



Contenido

INTRODUCCION	2
DISPOSITIVOS	4
EDGE INTELLIGENCE	5
FOG LAYER.....	5
ACTUALIZACION REMOTA	6
ENVIO DE DATOS COLECTADOS.....	6
ACTUACION	7
ESQUEMA FINAL	7
REDES DE COMUNICACIÓN	9
REQUISITOS TECNICOS.....	11
CONSUMO DE BATERIA.....	12
BIDIRECCIONALIDAD, THROUGHPUT DE DATOS Y LATENCIA	12
REDES PÚBLICAS Y PRIVADAS.....	13
PLATAFORMAS	14
NECESIDAD DE UNA PLATAFORMA IoT.....	14
APLICACIONES	14
EJEMPLO DE USO	16
PLATAFORMAS.....	16
ADAFRUIT	16
CAYENNE	23
UBIDOTS	29
UBIDOTS STEM.....	35
PUBLICACION EN UBIDOTS CON MOSQUITTO	36
PUBLICACION EN UBIDOTS CON MQTT.fx.....	40
REFERENCIAS.....	45



INTRODUCCION

Tecnológicamente, la revolución introducida por las tecnologías IoT nos permiten conectar a Internet (o a redes privadas) cualquier objeto, eléctrico o no que pueda ser adosado de inteligencia electrónica, que funcionaba de manera autónoma, sin interacción con el resto del mundo más allá de la física y netamente manual.

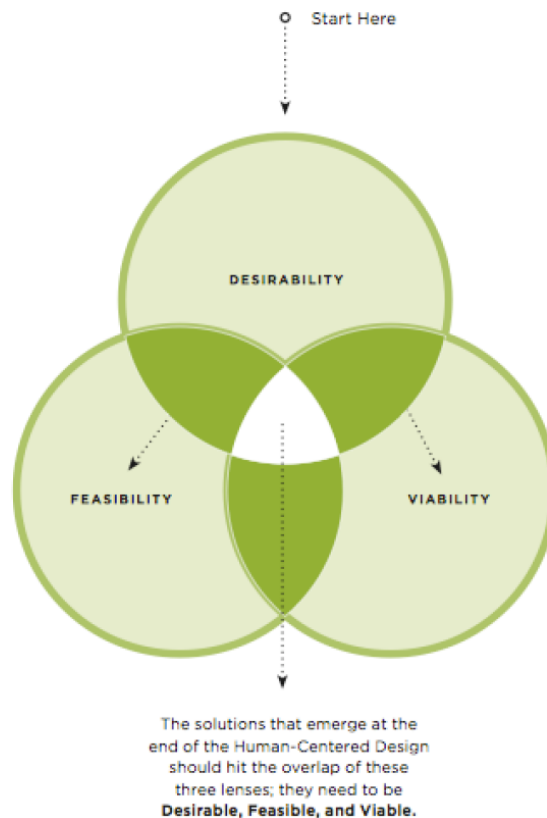
Estas tecnologías nos permiten actualmente poder realizar el monitoreo y control de dichos dispositivos a través de internet, lo cual nos permite en el mayor de los casos no estar absolutamente pendiente de sus acciones.

En simples palabras podríamos olvidarnos de ellos y programar remotamente su funcionamiento, a través de técnicas apropiadas para tal fin, para que luego estos dispositivos puedan ejecutar acciones ante determinados eventos, y monitorear cualquier variable de su entorno para su posterior análisis mediante herramientas de Business Intelligence o Big Data basándonos en los conocimientos actuales de aprendizaje automático o machine learning.

Pero al igual que ocurre en otras áreas de trabajo y desarrollo, no basta con que algo sea tecnológicamente factible, sino que además debe haber una necesidad real detrás de toda solución IoT, y esta además debe ser económicamente viable, en términos específicamente económicos como TIR VAN y período de repago.



Son los 3 principios básicos tras la metodología “Design Thinking”.



Fuente: Human Centered design by IDEO

El diseño de una solución IoT **no está asociado necesariamente al uso de la última tecnología disponible**, de hecho, es sugerido que sea así, no es necesario tener que correr detrás de los últimos avances tecnológicos, sino usar aquella tecnología que satisface la resolución del problema de nuestro usuario cliente y se ajuste a la viabilidad económica del proyecto.

Por lo que, usaremos módulos IoT, propios o desarrollados por terceros que mejor se ajusten a dichos requerimientos, introduciendo por supuesto requisitos que no estén específicamente relacionados con lo funcional, pero que sean considerados necesarios para la sostenibilidad a mediano y largo plazo en nuestra solución:

1. escalabilidad,
2. Mantenibilidad,
3. Soporte, entre otros.

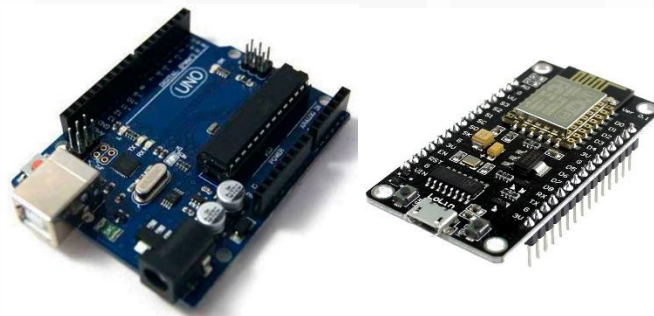


DISPOSITIVOS

Los dispositivos a utilizar son tal como lo vimos en unidades anteriores, aquellos que susceptibles de ser grabados a través de alguna plataforma o IDE (Entorno de desarrollo), que nos permita grabarle un firmware, que a su vez podría, en el mejor de los casos, disponer de un software o controlador local que, integrado con un servidor de gestión de dispositivos (“device management”), se encargue de gestionar las actualizaciones solicitadas remotamente desde el servidor central.

De esta manera, el responsable de los dispositivos podrá ordenar la actualización de todo su parque de forma sencilla, sin desplazamientos, y con reportes de resultados a medida que la herramienta establece la comunicación con los múltiples controladores locales.

Algunos de los ejemplos que se utilizarán en este curso serán, Arduino UNO, y ESP8266, que serán explicados oportunamente.



Placa Arduino UNO (izq) , ESP8266 NodeMCU (der)

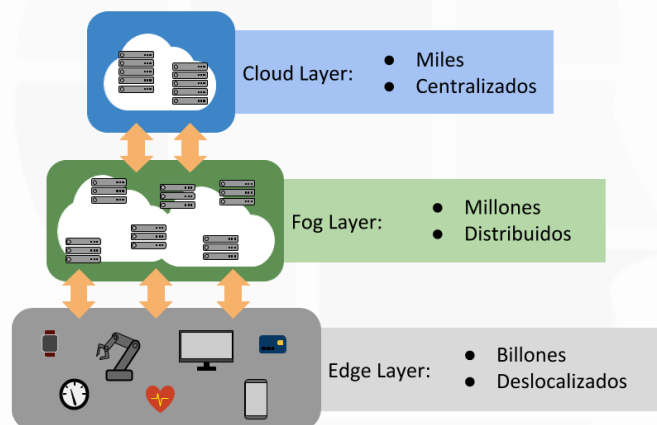


EDGE INTELLIGENCE

Cuando hablamos de EDGE estamos refiriéndonos a la frontera entre el mundo de la recolección de datos y la plataforma que los integra, es decir se trata del software que dispone la inteligencia local que se desea ejecutar en el dispositivo, de forma que su interacción con el entorno que se desea medir y controlar sea lo más rápida posible y sin dependencia de la red o software remoto.

En muchos casos se encuentra dentro del mismo endpoint, y no como un elemento concentrador.

Este software es susceptible de necesitar cambios, no sólo por alguna anomalía en su desempeño, sino también a cambios relacionados con las condiciones de contorno, es decir algún cambio tecnológico no previsto en la etapa de desarrollo que desee ser implementado de manera rápida y ágil, por lo que la disponibilidad del software de actualización remota facilita dicha tarea.



Fuente: <https://www.gradiant.org>

FOG LAYER

Por otro lado, a medida que un proyecto se hace cada vez más escalable aparece la necesidad de poner un pre procesamiento al servicio Cloud que deseamos utilizar para la implementación.

Es una descentralización de procesamiento de los datos, y no pretende en ningún caso el reemplazar a la estructura edge-cloud services, sino de aliviar el sistema.



ACTUALIZACION REMOTA

Uno de los principales objetivos de conectar un objeto a la red es permitir la actualización remota de este, esto se hace por ejemplo en servicios de WiFi a través de los denominados Over the Air, (OTA), existen ya servicios que se ofrecen que te brindan un id único para que te puedas conectar a través de estos servicios y programarlos y reconfigurarlos a deseo, los mismos son servicios pagos.

Uno de los que ofrece este servicio de forma gratuita es the things network (TTN), para equipos que trabajen con LPWAN, y utilizando sus gateways.

Así sea la actualización de su firmware, o la de cualquier software instalado en el mismo, supone normalmente un desafío para su responsable por diversos motivos:

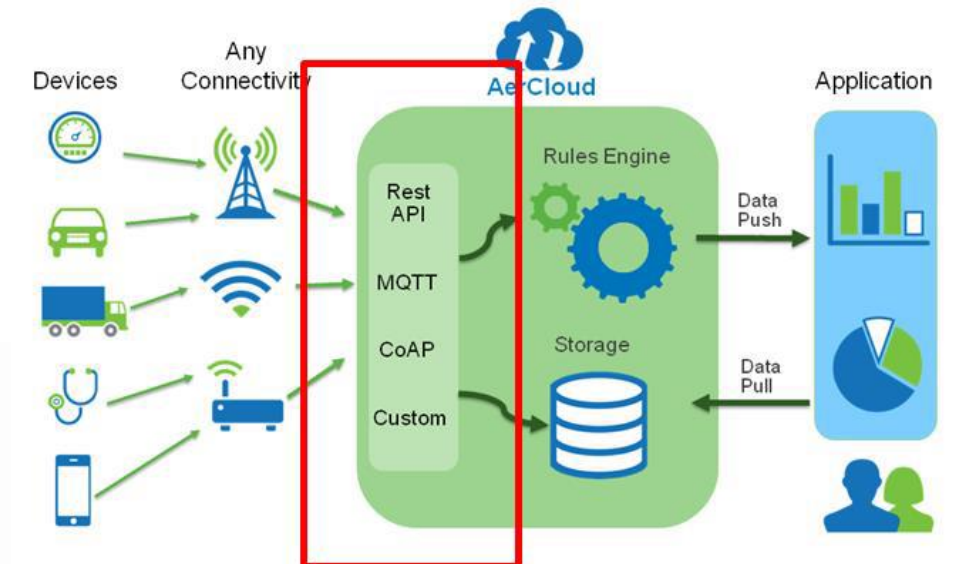
- ✓ Dificultad de acceso al mismo, pudiendo estos ubicarse a kilómetros de distancia o en un lugar de difícil acceso.
- ✓ Costos de desplazamiento y del tiempo de mano de obra utilizado en conectar físicamente al dispositivo para realizar la actualización manual.
- ✓ Mejoras debido a la seguridad, al detectar, por ejemplo, una vulnerabilidad del software instalado, requiriendo una instalación inmediata ante un bug vinculado a la seguridad.

ENVIO DE DATOS COLECTADOS

Otro de los principales objetivos es el de enviar a la Plataforma los datos recolectados, de forma que esta almacene, para su posterior procesamiento (reportes, análisis big data, predicción ML) y muestre los mismos para la identificación de situaciones en los usuarios.

Para ello, el dispositivo deberá ejecutar un software integrado con uno de los protocolos soportados por la Plataforma, a través del cual enviará periódicamente los datos solicitados.

Para ello, se podrían usar distintas estrategias como el pooling (consulta de datos desde la plataforma) o el push (envío periódico de los datos a un endpoint de la plataforma), utilizando distintos tipos de protocolos, y funcionalidades MQTT, CoAP, API, en un esquema similar al que se muestra en la figura.



Fuente: AerCloud

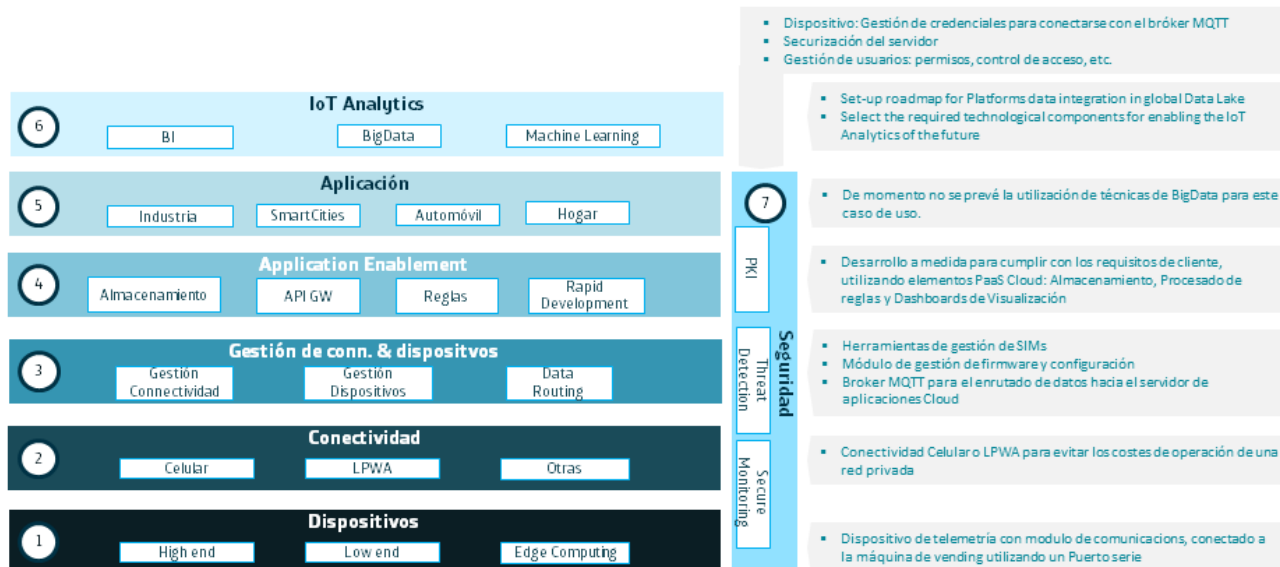
ACTUACION

Por último, otro de los objetivos es permitir la acción remota sobre el dispositivo: activación de contactor, cierre de válvulas, arranque de algún sistema de control, o cualquier otro.

Esta actuación es ejecutada por un software (firmware) del dispositivo específico, que es el mismo que ejecuta el envío de datos, o incluso el de actualización.

ESQUEMA FINAL

Un esquema final a modo de resumen, es el que se indica en la figura siguiente, se observa que se detallan layers adicionales a las comentadas, de acuerdo al grado de complejidad de la solución.



Fuente: Redes Plataformas y herramientas USAL, IoT.



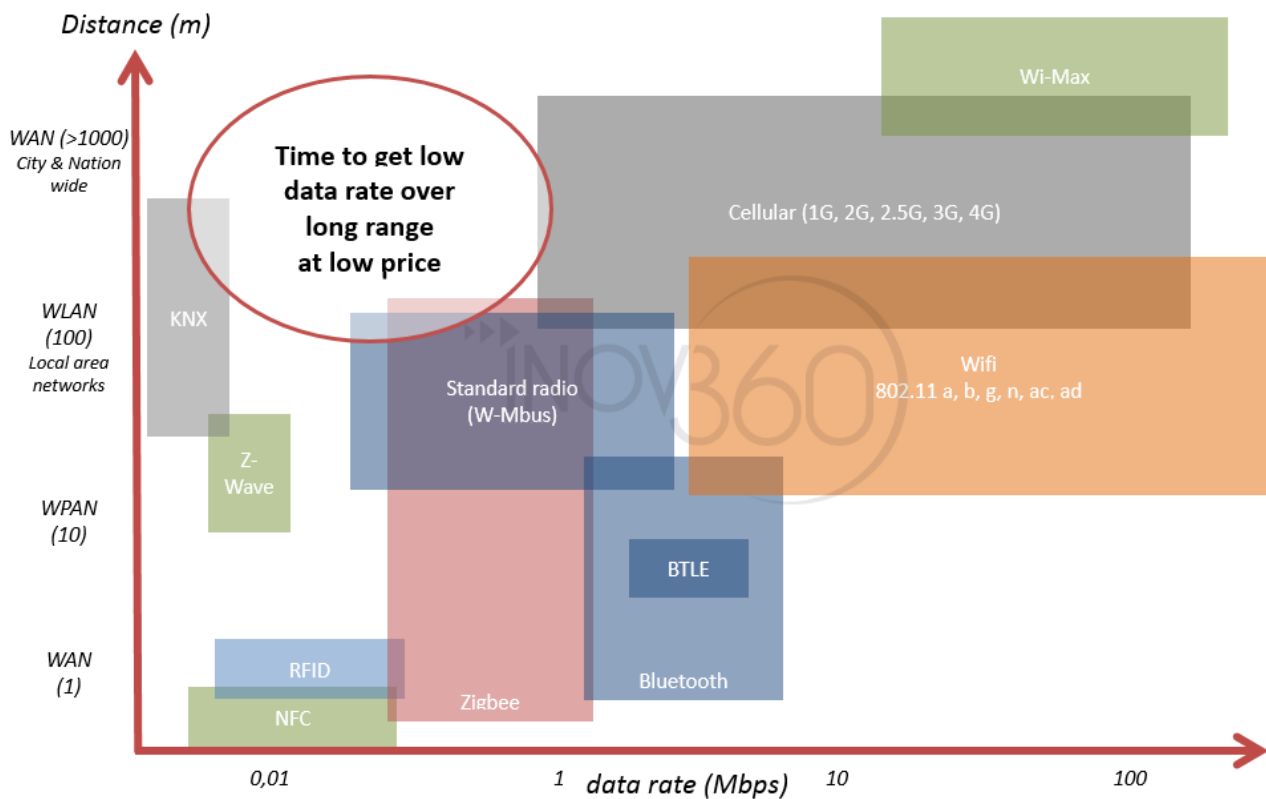
REDES DE COMUNICACIÓN

Para realizar la implementación final, es imprescindible elegir Uno de los principales factores que se deberán considerar a la hora de diseñar la solución es el tipo conectividad mediante el cual el dispositivo transmitirá y recibirá datos de manera inalámbrica.

En el mercado existen diversas ofertas tecnológicas, como se puede ver en la observar en la siguiente figura:



Fuente: INOV360



Fuente: INOV360

La selección de esta conectividad se puede efectuar utilizando los siguientes criterios:

- ✓ Requisitos técnicos:
 - Alcance, consumo de batería, cobertura en interiores.
 - Bidireccionalidad.
 - Throughput de datos.
 - Latencia.
- ✓ Tipo de red:
 - red pública
 - red privada
- ✓ Costo: uC + módulo de comunicaciones, (conexión si es pagando >>US\$/mes o año) mantenimiento de la red. [1]



REQUISITOS TECNICOS

Alcance y cobertura en interiores

Si el caso de uso requiere cobertura en interiores, por ejemplo, hospitales, subsuelos, o un gran alcance respecto al punto a punto en la base de comunicaciones, será necesario utilizar tecnologías adecuadas a estas situaciones:

- ✓ Redes de Wide Area para garantizar un gran alcance: Celular (incluyendo NB-IOT/LTE-M), Sigfox y Lora. [1]
- ✓ Redes con buena penetración en interiores o estructura mesh / repetidores:
 - Sin repetidores: NB-IOT/LTE-M, Sigfox y Lora.
 - Con repetidores o redes Mesh: Cualquier tecnología de corto alcance (Bluetooth, Zigbee, 802.14.x, ZWave, Wifi, W-MBUS) que se conecte a un router con salida a red pública (ADSL, Fibra, LTE) [1]



Fuente: Redes Plataformas y herramientas USAL, IoT.



CONSUMO DE BATERIA

Uno de las consideraciones más importantes, incluso en algunos casos la más importante es la alimentación del dispositivo.

Si el dispositivo está alimentado a batería será muy importante contar con una tecnología de bajo consumo, como pueden ser las redes LPWA, tanto funcionando en bandas no licenciadas como licenciadas, por otro lado, sistemas como WIFI, Bluetooth, Zigbee, u algún otro al tener más consumo deberán ser provistos de algún sistema de recarga a través de fuentes renovables, por ejemplo.

La mayoría de las opciones LPWAN proporcionan duraciones de batería de años si el número de mensajes enviados al día es reducido:

- ✓ LPWA licenciada: NB-IOT y LTE-M.
- ✓ LPWA en banda no licenciada: Sigfox y Lora.

BIDIRECCIONALIDAD, THROUGHPUT DE DATOS Y LATENCIA

Estos tres parámetros van a ser los que ayuden a descartar ciertas tecnologías de acceso.

- ✓ **Bidireccionalidad:** Si es requerido enviar mucha información desde el lado servidor (comandos, actualizaciones de firmware) de manera continua las tecnologías LPWA en banda no licenciada se podrán descartar (Sigfox y Lora), pues únicamente se permite el envío de un número limitado de mensajes al día. En banda licenciada LTE-M es la que ofrece un mayor throughput en sentido descendente. [1]
- ✓ **Throughput de datos.** Está relacionado tanto con el uso de bandas no licenciadas como el impacto en batería.
 - Si el throughput de datos es bajo, se recomienda el uso de tecnologías LPWA
 - Si el throughput de datos es alto, se deberá utilizar otras tecnologías de acceso: Ej. celular tradicional (LTE CAT-1 o CAT-4), WIFI o Bluetooth [1]
- ✓ **Latencia.** Hablamos de los retardos temporales que se producen dentro de una red. Un retardo es producido por la demora en la propagación y transmisión de paquetes dentro de la red. [1]



- Si nuestro caso de uso tiene unos requisitos de retardo por debajo de unos pocos milisegundos (aplicación de misión crítica), se deberá optar por redes que garanticen bajas latencias: ej. 5G o 802.11p. [1]
- Si el caso de uso admite unos pocos segundos de latencia, se deberá optar por tecnologías celulares o LTE-M [1]
- Si el caso de uso admite segundos de latencia, se podrá optar por Sigfox, Lora o NB-IoT. [1]

REDES PÚBLICAS Y PRIVADAS

La elección del tipo de red, es otra característica fundamental, ya que impacta directamente en las variables económicas a analizar. Estas se pueden clasificar en públicas, privadas o híbridas:

- ✓ **Públicas:** Son ofrecidas por un operador de red o de telecomunicaciones. (2G, 3G, LTE, NB-IoT, Sigfox.)
- ✓ **Privadas:** Son instaladas directamente por el usuario o una empresa, o ambos implementando redes *ad hoc*. (WiFi, LoRa, Bluetooth, ZigBee 802.15.4)
- ✓ **Híbridas:** Están compuestas por una mezcla de ambas. Por ejemplo, un gateway que proporciona conectividad LoRa entre dispositivos cercanos, y luego dispone de una salida a través de 3G o Ethernet.

WIRELESS TECHNOLOGIES AT A GLANCE					
Technology	Frequency	Data rate	Range	Power	Cost
2G/3G	Cellular bands	10 Mb/s	Several km	High	High
802.15.4	2.4 GHz	250 kb/s	100 m	Low	Low
Bluetooth	2.4 GHz	1, 2, 1, 3 Mb/s	100 m	Low	Low
LoRa	< 1 GHz	<50 kb/s	2-5 km	Low	Medium
LTE Cat 0/1	Cellular bands	1-10 Mb/s	Several km	Medium	High
NB-IoT /LTE-M	Cellular bands	0.1-1 Mb/s	Several km	Medium	Medium
SIGFOX	<1 GHz	Very low	Several km	Low	Medium
Weightless	<1 GHz	0.1-24 Mb/s	Several km	Low	Low
Wi-Fi (11n/h)	2.4, 5, <1 GHz	0.1-1 Mb/s	Several km	Medium	Low
WirelessHART	2.4 GHz	250 kb/s	100 m	Medium	Medium
ZigBee	2.4 GHz	250 kb/s	100 m	Low	Medium
Z-Wave	908.42 MHz	40 kb/s	30 m	Low	Medium

Fuente: Redes, plataformas y herramientas, USAL IoT.



PLATAFORMAS

NECESIDAD DE UNA PLATAFORMA IoT

Las Plataformas IoT son las soluciones en una implementación de internet de las cosas que nos permite observar la interacción entre los distintos dispositivos de la infraestructura.

Podríamos desarrollar estas soluciones sin la necesidad de integrar los datos en ninguna Plataforma, es decir, como mencionamos que otras oportunidades realizar implementaciones del tipo M2M, desarrollando un backend con toda la funcionalidad requerida, API, bases de datos, diseño web que integre una visualización adecuada.

El uso de alguna de las Plataformas IoT disponibles conlleva una serie de ventajas.

Las Plataformas IoT ofrecen:

- ✓ **Protocolos de integración de datos standard.** Con esto, evitamos tener que pensar un formato de intercambio de datos propio, y lo sustituimos por uno previamente probado en cientos de Soluciones previas. Además, nos facilita el desarrollo en el dispositivo si ya existen librerías que lo implementan. [1]
- ✓ **Almacenamiento de datos.** Muchas de las Plataformas IoT nos permiten (si no lo incluye de serie) el almacenamiento seguro, redundado y escalable de los datos recolectados. [1]
- ✓ **Exposición de APIs.** Mediante estas interfaces podremos recuperar los datos recolectados desde múltiples aplicaciones: web, móvil, integración en sistemas IT existentes. [1]
- ✓ **Tratamiento de datos:** Una vez integrados los datos en la Plataforma IoT, ésta nos suele brindar un catálogo de servicios adicionales como o Agregación de datos (medias, máximos, mínimos), En distintos intervalos temporales)
 - Creación de reportes
 - Análisis Big Data
 - Desarrollo y Hosting de aplicaciones [1]

APLICACIONES

La aplicación es el punto de contacto de la solución con nuestro cliente, dado que es la parte de nuestra infraestructura que interacciona con nuestro sistema para resolver su problema o necesidad, y lo muestra.

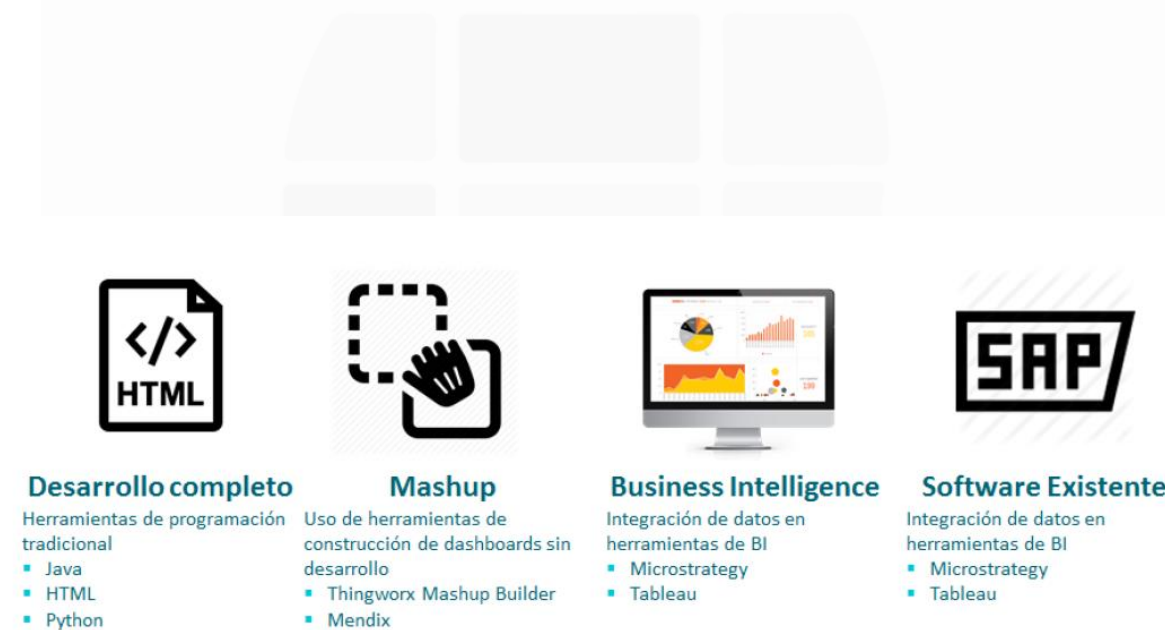
Le permite procesar de una manera amigable los datos recolectados, resaltando aquella información que le es más relevante según su perfil; así como actuar sobre el entorno gestionado cuando sea necesario (o programar las reglas de actuación en función de los datos reportados)



Las aplicaciones deben customizarse en función del perfil del usuario que esté accediendo, ya que sus necesidades pueden ser diferentes, por lo que un usuario final, seguramente no deba observar los detalles más minuciosos de la comunicación o incluso mismo del desempeño del sistema.

En general, las aplicaciones IoT tratan de dar un servicio de valor añadido sobre el producto inicialmente vendido (la “Thing” que se monitoriza mediante la Solución). Pudiendo en muchos casos, analizando la escalabilidad antes de disparar la venta a ayudar a nuestro cliente a pasar de ser un proveedor de productos, a ser un proveedor de Servicios.

En general, podemos aplicar soluciones IoT a múltiples sectores de la sociedad para resolver problemas específicos o añadir valor a soluciones existentes. Podríamos citar soluciones de **Smart Cities** (recolección de indicadores de la ciudad, eficiencia y control de servicios públicos), **Smart Energy** (ahorro de energía manteniendo el confort y la seguridad), **Smart Mobility** (gestión de flotas, vehículo autónomo). [1]



Fuente: Redes, plataformas y herramientas USAL IoT.



EJEMPLO DE USO

PLATAFORMAS

Ahora es necesario poder integrar todo este conocimiento en plataformas que sean del tipo abierta a la comunidad de tal manera que podamos integrar nuestra solución a una plataforma pre existente sin la necesidad del desarrollo e nuestro dashboard.

Existen distintos tipos de plataforma, numerosa cantidad, en este curso nos vamos a abocar a un par de ellas para poder aprender cómo manejarlos.

ADAFRUIT

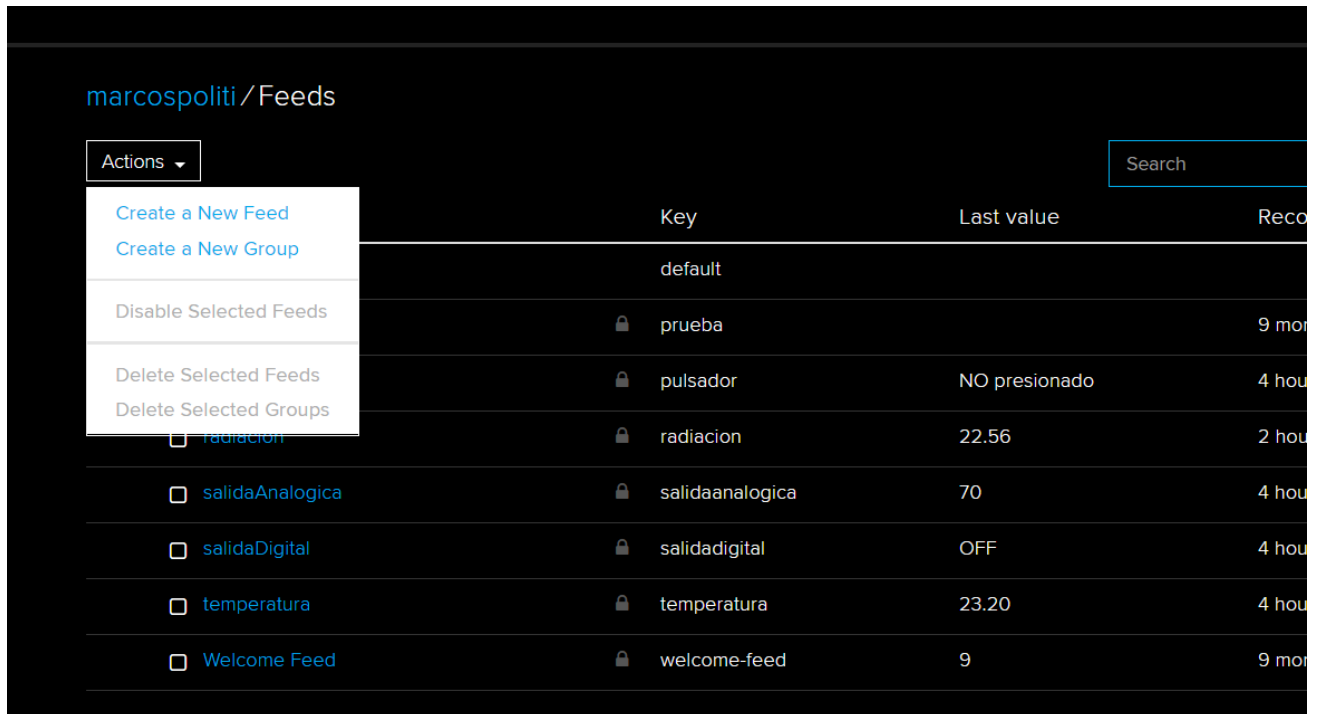
<https://io.adafruit.com>

EL primer caso de plataforma es la de Adafruit, es una plataforma que presenta buenas prestaciones para quienes se inicien en el campo de IoT, de aprendizaje bastante intuitivo, sin embargo, presenta limitaciones a la hora de la comunicación y la publicación de datos en ella.

Por empezar nos limitan la cantidad de dashboard, (tablero de control), y además la cantidad de reportes por minuto de estos datos.



Primeramente, debemos crear una cuenta, y cuando deseemos utilizar la misma con algún topic en particular asociado a algún objeto visual, debemos recurrir a FEEDS, allí debemos ingresar en CREATE NEW FEED



The screenshot shows the 'marcospoliti/Feeds' interface. On the left, there is an 'Actions' dropdown menu with options: 'Create a New Feed', 'Create a New Group', 'Disable Selected Feeds', 'Delete Selected Feeds', and 'Delete Selected Groups'. The main area displays a table of feeds with columns: 'Key', 'Last value', and 'Reco' (likely 'Recent'). The table contains several rows, each with a checkbox, a feed name, a lock icon, a key, a last value, and a recent update time.

	Key	Last value	Reco
	default		
<input type="checkbox"/>	prueba		9 mo
<input type="checkbox"/>	pulsador	NO presionado	4 hou
<input type="checkbox"/>	radiacion	22.56	2 hou
<input type="checkbox"/>	salidaAnalogica	70	4 hou
<input type="checkbox"/>	salidaDigital	OFF	4 hou
<input type="checkbox"/>	temperatura	23.20	4 hou
<input type="checkbox"/>	welcome-feed	9	9 mo

Y damos un nombre que represente lo que queremos monitorear, CO2, por ejemplo.



Create a new Feed

Name
CO2

Description

Cancel Create

Y presionamos CREATE

marcospoliti / Feeds

Actions Search

Group / Feed	Key	Last value	Recorded
<input type="checkbox"/> Default	default		
<input type="checkbox"/> CO2	  co2		a few sec
<input type="checkbox"/> pulsador	 pulsador	NO presionado	4 hours a
<input type="checkbox"/> radiacion	 radiacion	22.56	2 hours a
<input type="checkbox"/> salidaAnalogica	 salidaanalogica	70	4 hours a
<input type="checkbox"/> salidaDigital	 salidadigital	OFF	4 hours a
<input type="checkbox"/> temperatura	 temperatura	23.20	4 hours a

Posteriormente debemos crear un DASHBOARD



Home
Feeds
Dashboards
Triggers
View AIO Key
API Docs
FAQ
Learn
News

marcospoliti / Dashboards

Actions ▾

- Create a New Dashboard
- Edit Selected Dashboard
- Remove Selected Dashboards

- ☐ iotPlacaA
- ☐ iotPlacaB
- ☐ Welcome Dashboard

Dándole el nombre deseado

Create a new Dashboard

Name

UTN FRBA IoT

Description

Cancel Create



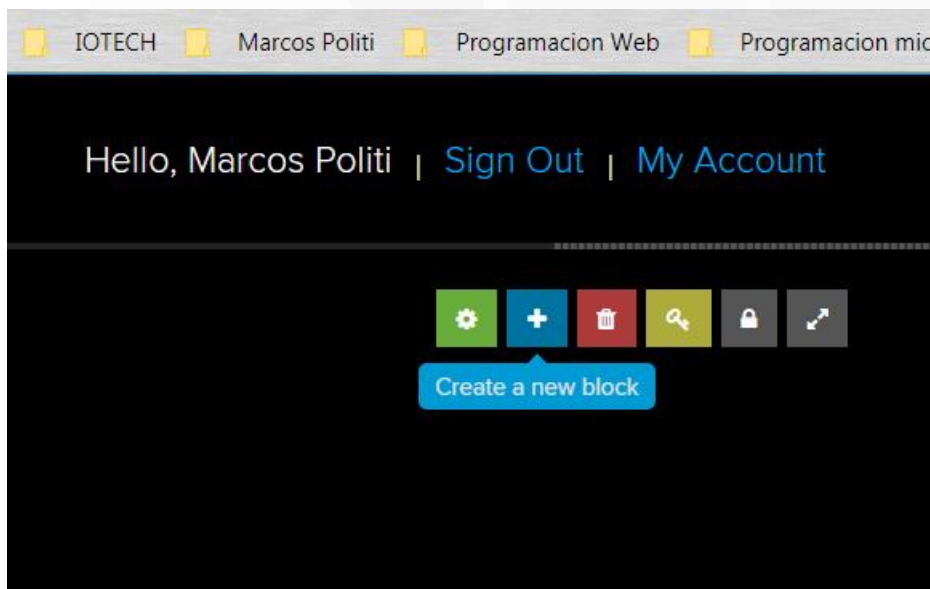
marcospoliti / Dashboards

Actions ▾

<input type="checkbox"/> Name ▾	Key ▾
<input type="checkbox"/> UTN FRBA IoT	utn-frba-iot
<input type="checkbox"/> iotPlacaC	iotplacac
<input type="checkbox"/> iotPlacaA	iotplacaa
<input type="checkbox"/> iotPlacaB	iotplacab
<input type="checkbox"/> Welcome Dashboard	welcome-dashboa

1

Luego hay que asociar esos feeds a un objeto en particular para eso se va a “+” y agrego.



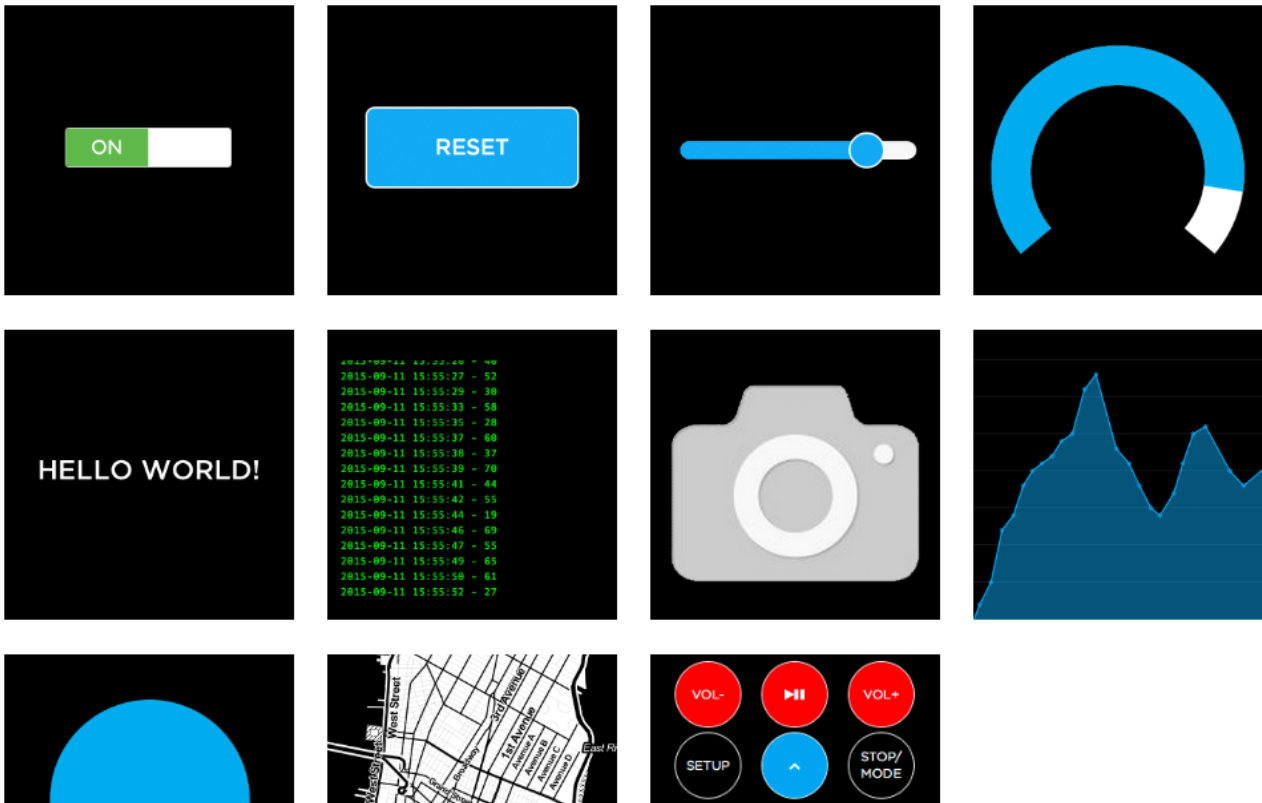
Y debo elegir entre los que deseo.



Create a new block



Click on the block you would like to add to your dashboard. You can always come back and switch the block type later if you change your mind.



El elegido es GAUGE



Posteriormente, debo asociar el feed.



Choose feed



A gauge is a read only block type that shows a fixed range of values.

If you have lot of feeds, you may want to use the search field. You can also create a feed quickly below.

Group / Feed	Last value	Recorded
<input checked="" type="checkbox"/> CO2		6 minutes ago
<input type="checkbox"/> pulsador	NO presionado	4 hours ago
<input type="checkbox"/> radiacion	22.56	3 hours ago
<input type="checkbox"/> salidaAnalogica	70	4 hours ago
<input type="checkbox"/> salidaDigital	OFF	4 hours ago
<input type="checkbox"/> temperatura	23.20	4 hours ago
<input type="checkbox"/> Welcome Feed	9	9 months ago

Y allí configurarlo

Choose feed

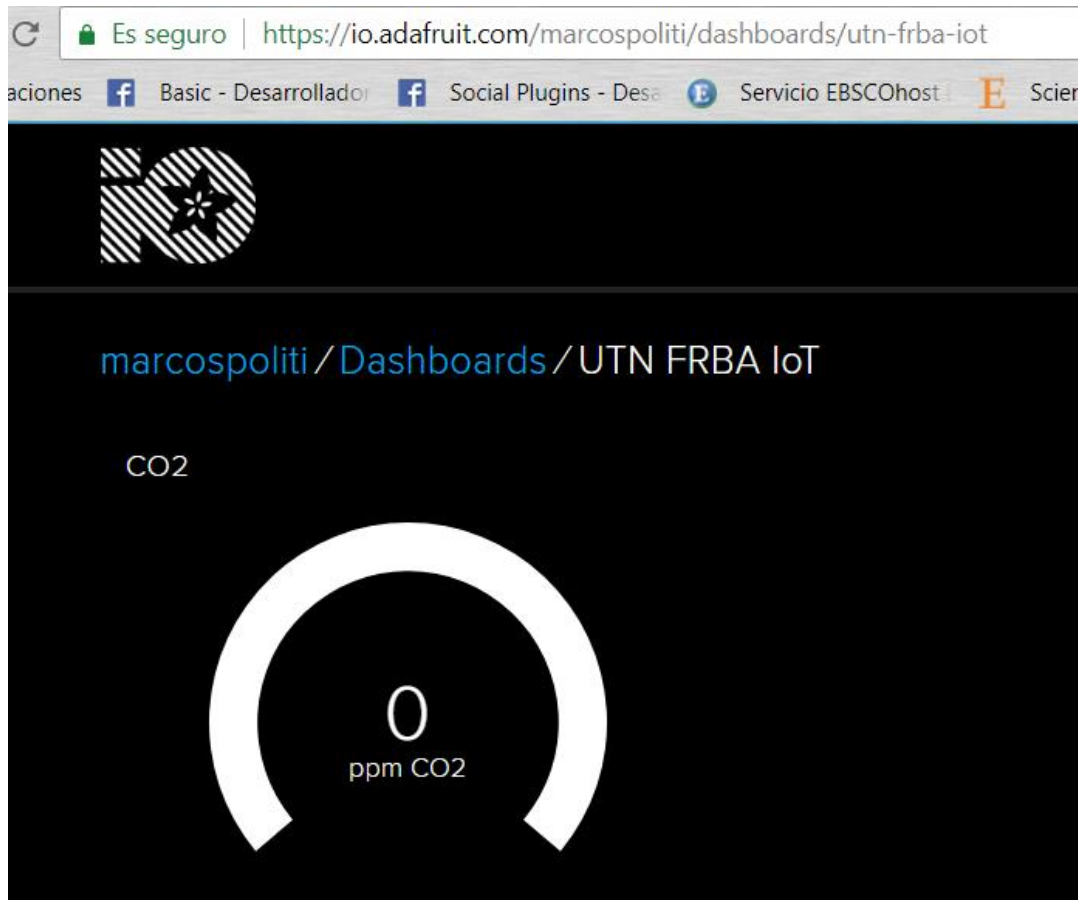


A gauge is a read only block type that shows a fixed range of values.

If you have lot of feeds, you may want to use the search field. You can also create a feed quickly below.

Group / Feed	Last value	Recorded
<input checked="" type="checkbox"/> CO2		6 minutes ago
<input type="checkbox"/> pulsador	NO presionado	4 hours ago
<input type="checkbox"/> radiacion	22.56	3 hours ago
<input type="checkbox"/> salidaAnalogica	70	4 hours ago
<input type="checkbox"/> salidaDigital	OFF	4 hours ago
<input type="checkbox"/> temperatura	23.20	4 hours ago
<input type="checkbox"/> Welcome Feed	9	9 months ago

Quedando



No vamos a entrar en detalles de cómo se llega a publicar un valor en dicha plataforma ya que será tema de la siguiente unidad.

Lo que se espera entonces es, poder configurar un microcontrolador, para que el mismo, pueda medir un sensor de concentración de dióxido de carbono y el mismo subirlo a dicha plataforma para que el usuario final, pueda observar cual es la concentración de dióxido de carbono en algún lugar del mundo.

CAYENNE

<https://cayenne.mydevices.com/cayenne/login>

La plataforma que ofrece Cayenne, es muy amigable e intuitiva, muy recomendable para usar en proyectos que integren Arduino, o algún dispositivo que contenga ESP8266, en sus distintas versiones.

Desde mi punto de vista al igual que Adafruit, sin embargo existen muchas cosas que no son asequibles al usuario respecto a la transferencia de datos a este servidor, lo que hace que uno deba meditar antes de implementar una solución en la que uno desea escalar, en la cantidad de dispositivos y se desea tener control de todo lo que ocurre en la transferencia de datos.



Aunque bien es cierto que es una plataforma muy cuidada desde los aspectos de seguridad, y muy probada por lo que no hace imposible su implementación, en ese sentido su ubicación es superior en el mercado respecto a la de Adafruit que es más hobbista.

Es seguro | <https://cayenne.mydevices.com/cayenne/login>

Aplicaciones Basic - Desarrollador Social Plugins - Des Servicio EBSCOhost ScienceDirect - Search www.scopus.com UTN FRBA IoT IOTECH Marcos Politi

myDevices

myDevices Cayenne

LOG IN

EMAIL marcospoliti@outlook.com

PASSWORD [Forgot password?](#)

Log In

☒ KEEP ME LOGGED IN

Una vez creada la cuenta, y logueado, vamos a ADD, DEVICE/WIDGET

Cayenne
Powered by myDevices

Temperatura y Led Led ESP8266 TempHum ESP8266 +

Add new...
Device/Widget
Event
Trigger
Project

Devices & Widgets

Search

DEVICES

Single Board Computers >

MicroControllers >

Sensors >

Actuators >

Extensions >

LoRa >

Single Board Computers

Raspberry Pi

Microcontrollers

SparkFun



Y allí tendremos la posibilidad de asociar nuestros dispositivos, clickeando sobre ellos si es que se encuentran en la lista, o bien seleccionado BRING YOUR OWN THING en el caso que no se encuentre en la misma

The screenshot displays the Cayenne IoT dashboard interface. On the left is a sidebar titled 'Devices & Widgets' containing a search bar, a list of device categories (Single Board Computers, MicroControllers, Sensors, Actuators, Extensions, LoRa), and a 'Custom Widgets' section. Below the sidebar is a 'CAYENNE API' section with a prominent blue button labeled 'Bring Your Own Thing'. The main content area is divided into three sections: 'Single Board Computers' featuring a Raspberry Pi; 'Microcontrollers' featuring an Arduino, a SparkFun ESP8266 Thing Dev Board, and a Generic ESP8266; and 'Sensors' featuring Temperature, Luminosity, Pressure, Proximity/Distance, Motion, and Generic sensor modules.

Una vez allí, se le asignara un nombre aleatorio al dispositivo, otorgando al usuario, datos importantes para ser volcados en la programación del dispositivo a conectar, quedando a la espera de la conexión del mismo a la red.



Aplicaciones Basic - Desarrollador Social Plugins - Des Servicio EBSCOhost ScienceDirect - Search www.scopus.com UTN FRBA IoT IOTEC Marcos Politi Programacion Web

Cayenne
Powered by myDevices

Temperatura y Led Led ESP8266 TempHum ESP8266 + Create App Submit Project

Add new...

- Control Led ESP8266
- Device 377e**
- ESP8266 DHT11
- ESP8266 Placa B

Step 2: Connect your Device

OFFICIAL SDKS

- Arduino MQTT
- Cayenne MQTT mbed
- Embedded C
- C++
- Cayenne MQTT Python
- Node.JS

[View all SDKs on GitHub](#)

NEED HELP?

MQTT USERNAME:
554cdf50-f0a9-11e7-abe9-1721c4c13600

MQTT PASSWORD:
598bec3b273b7116ce1f6bdb5387a53dd3b688f8

CLIENT ID:
fb220e60-84aa-11e8-9179-3d0af28602c7

MQTT SERVER: mqtt.mydevices.com MQTT PORT: 1883

NAME YOUR DEVICE (optional):
Device 377e

Waiting for board to connect...



MQTT USERNAME:

554cdf50-f0a9-11e7-abe9-1721c4c13600



MQTT PASSWORD:

598bec3b273b7116ce1f6bdb5387a53dd3b688f8



CLIENT ID:

fb220e60-84aa-11e8-9179-3d0af28602c7



MQTT SERVER:

mqtt.mydevices.com

MQTT PORT:

1883

NAME YOUR DEVICE (optional):

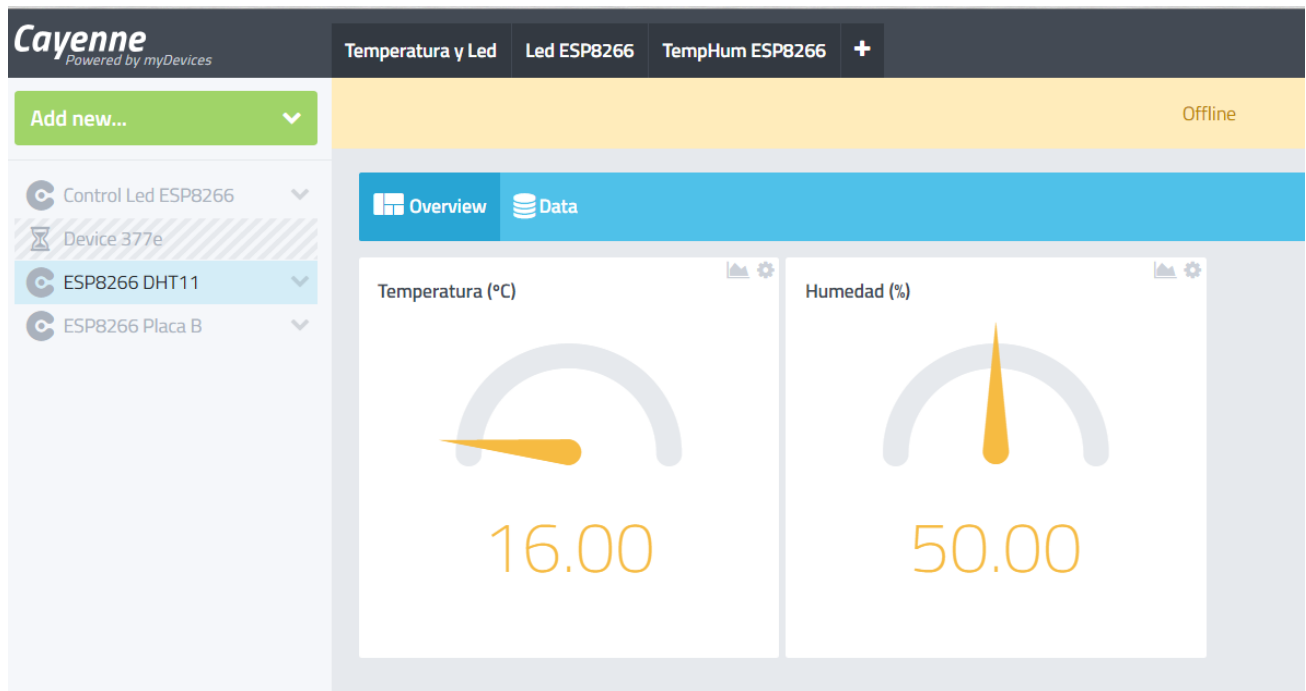
Device 377e

⌛ Waiting for board to connect...

Remove



Una vez conectado, se podrá monitorear y controlar a distancia dispositivos conectados a la red.

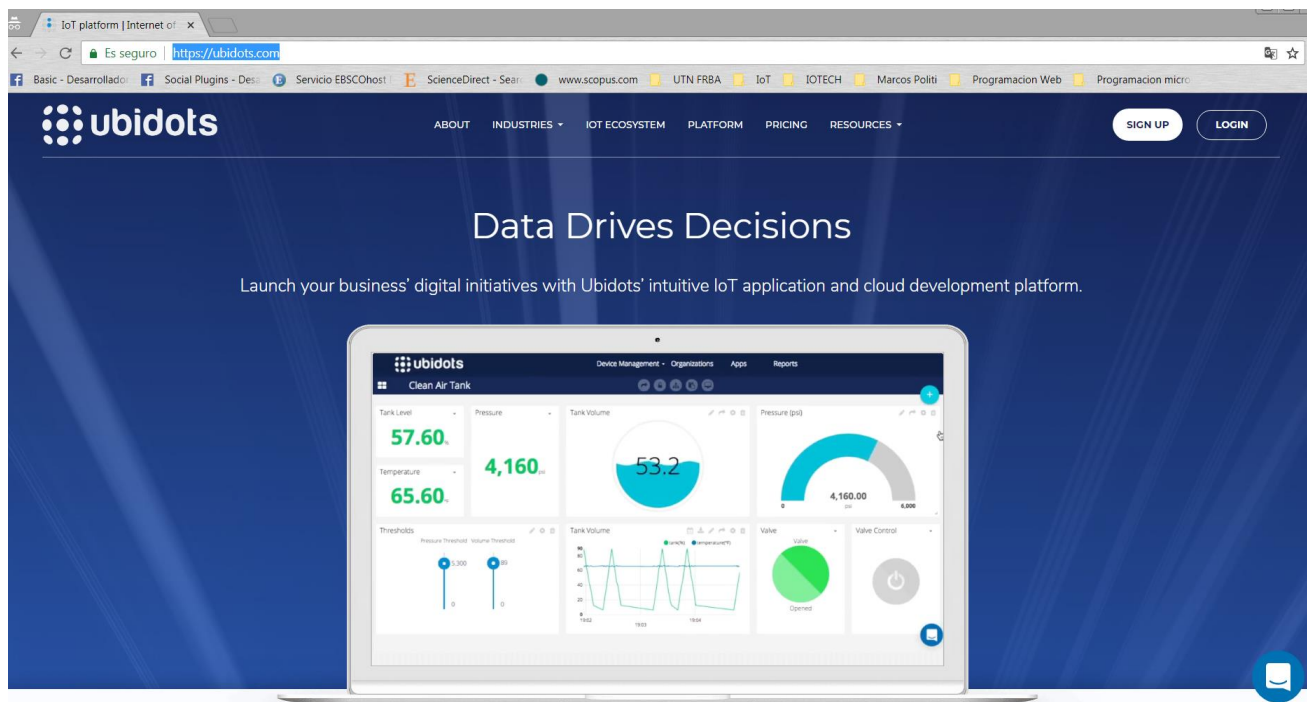


Nuevamente no vamos a entrar en detalles de cómo se Publican y Subscriben a estos topics, ya que será motivo de la unidad siguiente.



UBIDOTS

<https://ubidots.com/>



Esta es una plataforma que tranquilamente podría ser utilizada para ofrecer en una implementación, escalable y con posibilidad de mantenimiento y soporte adecuado por parte de la empresa.

Existen las versiones pagas, con distintos planes a medida.

Flexible pricing to take your project from prototype to production.

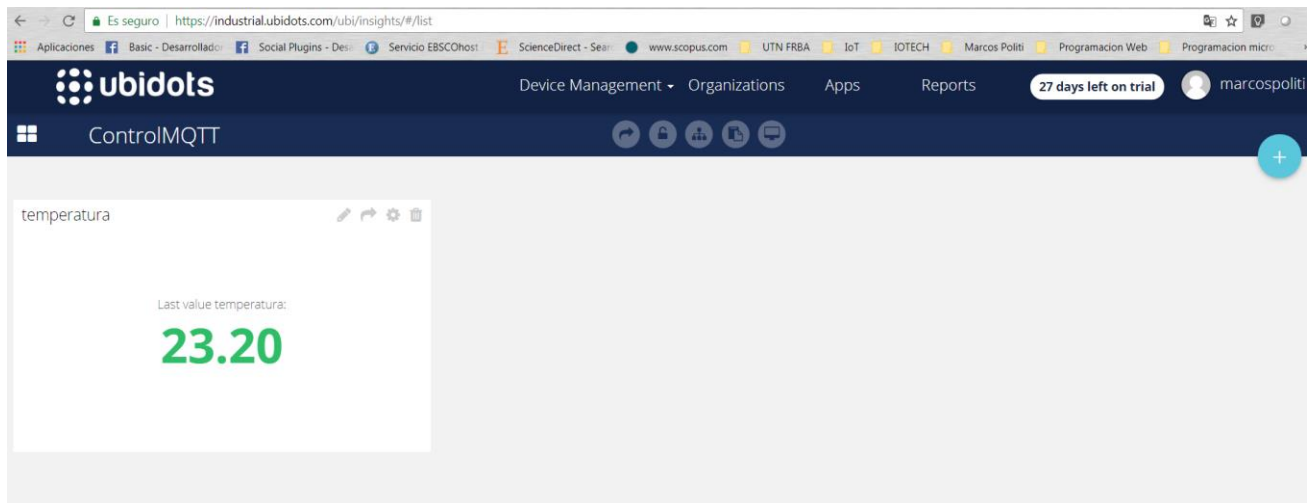
	IoT Entrepreneur	Professional	Industrial	Scale
	\$49 /month	\$199 /month	\$499 /month	\$1,799 /month
	REQUEST A FREE TRIAL	REQUEST A FREE TRIAL	REQUEST A FREE TRIAL	REQUEST A FREE TRIAL
Devices 1 device = 20 variables	25	200	1,000	4,000
Data Ingestion	2 Million dots per month	15 Million dots per month	50 Million dots per month	200 Million dots per month
Data Extraction	2 Million dots per month	15 Million dots per month	50 Million dots per month	200 Million dots per month

Contacto: consultas@elearning-total.com
Web: www.elearning-total.com



Ofreciendo un TRIAL por 30 días de formato “Industrial”

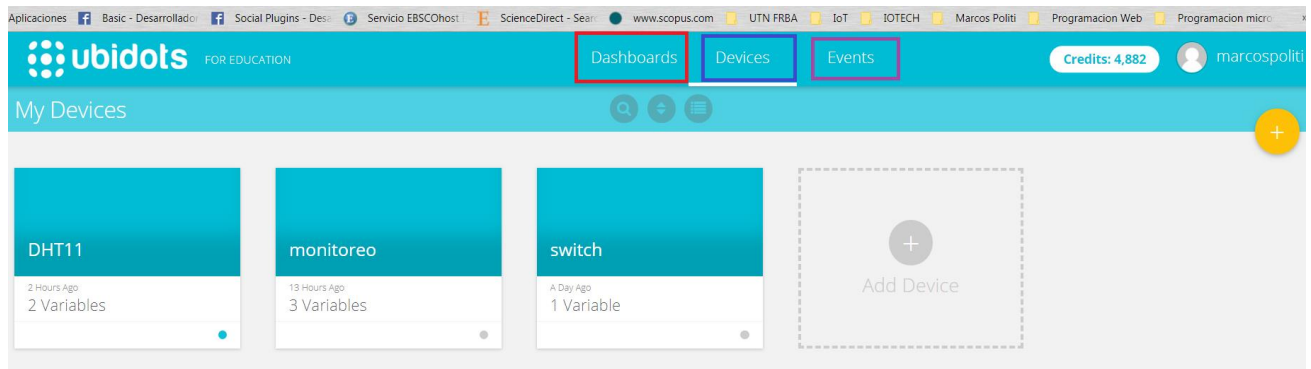
<https://industrial.ubidots.com>



Hasta 2019, existía la versión **Estudiantil/ Investigación**, actualmente ese tipo de cuenta ha sido suspendida para la oferta, sin embargo las que fueron creadas bajo esta modalidad siguen funcionando.

<https://app.ubidots.com/accounts/signin/>

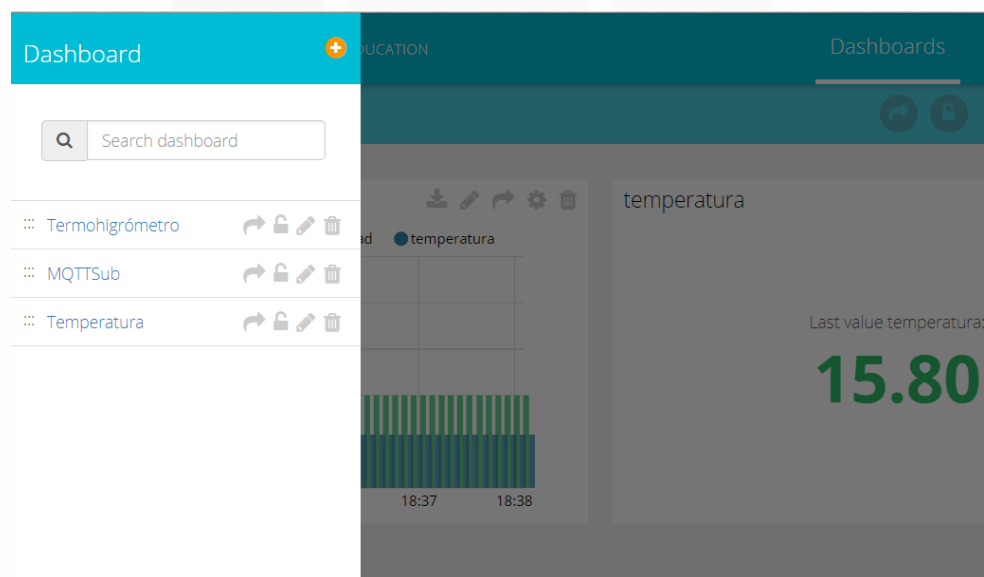




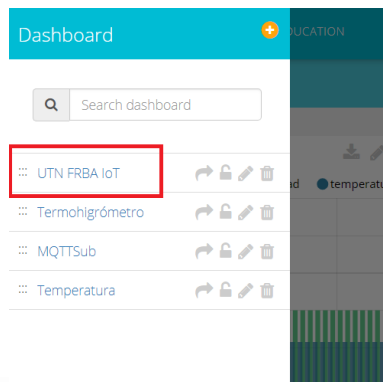
El paradigma de trabajo es similar al del ambiente industrial, entro de su menú, existen tres posibilidades

DEVICES: Administra los dispositivos a los que voy a poder ingresarle datos a distancia, publicando o subscribiendome a ellos, con las bibliotecas para tal fin.

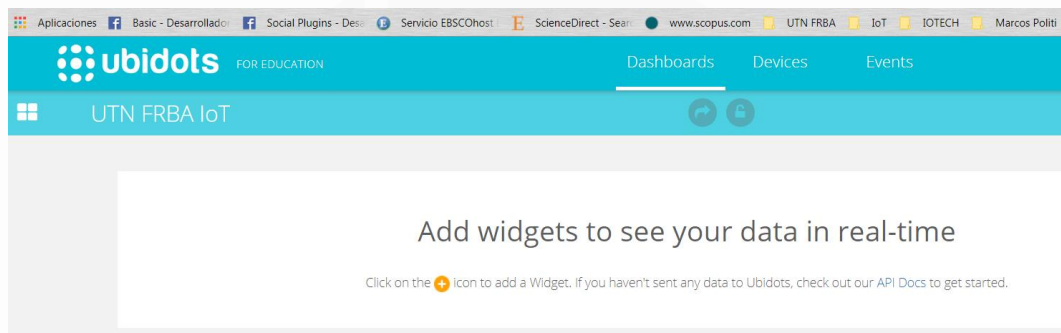
DASHBOARDS: Puedo administrar tableros que incluyen estos DEVICES, para poder Publicar y Subscribir a ellos.



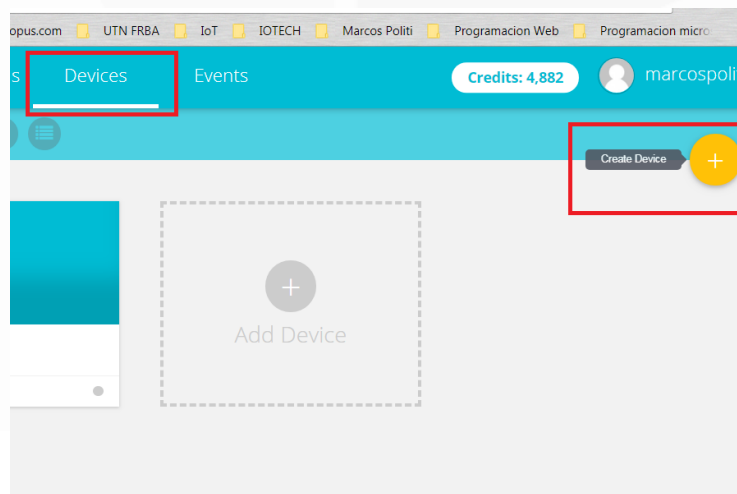
Para generar un nuevo **DASHBOARD**, presiono en “+”

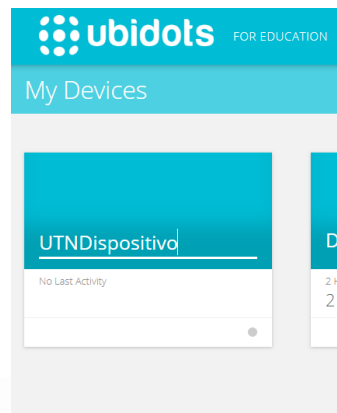


Ahora debo asociarle **WIDGETS** que estarán posteriormete asociados a **DEVICES**.

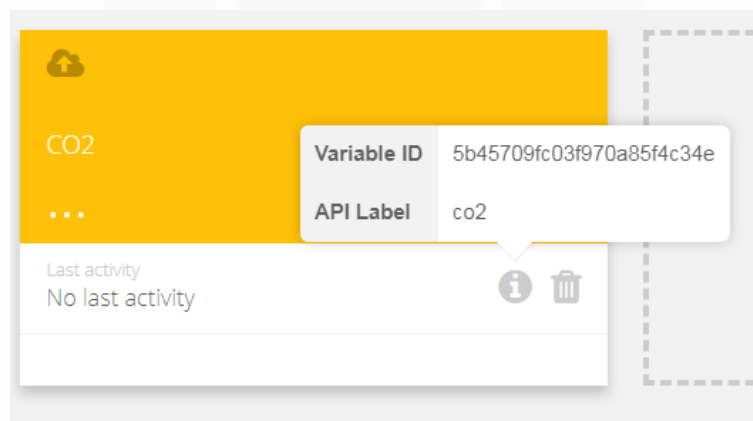


Creemos primeramente un **DEVICE**.





Ahora debo agregar una variable al mismo, presiono “+” y elijo DEFAULT, allí obtengo:



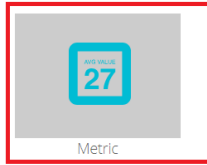
Con el ID correspondiente a este medidor de CO2, vuelvo a DASHBOARD y selecciono **METRIC, LAST VALUE, UTN DISPOSITIVO**



How would you like to see your data?



Chart



Metric



Map



Table



Indicator



Control



HTML Canvas

Select an operation and a time frame for your metric:

Show me the



Average



Maximum



Minimum



Sum



Count



Last value

Select a Device



UTNDispositiv



DHT11



monitoreo



switch

Filter Devices

Y finalmente CO2



UTNDispositiv



DHT11



monitoreo



switch



CO2

Finish

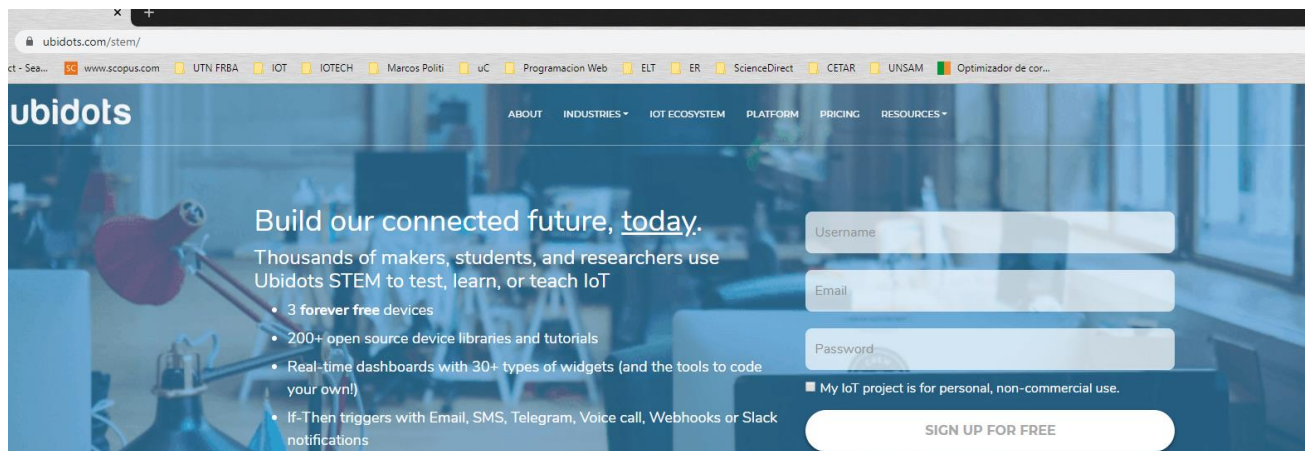


UBIDOTS STEM

En reemplazo de esta versión app educativa Ubidots lanzó la versión stem

<https://ubidots.com/stem/>

El sistema ofrecido tal como lo indican es para 3 dispositivos gratis de por vida, el beneficio que incluye esta plataforma es que la interfaz es la misma que la industrial, (pero sin pagar), aunque con ciertas desventajas en lo referido a la visualización de datos y posibilidad de agrupar por perfiles de acceso.



La url de su API de acceso es **industrial.api.ubidots.com** y el port es el 1883.



PUBLICACION EN UBIDOTS CON MOSQUITTO

Ahora vamos a hacer uso del Cliente **MOSQUITTO**, para poder emular un dispositivo, y ver cómo queda esos datos en pantalla.

Para ello necesito el formato siguiente, recordando la clase anterior.

-h: Es el HOST.

-p: Es el PUERTO.

-u: Es el USUARIO.

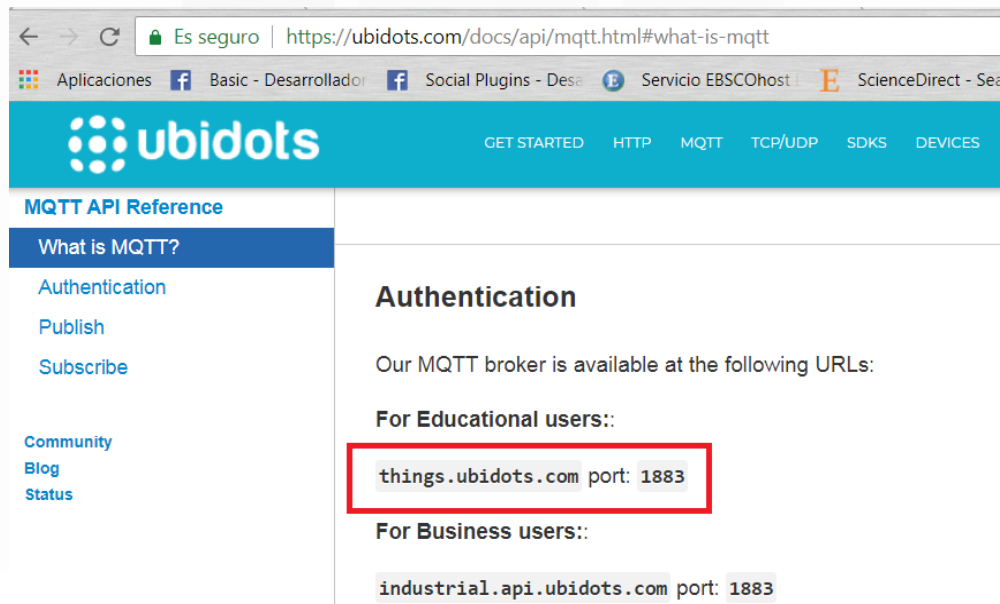
-P: Es el PASSWORD.

-t: Es el TOPIC donde publicar.

-m: Es el MENSAJE a enviar.


Para tal fin y conocer el formato de la trama se debe consultar:

<https://ubidots.com/docs/api/mqtt.html#publish>





Add widgets to see your data in real-time


Click on the  icon to add a widget. If you haven't sent any data to Ubidots, check out our [API docs](#) to get started.

To connect to our MQTT broker, use your Ubidots **TOKEN** as the MQTT **username**, and leave the **password** field empty.

MQTT Credentials	Ubidots
username	→ TOKEN
password	→ <i>Leave blank</i>

If the connection is successful you should be able to publish and subscribe to MQTT topics, otherwise the broker will return an error and disconnect the client.

El **TOKEN** debe ser consultado aquí, seleccionando **DEFAULT TOKEN**.

Credits: 4,882  marcospoliti ▾

- My Profile
- API Credentials**
- Docs
- Log out

Los topics tendrán el formato que se indica

Publish values to a Device

To send values to several variables to the broker you have to send a **PUBLISH** message with the topic:

`/v1.6/devices/{LABEL_DEVICE}`

Ahora tal como se hizo en la Unidad 2, debemos abrir el Cliente mosquitto, ejecutando con doble clic, primeramente, **mosquitto.exe**, y luego abriendo en **CMD** y en la ruta que corresponda ejecutando **mosquitto_pub** con los parámetros que se indican. (-u fue quitado ya que existe un único **DEFAULT TOKEN** para todo los **DEVICES**)

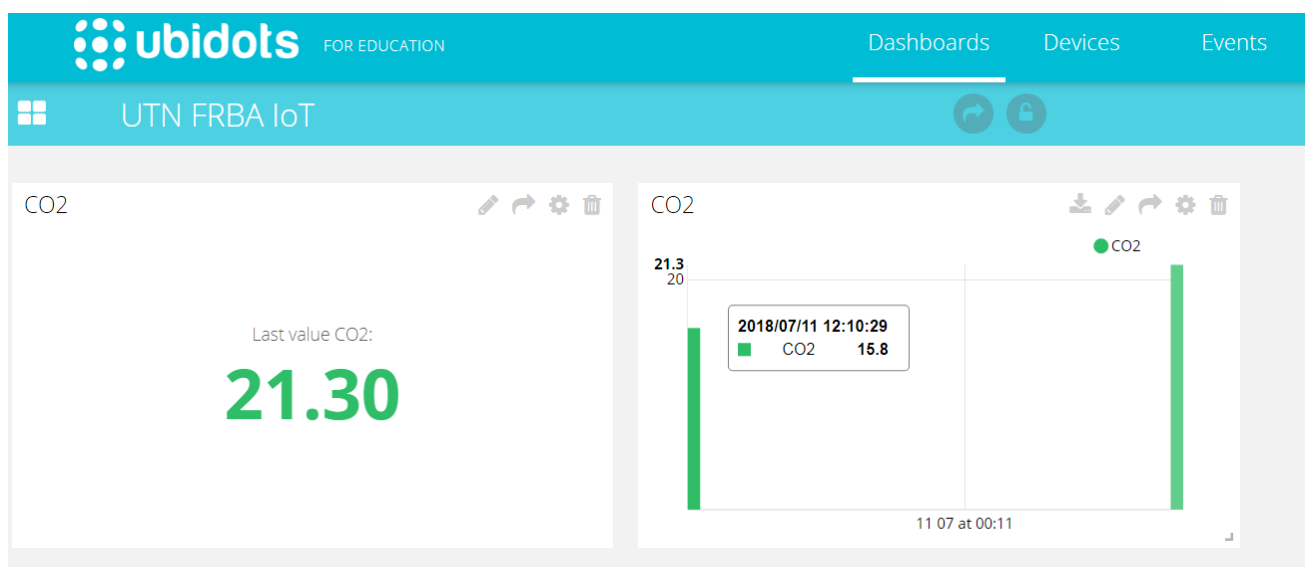


```
C:\Program Files (x86)\mosquitto>mosquitto_pub -h "things.ubidots.com" -p "1883" -u "A1E-[redacted]" -t "/v1.6/devices/UTNDispositivo/CO2" -m "15.8"

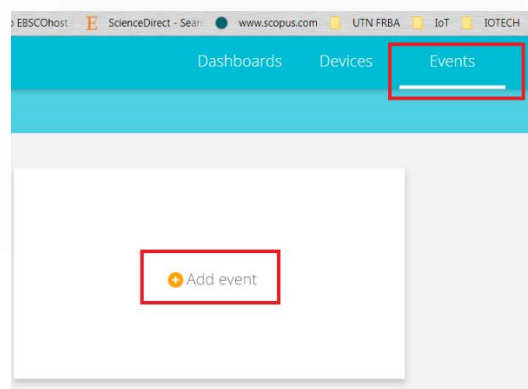
C:\Program Files (x86)\mosquitto>mosquitto_pub -h "things.ubidots.com" -p "1883" -u "A1E-[redacted]" -t "/v1.6/devices/UTNDispositivo/CO2" -m "21.3"

C:\Program Files (x86)\mosquitto>
```

Si desea ver la evolución temporal, podría agregar un nuevo **WIDGET** donde observaría.



Existe la posibilidad de poder programar eventos, esto es, acciones que se ejecutaran si se cumplen con ciertas características, por ejemplo, envío de emails o SMS a números de celular.



Indicamos primero la toma de decisión



Select a Device

unfrabiot UTNDIspositiv DHT11 monitoreo switch

Select a Variable

CO2

if

CO2

less greater less or equal greater or equal equal

than

50

Indicando por ejemplo Destinatario de email, y demás características del mismo

Send e-mail Send SMS Send Telegram WebHook Set a variable

+ Add email

Email marcospoliti@outlook.com

Subject CO2 > 50

Message HI marcospoliti, variable CO2 is greater than 50

Finish

Procediendo como antes con cliente **mosquitto**, cargando un valor mayor a 50 nos encontramos en nuestra cuenta de email el siguiente mensaje.



Nuevo | Responder | Eliminar Archivar

CO2 > 50



Notifications Ubidots<service@ubidots.com>

Hoy, 09:53

Usted

Hi marcospoliti, variable CO2 is greater than 50

PUBLICACION EN UBIDOTS CON MQTT.fx

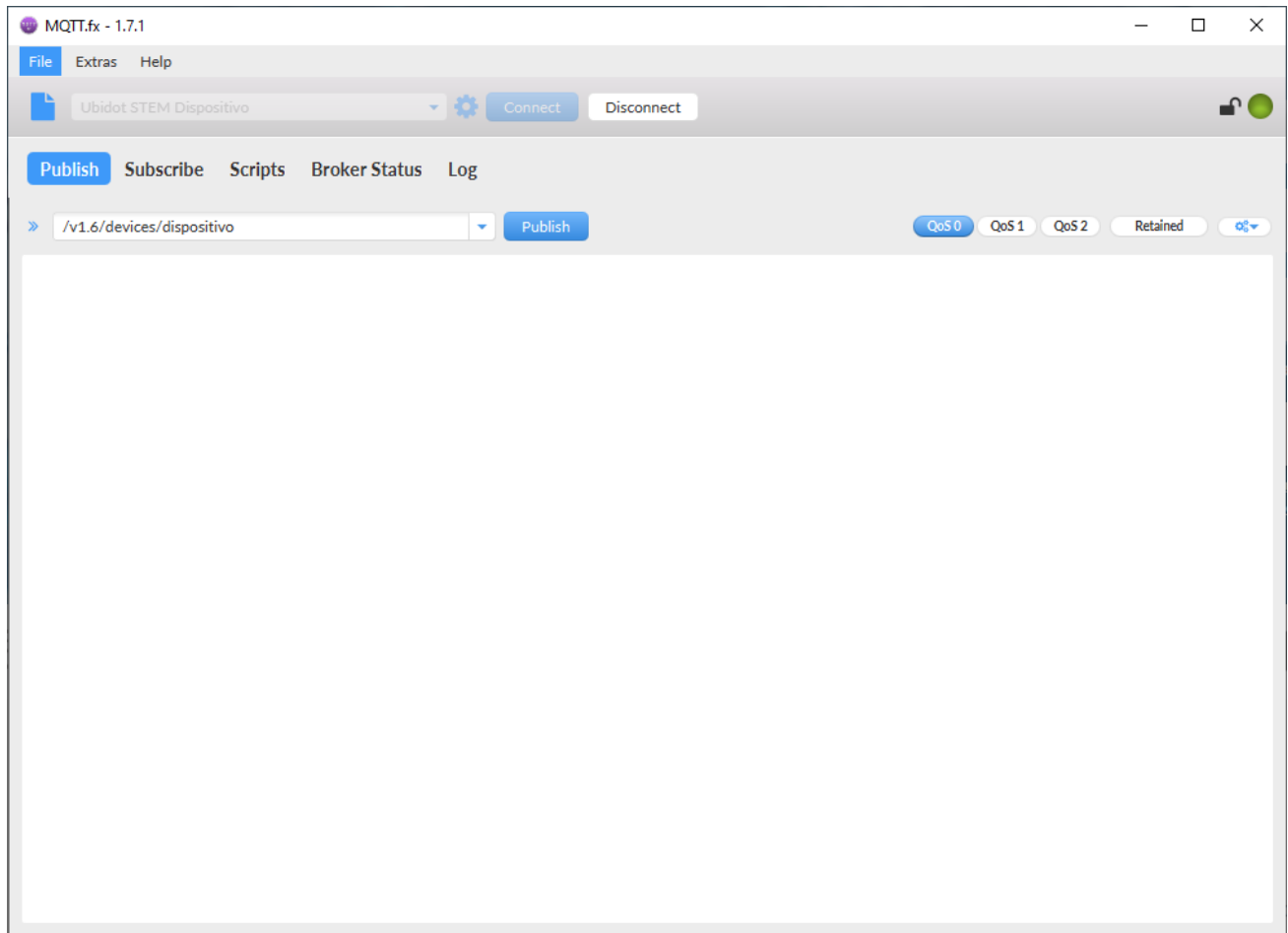
Primero generamos un DEVICE, en stem, e incluimos dentro de él dos variables var1 y var2 (en un comienzo las mismas no poseen datos ingresados, los valores 80 y 15 son producto del desarrollo adecuado de los pasos que siguen.)

The screenshot displays the Ubidots 'Devices' page for a device named 'dispositivo'. The device has two variables: 'var1' with a value of 80.00 and 'var2' with a value of 15.00, both recorded 3 hours ago. The interface includes a map, a 'SET LOCATION' button, and an 'Add Variable' button at the bottom.



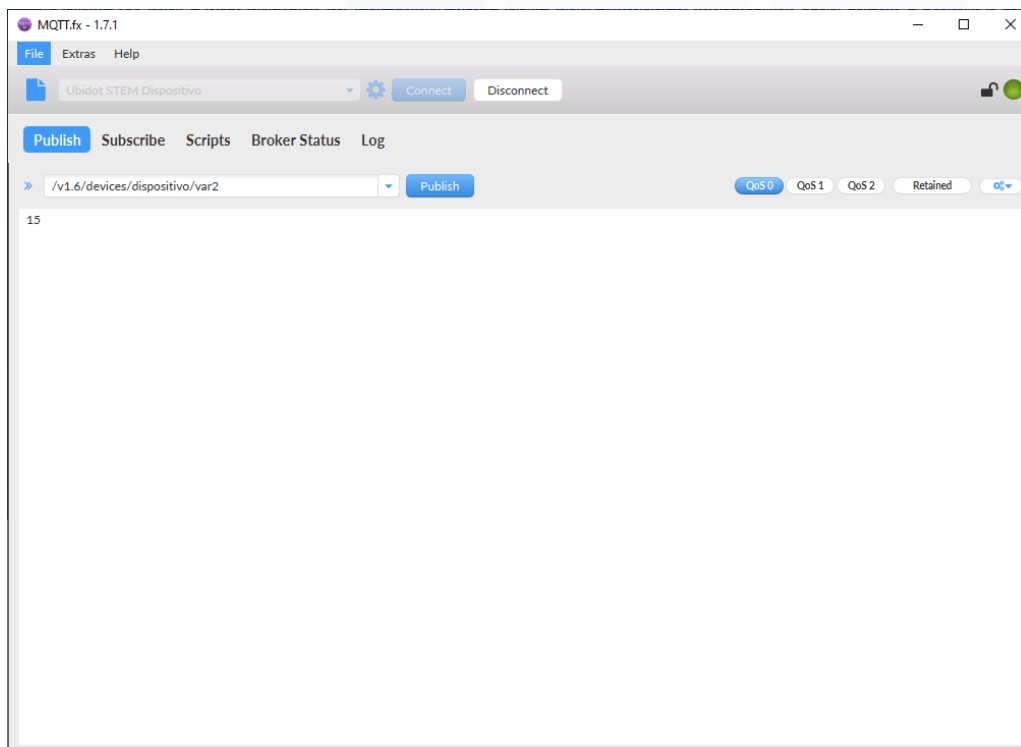
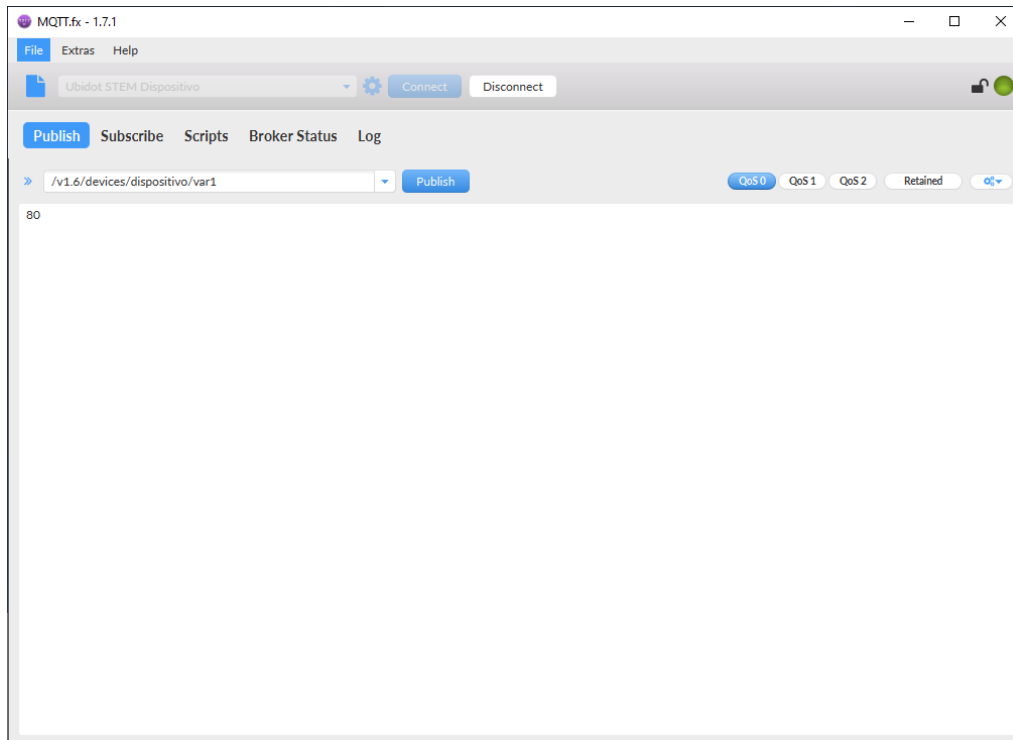
De manera similar que en Mosquitto, en cuanto a la configuración, aunque de una manera mucho más dinámica configuramos un nuevo perfil de conexión en MQTT.fx.

Guardamos y vamos a la pantalla principal y nos conectamos,



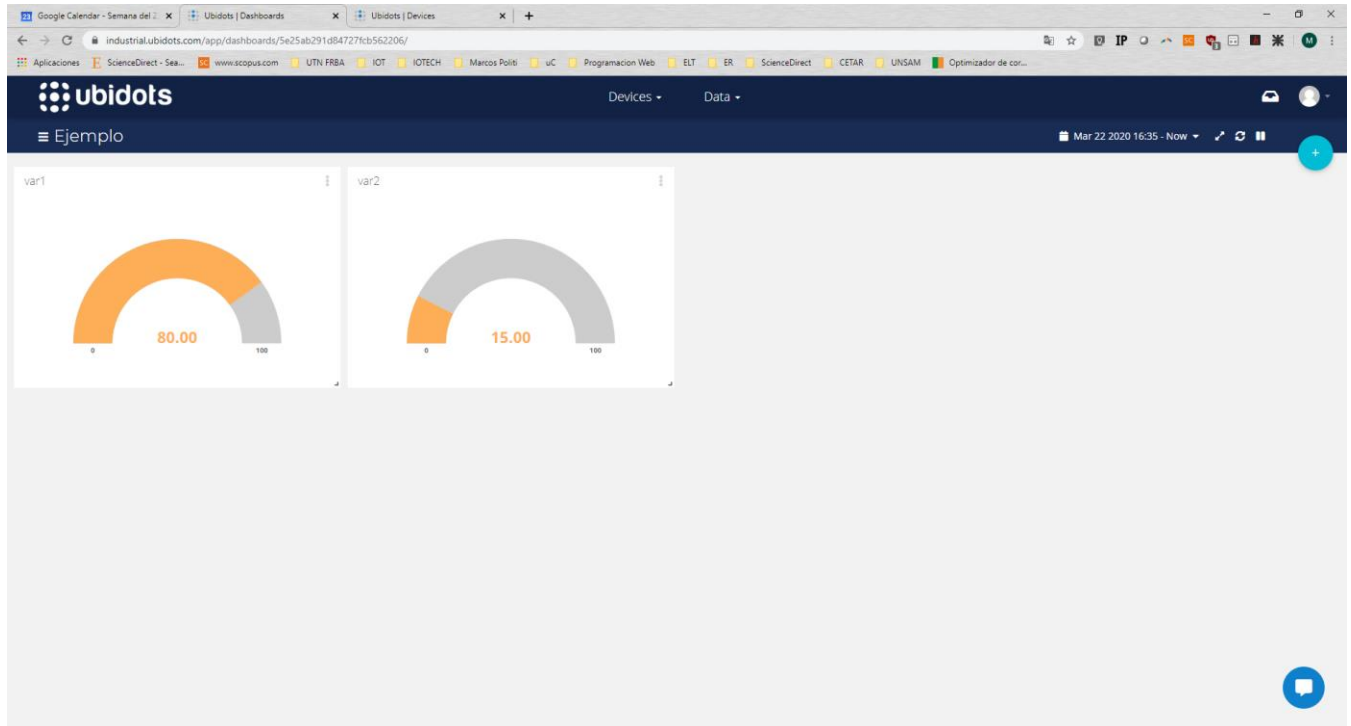


Aquí publicaremos a los topic correspondientes





quedando



Mg.Ing. Marcos Politi



REFERENCIAS

[1] Documentos, Maestría en Internet de las Cosas Universidad de Salamanca.

[2] <https://ubidots.com/docs/>

[3] <https://io.adafruit.com/api/docs/#section/Authentication>

[4] <http://mqtt.org/>