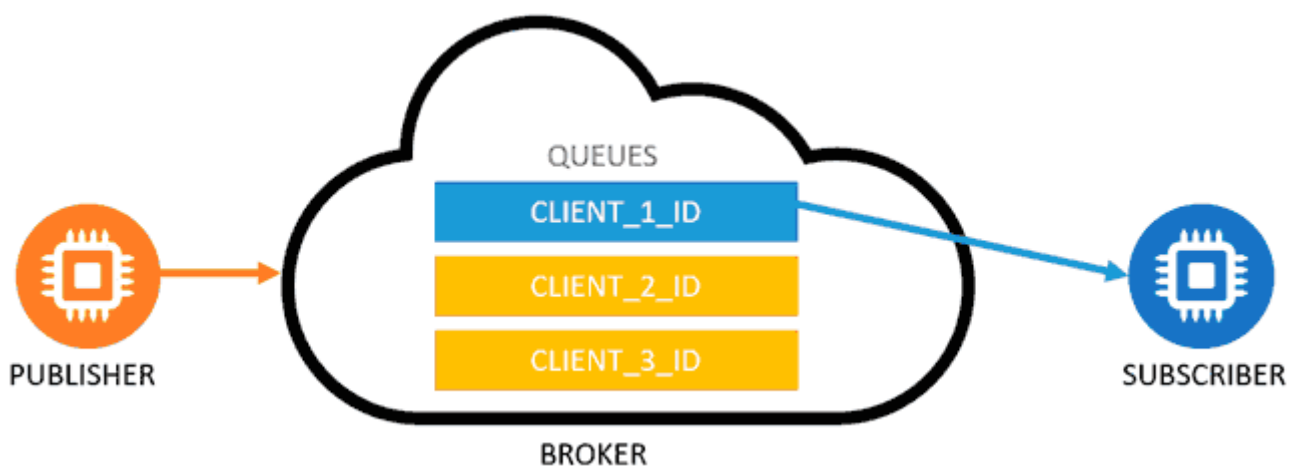


IoT-Wifi-MQTT

- [MQTT](#)
- [IoT - Wifi- MQTT - ADAFRUIT](#)
- [IoT-Wifi-MQTT-ADAFRUIT-FEED](#)
- [IoT-Wifi-MQTT- ADAFRUIT-Dashboard](#)
- [IoT-Wifi-MQTT-ADAFRUIT-ARDUINOBLOCKS](#)

MQTT

MQTT es un protocolo dentro de los protocolos entre máquina a máquina M2M. MQ significa Message Queue, son mensajes que se publican y otros se suscriben. Llegan a un servidor que también se llama **Broker** o Router, los va almacenando en una cola y se eliminan cuando son entregados. Un ejemplo fácil de entender son los mensajes SMS, que mientras no estas conectado, tus mensajes están en un servidor, y cuando te conectas, se descargan tus mensajes en tu móvil. Las últimas siglas TT significa Telemetry Transport



Fuente Luis Llamas CC-NC-BY-SA <https://www.luisllamas.es/protocolos-de-comunicacion-para-iot/>

“ Los clientes inician una conexión TCP/IP con el broker, el cual mantiene un registro de los clientes conectados. Esta conexión se mantiene abierta hasta que el cliente la finaliza. Por defecto, MQTT emplea el puerto 1883 y el 8883 cuando funciona sobre TLS.

Fuente Luis Llamas CC-NC-BY-SA <https://www.luisllamas.es/protocolos-de-comunicacion-para-iot/>

TCP/IP son los protocolos de comunicación en Internet. TLS Transport Layer Security es un protocolo que se añade para asegurar la privacidad y seguridad de los datos. Por eso hay que enviar un usuario y una contraseña para conseguir esta seguridad.

En los mensajes, se añade un texto que haga de filtro, que se llama **topic**, este topic permite al broker filtrar de todos los mensajes que se reciben a qué clientes hay que entregar estos

mensajes. Estos topic suelen estar jerarquizados, por ejemplo Casa/Cocina/Temperatura y un cliente puede estar suscrito a Casa/+/Temperatura por lo que recibiría los datos de Casa/Cocina/Temperatura pero también los de Casa/Salon/Temperatura, etc... Si quieres saber más de los tópic te recomendamos la página <https://www.luisllamas.es/que-son-y-como-usar-los-topics-en-mqtt-correctamente/>

El MQTT se ha hecho muy popular en los dispositivos electrónicos, porque:

- Es **libre**, al principio en 1999 era un protocolo propietario y se usaba para la industria petrolera, pero desde el 2010 se liberó y su consecuente popularidad que se ha convertido en un estándar de comunicación electrónica.
- Es **sencillo**, esto es importante para no pedir mucho ancho de banda
- Consume **poca potencia**, esto es importante pues la mayoría están conectados todo el tiempo.
- Un protocolo con una capa de **seguridad**, robusta y largamente testada.

Para saber más te recomendamos la página de Luis Llamas <https://www.luisllamas.es/protocolos-de-comunicacion-para-iot/>

Y el broker ¿de donde?

Puedes montarte tu propio broker en un ordenador, pues es código abierto, por ejemplo instalar Mosquitto <https://mosquitto.org/> en una raspberry.

O bien usar brokers MQTT gratuitos y públicos en Internet, con sus limitaciones de anchos de banda o número de conexiones... como HIVEMQ Info: <http://www.mqtt-dashboard.com/> con la dirección Broker.hivemq.com o también el propio Mosquito <https://mosquitto.org/> con la dirección test.mosquitto.org

Tenemos que tener en cuenta que los servidores MQTT públicos gratuitos no son seguros, cualquiera puede suscribirse a nuestros mensajes y publicar en ellos por lo tanto enviar órdenes. Evitar usar estos brokers en sistemas que no sean educativos y de pruebas.

El protocolo MQTT también se usa en Domótica, por ejemplo [aquí en el curso de Raspberry apartado domótica](#)

<https://www.youtube.com/embed/YPS5xI6Bx3I>

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU

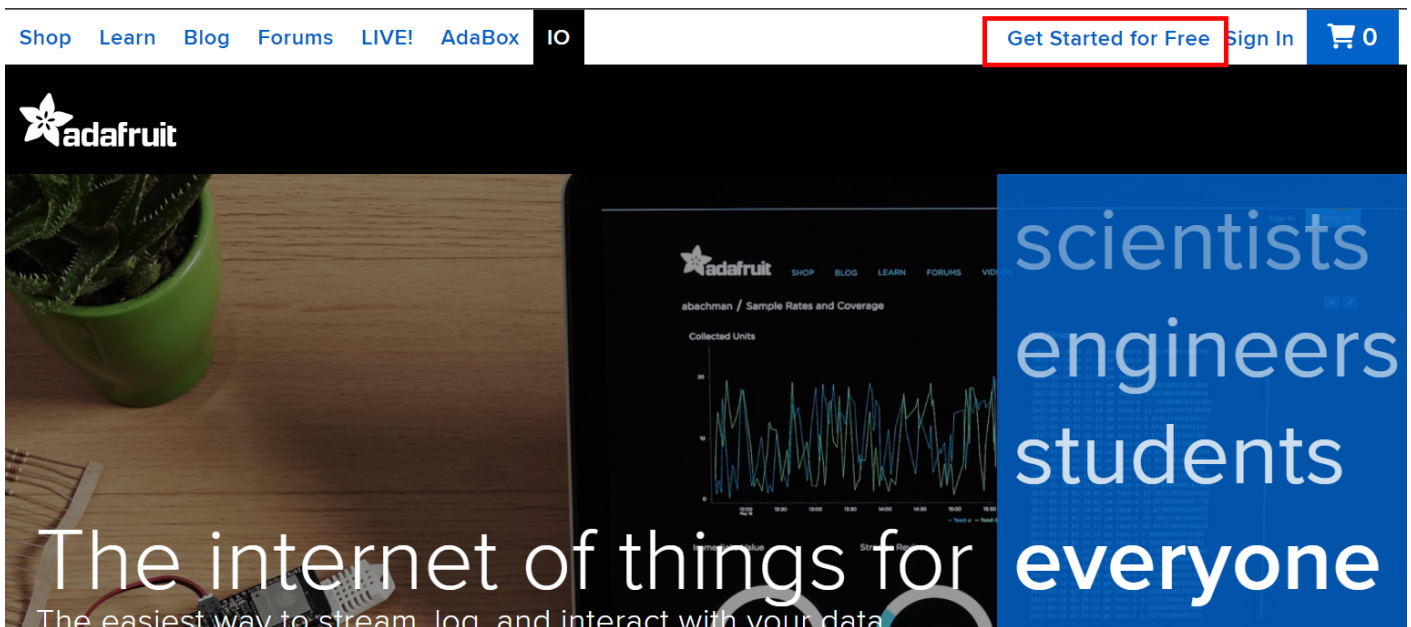


IoT - Wifi- MQTT - ADAFRUIT

Esta compañía de circuitos electrónicos se hizo famosa por sus **LEDS NEOPIXEL** que lo vimos [en el curso de mClon con nanoArduino](#), dedica parte de su producción a la didáctica y vamos a utilizar **los servicios MQTT** que ofrece gratuitamente. *(Esto es típico, por ejemplo en los cursos de mBot utilizamos el software mBlock de la compañía Makeblock.)*

Registro

Nos registramos dentro del plan gratuito de su plataforma <https://io.adafruit.com/>



El registro es típico, pidiendo un nombre de usuario, una contraseña y un correo electrónico para recuperar la contraseña.

Utiliza un nombre de usuario lo más sencillo posible, pero a la vez original, pues se convertirá parte de la URL del resto de acciones que vas a realizar.

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU



IoT-Wifi-MQTT-ADAFRUIT-FEED

Vamos primero a definir las variables que vamos a utilizar, o como lo llama ADAFRUIT la "comida" **FEED**, en estos FEED se almacenarán las salidas de nuestro Arduino que utilizaremos el sensor de humedad y temperatura y también se almacenará la entrada para encender un led. Esto lo tenemos que interpretar según el protocolo MQTT que hemos visto, como **colas de mensajes** que el proveedor almacena y el cliente se suscribe.

En el caso de los canales de humedad y temperatura, el proveedor será nuestro Arduino y el cliente será un **dashboard** que crearemos dentro de Adafruit. En el caso del led, el proveedor será el dashboard y el cliente será el Arduino.

Entramos en feed, creamos un **grupo** que le ponemos un nombre acorde al conjunto de feeds que tendrá (esto es útil si tenemos diferentes placas o el lugar donde están esas placas: cocina, salón, ... y dentro de cada grupo los diferentes sensores y actuadores: temperatura, persianas, luz....

Una vez creado el grupo, procedemos a crear **3 feeds** : hum, temp y led

io.adafruit.com/javierquintana/feeds

Shop Learn Blog Forums LIVE! AdaBox IO

Hi, JAVIER QUINTANA PEIRO | Account

adafruit Devices Feeds Dashboards Actions Power-Ups New Device

javierquintana / Feeds Help

New Feed New Group

Feed Name	Key	Last value	Recorded
<input type="checkbox"/> hum	hum	62.00	3 days ago
<input type="checkbox"/> led	led	apagado	about 15 hours ...
<input type="checkbox"/> temp	temp	22.00	3 days ago

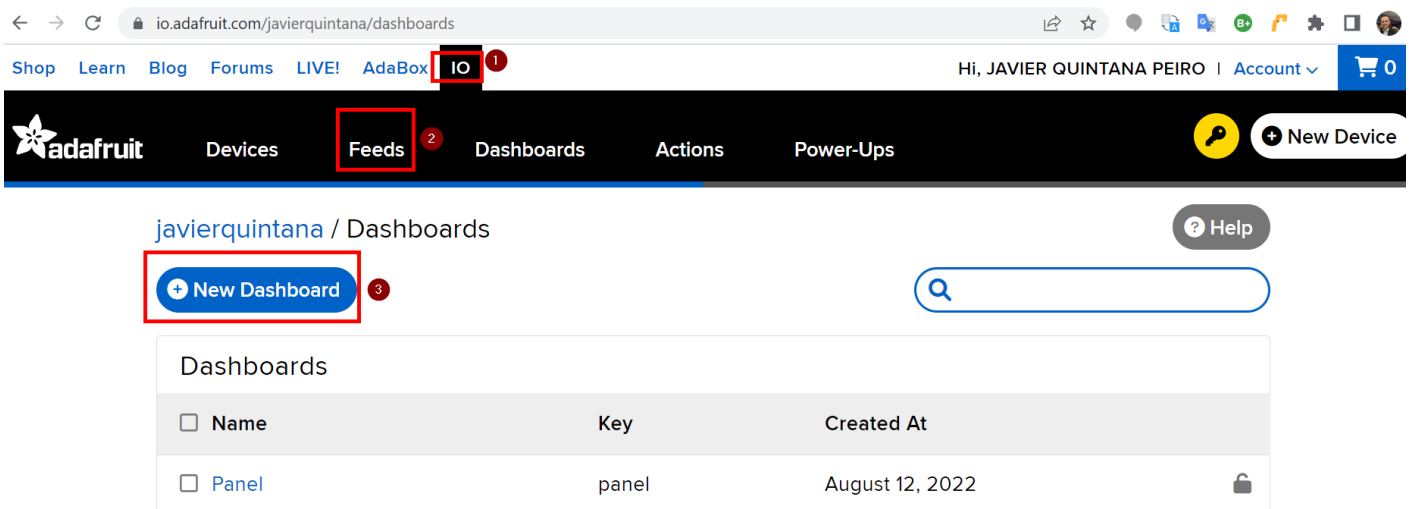
Como se puede ver, la URL es personalizada con el nombre de usuario, que para mi caso es <https://io.adafruit.com/javierquintana/feeds> (no intentes entrar, tú no verás nada, yo sí)

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU



IoT-Wifi-MQTT- ADAFRUIT- Dashboard

Ahora vamos a crear un panel, para ello entramos en IO - DASHBOARD y creamos un panel



javierquintana / Dashboards Help

+ New Dashboard 3

<input type="checkbox"/> Name	Key	Created At
<input type="checkbox"/> Panel	panel	August 12, 2022 🔒

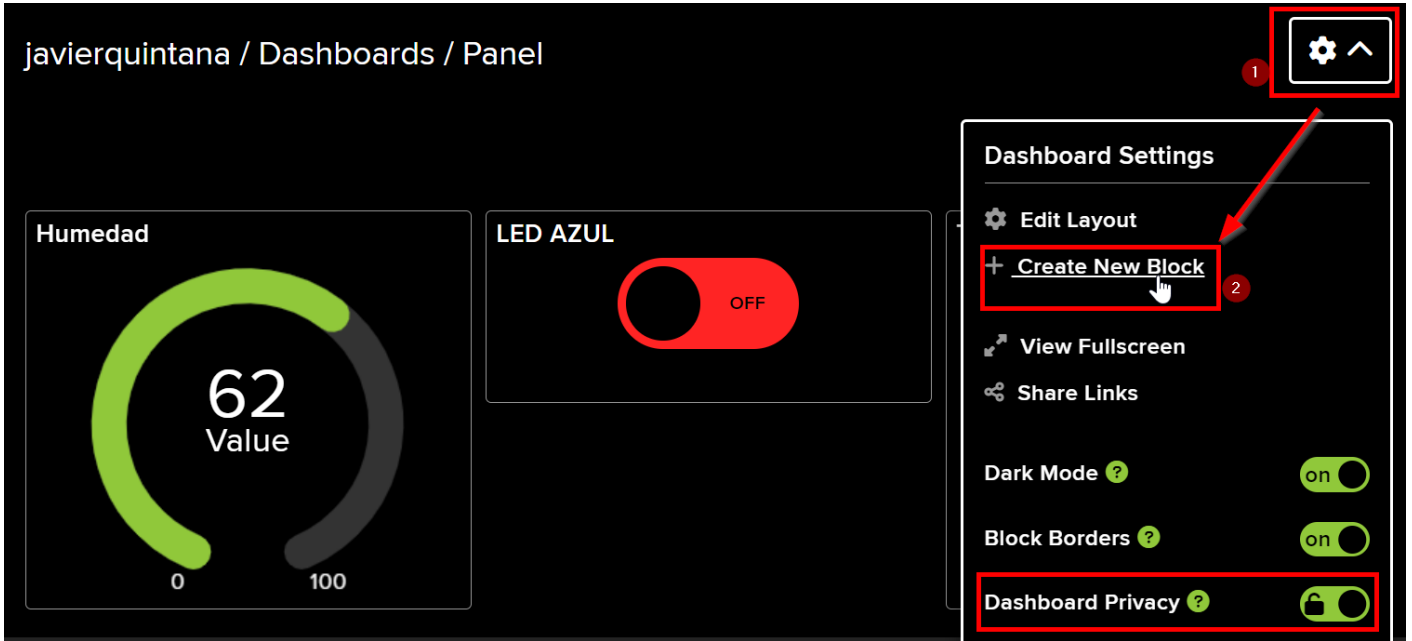
Editando el Dashboard

Dentro del panel podemos hacerlo público, el mio es este

<https://io.adafruit.com/javierquintana/dashboards/panel>

Y podemos crear los bloques que nosotros utilizaremos los de tipo gauge y tipo switch (marcados en la figura)

javierquintana / Dashboards / Panel



The image shows a dashboard interface with a settings menu open. The dashboard contains two widgets: 'Humedad' (Humidity) with a gauge showing a value of 62, and 'LED AZUL' (Blue LED) with a red toggle switch currently in the 'OFF' position. The settings menu, titled 'Dashboard Settings', is located on the right side and includes the following options: 'Edit Layout', '+ Create New Block', 'View Fullscreen', 'Share Links', 'Dark Mode' (on), 'Block Borders' (on), and 'Dashboard Privacy' (on). Red boxes and numbers highlight the settings menu icon (1) and the '+ Create New Block' option (2).

Humedad

62
Value

0 100

LED AZUL

OFF

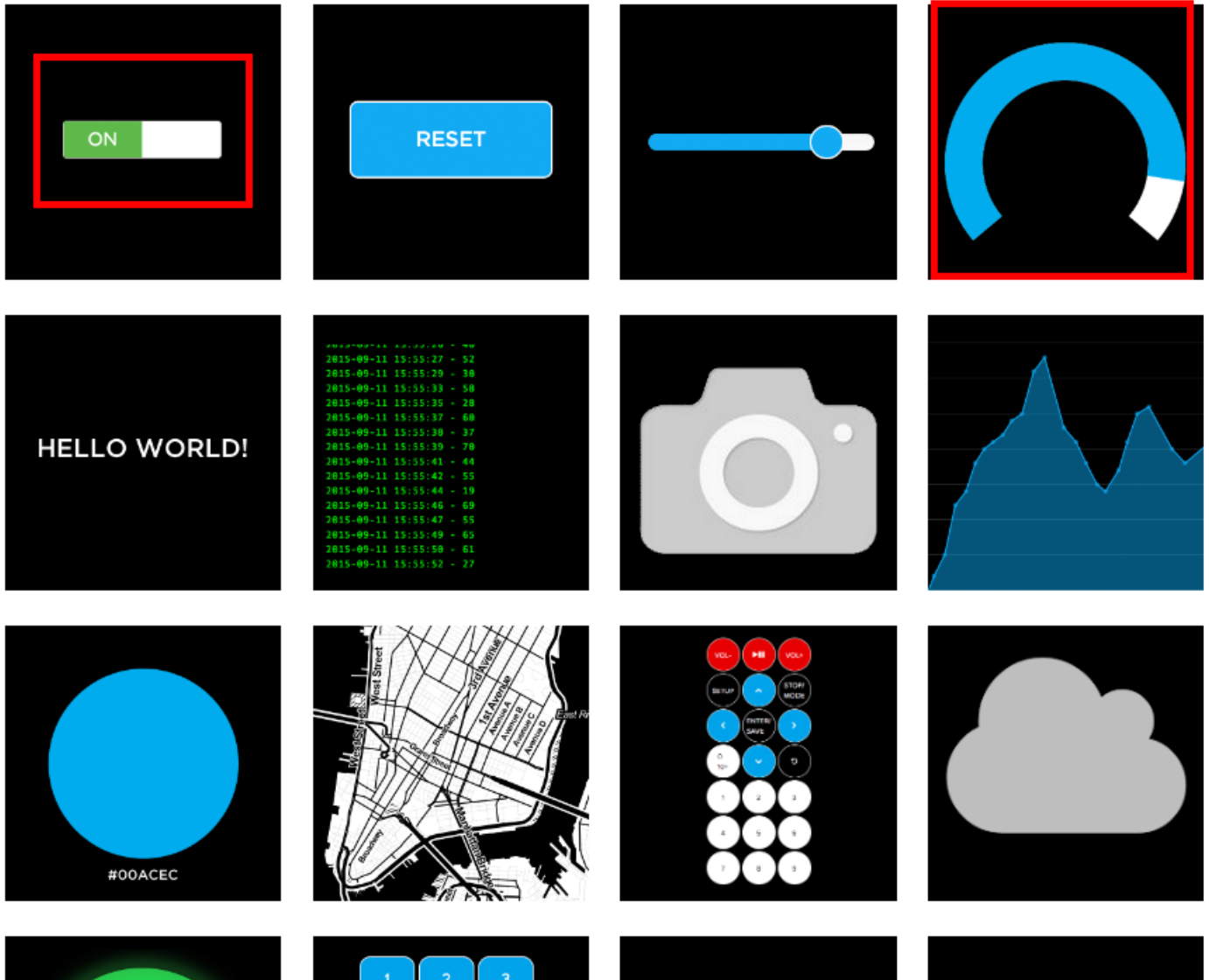
Dashboard Settings

- Edit Layout
- + Create New Block
- View Fullscreen
- Share Links
- Dark Mode ? on
- Block Borders ? on
- Dashboard Privacy ? on

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.



Cada vez que creas un bloque, pide a qué **feed** lo asocias :

Connect a Feed ✕

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

Choose a single feed you would like to connect to this gauge. You can also create a new feed within a group.



Mi-TDR-STEAM-IMAGINA ▼

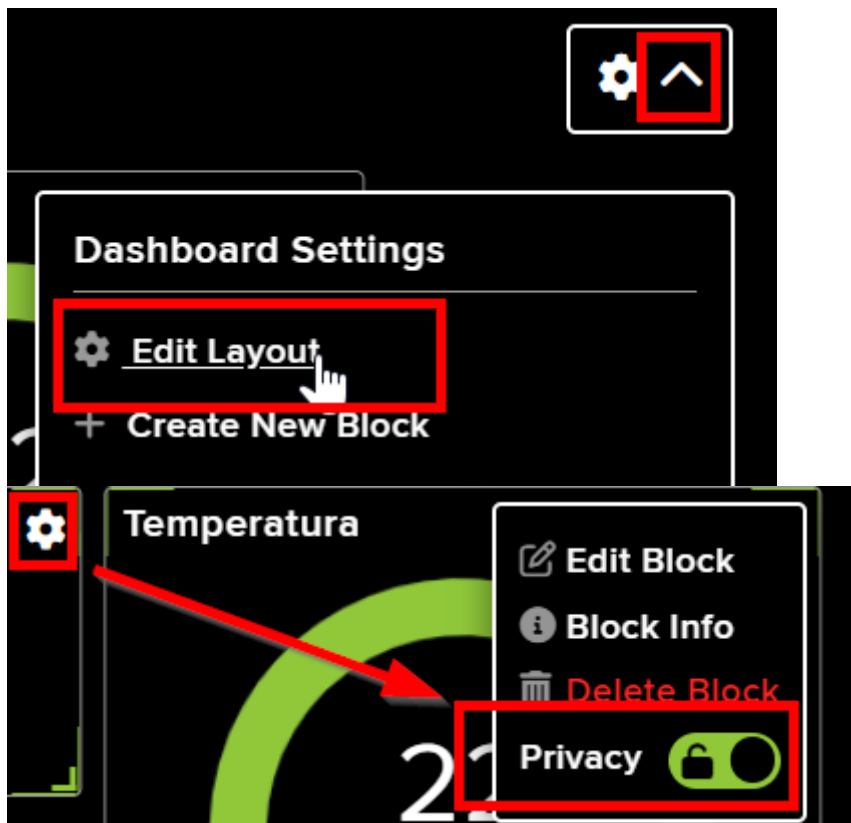
Feed Name	Last value	Recorded	
<input type="checkbox"/> hum	62.00	3 days	
<input type="checkbox"/> led	apagado	about 16 hours	
<input type="checkbox"/> temp	22.00	3 days	

0 of 1 feeds selected

< Previous step

Next step >

Los bloques puedes definir que sean **públicos**:



Bloques de los sensores de visualización (El Arduino como proveedor, el Dashboard como cliente)

Vamos a crear dos bloques:

- **Humedad** tipo de bloque **gauge** asociado al feed **hum**
- **Temperatura** tipo de bloque **gauge** asociado al feed **temp**

Y pondremos las propiedades de los bloques HUMEDAD Y TEMPERATURA acordes a las características de los sensores:

Block Title (optional)

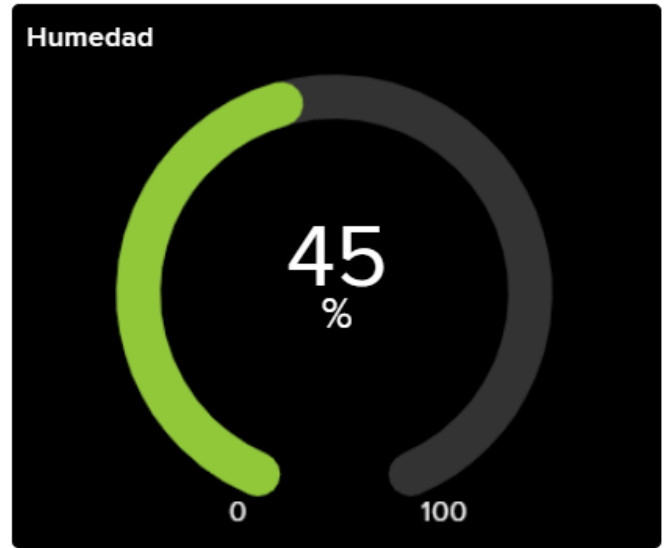
Gauge Min Value

Gauge Max Value

Gauge Width

Gauge Label

Block Preview



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

Block Title (optional)

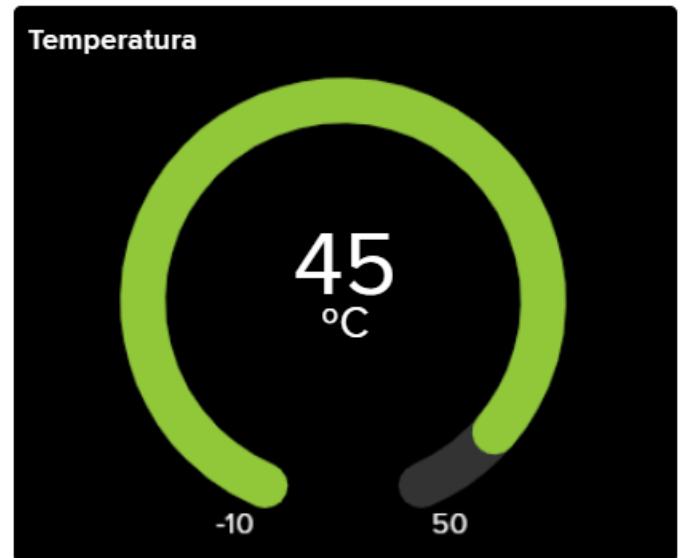
Gauge Min Value

Gauge Max Value

Gauge Width

Gauge Label

Block Preview



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

Bloques de acción (El Arduino como cliente, el Dashboard como proveedor)

Ahora vamos a crear un **bloque LED AZUL** tipo **switch** asociado al feed **led**

Definimos en las propiedades de este bloque que vamos a enviar:

- Cuando está OFF vamos a enviar la palabra "**apagado**"
- Cuando esta ON vamos a enviar la palabra "**encendido**"

Block Title (optional)

Button On Text

Limit of 6 characters for the toggle text. Use the block title to be more descriptive.

Button On Value (uses On Text if blank)

Button Off Text

Limit of 6 characters for the toggle text. Use the block title to be more descriptive.

Button Off Value (uses Off Text if blank)

Block Preview



Toggle A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent on press and release.

Test Value

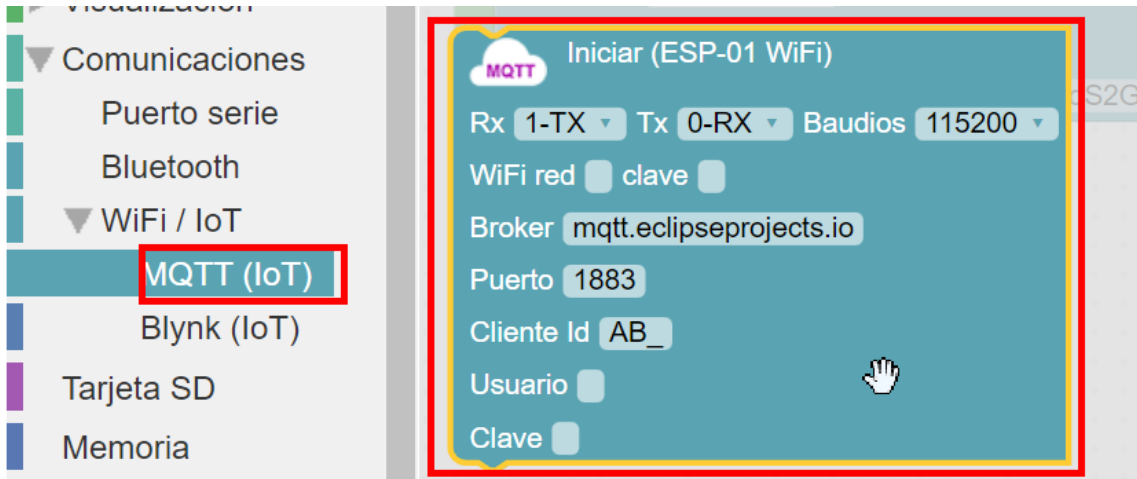
Los bloques de acción si los haces públicos **se pueden ver pero no se pueden tocar**, para que gente no autorizada actúen sobre las placas. Por eso en <https://io.adafruit.com/javierquintana/dashboards/panel> NO PUEDES ACCIONARLO

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU



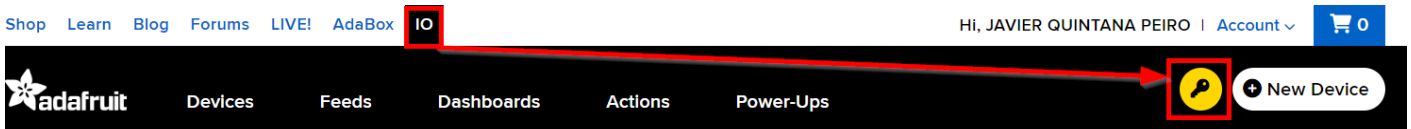
IoT-Wifi-MQTT-ADAFRUIT-ARDUINOBLOCKS

En ARDUINOBLOCKS seleccionamos Comunicaciones -IoT.MQTT y el bloque de Iniciar ESP01



Con estos datos:

- Definir RX al pin D5 y TX al pin D3 a 9600 baudios como lo hemos visto anteriormente
- El nombre de nuestra red wifi y su contraseña
- El broker será el servidor de **adafruit io.adafruit.com**
- El puerto, será el **1883** tal y como hemos visto anteriormente
- Como cliente ID puedes poner cualquier cosa
- Usuario : el usuario de nombre de usuario de adafruit, que en mi caso es javierquintana
- La clave lo tienes aquí



Así tiene que quedar:

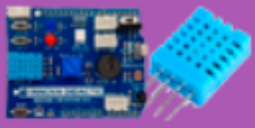



El programa buble hacemos:



- Creamos un buble que se ejecute cada 10 segundos, luego explicamos el por qué.
- Leemos el sensor de humedad y temperatura y lo almacenamos en una variable tipo numérica **Hum** y **Temp**
- Las variables Hum y Temp las **publicamos** en el tema asociado a tu usuario y a tu feed
- Creamos **una variable texto LED-TXT**
- Nos **suscribimos** al feed led y lo almacenamos en la variable texto creada
- Si la variable texto tiene la palabra encendido pues encendemos el led, apagamos en caso contrario



Bucle



Ejecutar cada **10000** ms

Establecer **Hum** =  **DHT-11** **Humedad %**


Establecer **Temp** =  **DHT-11** **Temperatura °C**


 **Publicar Tema**  **adafruit.io publish** **Valor** **Temp**
User **javierquintana**
Feed Key **temp**

 **Publicar Tema**  **adafruit.io publish** **Valor** **Hum**
User **javierquintana**
Feed Key **hum**

 **Suscribir Tema**  **adafruit.io subscribe** **>** **LED-TXT**
User **javierquintana**
Feed Key **led**

+ si **LED-TXT** **igual a** **“ encendido ”**

hacer  **Led Azul** **Estado ON**

sino  **Led Azul** **Estado OFF**



Lo grabamos en el Arduino con la placa TDR-STEAM y el ESP01 conectado como hemos definido anteriormente y si entramos en nuestro panel, veremos que visualiza la temperatura y humedad de ese momento. Si tenemos un poco de paciencia, pues sólo nos hace caso cada 10 segundos, actuamos sobre el switch y vemos que enciende o apaga el led.

¿Por qué cada 10 segundos?

Si entramos en nuestro perfil contratado, el plan gratuito permite **sólo 30 datos por minuto**, si **no tocamos** el switch de encender el led, podríamos bajar esos 10 segundos a 4 segundos pues 2 gauges temperatura y humedad * $(60/4) = 30$ datos por minuto pero en el momento que toquemos el switch te penalizará a esperarte pues te has pasado de 30 datos por minuto.

Para poder accionar sobre el switch, subimos esos 4 segundos a 10, esto nos da 2 gauges temperatura y humedad * $(60/10) = 12$ datos por minuto, luego podemos accionar el switch $30-12=18$ veces por minuto.

Puedes bajar de 10 pero cuanto más bajas, menos puedes tocar el switch, tú mismo.

Financiado por el Ministerio de Educación y Formación Profesional y por la Unión Europea - NextGenerationEU

