethings.iO





thethings.iO

- Plataforma Cloud Computing para la representación de los datos e interacción con los dispositivos.
- API en Python.
- Activación de dispositivos por medio de Token
- Llamadas: addVar(), write(), read()



Resultados

- Se observa una gran cantidad de datos generados en la solución Cloud Computing.
- Al aplicar una solución Fog Computing, la reducción del tráfico y del número de llamadas es de un 99.95% .

	Cloud Computing					Fog Computing	
N° Sensores	1 segundo (b)	1 minuto (kb)	1 hora (kb)	1 día (Mb)	1 día (Gb)	40 llamadas API - (Mb)	Reducción de tráfico
1,00	196,00	11,48	689,06	16,15	0,02	0,01	99,95%
2,00	392,00	22,97	1378,13	32,30	0,03	0,01	99,95%
3,00	588,00	34,45	2067,19	48,45	0,05	0,02	99,95%
4,00	784,00	45,94	2756,25	64,60	0,06	0,03	99,95%
5,00	980,00	57,42	3445,31	80,75	0,08	0,04	99,95%
6,00	1176,00	68,91	4134,38	96,90	0,09	0,04	99,95%
7,00	1372,00	80,39	4823,44	113,05	0,11	0,05	99,95%
8,00	1568,00	91,88	5512,50	129,20	0,13	0,06	99,95%
9,00	1764,00	103,36	6201,56	145,35	0,14	0,07	99,95%
10,00	1960,00	114,84	6890,63	161,50	0,16	0,07	99,95%
100,00	19600,00	1148,44	68906,25	1614,99	1,58	0,75	99,95%
1000,00	196000,00	11484,38	689062,50	16149,90	15,77	7,48	99,95%
10000,00	1960000,00	114843,75	6890625,00	161499,02	157,71	74,77	99,95%
100000,00	19600000,00	1148437,50	68906250,00	1614990,23	1577,14	747,68	99,95%

Conclusiones

- Se ha conseguido desarrollar un demostrador de IoT sobre un sistemas OpenWSN, utilizando la plataforma de thethings.iO y con una Raspberry Pi 3.
- Se ha realizado un análisis detallados de todas las tecnologías y estándares necesarios para el desarrollo de un sistema IoT funcional.
- Se ha conseguido la elaboración de una base de datos para el tratamiento de la información.
- Se ha comprobado la importancia de utilizar Fog Computing junto con Cloud Computing y como el uso de la red aumenta exponencialmente con el número de dispositivos conectados a la red y la reducción de costes que conlleva.
- Se mejora la seguridad de los datos obtenidos al procesarlos localmente.

Gracias por su atención

- **Francisco Mudarra Capdepont**
- **fmudarra@uoc.edu**