


PLUG 'EM ALL



APRENDIENDO  EL INTERNET  DE LAS COSAS



PEQUEÑA  FORMATIVA

 ALEJANDRO JUAN GARCÍA



@ALEXCORVIS84
@MAKERSASTURIAS

media
lab_

⚠️ DISCLAIMER ⚠️

- ESTO NO ES MI TRABAJO ✕ 💰 ¡ES LO QUE ME APASIONA 😍!
- OPINIONES PROPIAS BASADAS EN LA FILOSOFÍA 'JUAN PALOMO' 🧠 ⚡
- HE APRENDIDO GRACIAS A LA AYUDA DE OTROS 👥
- NO ES UN ANUNCIO O PROMOCIÓN DE NINGUNA COMPAÑÍA 🗣️ ✕
- MATERIAL DISPONIBLE BAJO LICENCIA CC BY-SA 📖
- MATERIAL USADO OPEN HARDWARE 🔧 ❤️ y OPEN SOURCE ⚙️ ❤️
- MATERIAL BASADO EN RECURSOS EXTERNOS Y PROPIOS 📚 🌐
- INTENTAR DIVULGAR LA CULTURA MAKER Y FILOSOFÍA DIY 🔧 🧐



Internet de las cosas (IoT)

CONCEPTOS

Internet de las cosas (*Internet Of Things – IOT*)

Definición:

Consiste en dotar a objetos cotidianos una interconexión digital gracias a Internet, permitiendo el intercambio de información con otros dispositivos (1999, Kevin Ashton, director centro Auto-ID del MIT)



¿Cómo funciona un sistema IOT? 🤖

Un sistema IOT integra 4 componentes claramente distintos:



- Dispositivos/Sensores
 - Protocolos ⚠️
- Conectividad
 - Tecnologías 📶
- Procesamiento de datos
 - Plataformas IoT
- Interfaz de usuario

Dispositivos/Sensores

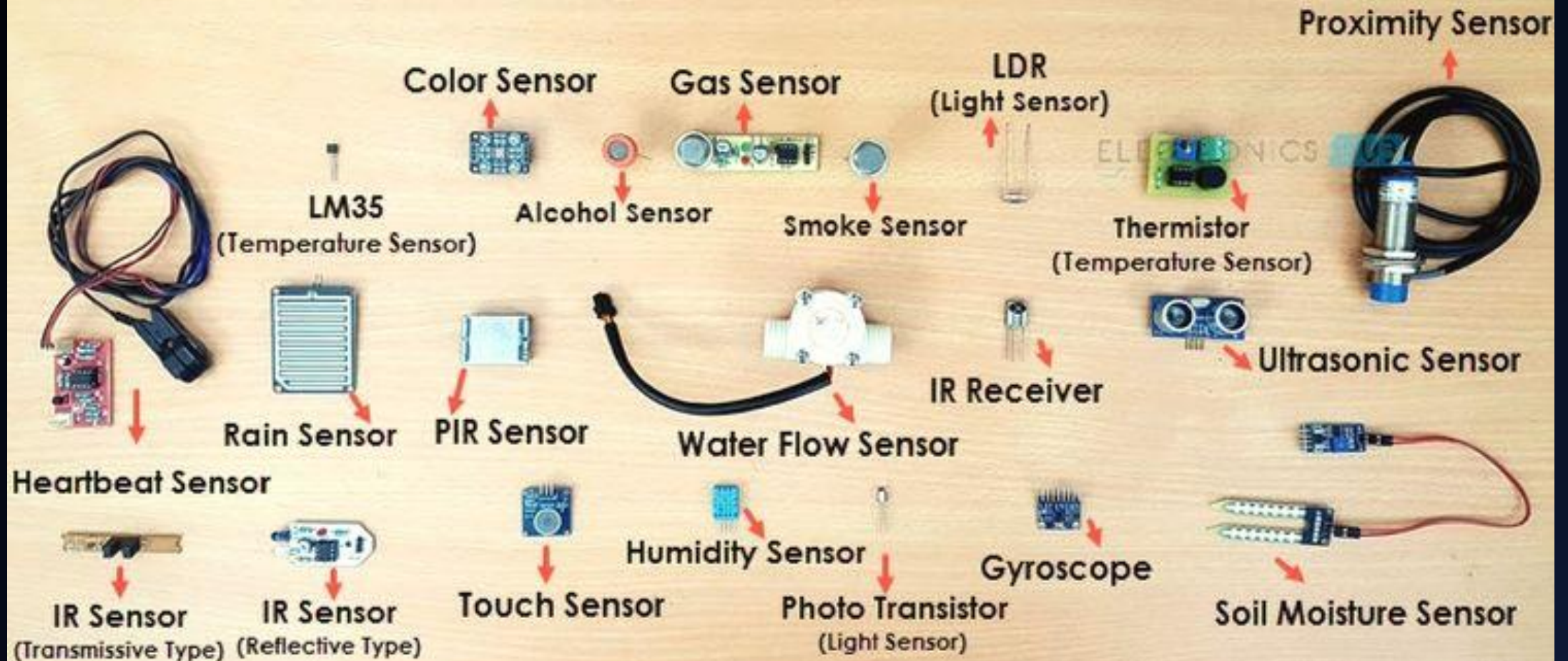
- Son los encargados de recolectar datos del entorno en el que se encuentran localizados
- Dispositivos de entrada que proporcionan una salida (señal) con respecto a una cantidad física específica (entrada)
- Convierten las señales de un dominio de energía a un dominio eléctrico
- De todo tipo y variedad
- Cada aplicación requerirá de su estudio y selección final

Dispositivos/Sensores

Clasificación:

- Activos y pasivos (señal de excitación externa)
- Medios de detección
 - Eléctricos, químicos, biológicos, radioactivos...
- Fenómenos de conversión (E/S)
 - Fotoeléctricos, termoelectrónicos, electroquímicos...
- Analógicos y Digitales
 - Señal continua en el tiempo 
 - Señal discreta/digital 

DIFFERENT TYPES OF SENSORS



Protocolos

- Definen cómo se comunican los dispositivos/sensores
- Existe una gran variedad de protocolos de comunicación
- Definidos para uso doméstico o industrial
- Problema: Estandarización
- Existen protocolos privados y de código abierto

AMQP

ZigBee

Sigfox

NFC

CoAP

Z-Wave

6LowPAN

RFID

DDS

Bluetooth

NB-IoT

LoRa

HTTP (REST/JSON)

Wi-Fi

Broadcom

...

MQTT

LoRaWAN

Digimesh

MQTT (Message Queue Telemetry Transport)

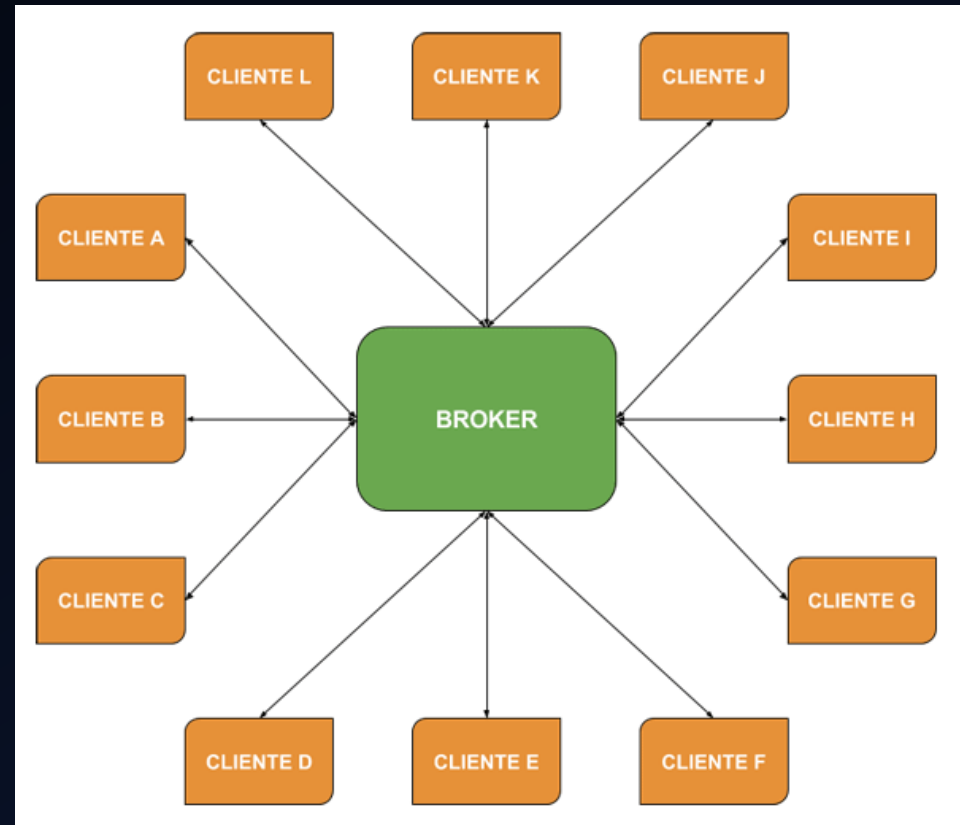
Protocolo de comunicación M2M ideal para el IOT

- ✓ Simple y ligero (consume poco ancho de banda)
- ✓ Bidireccional
- ✓ Fiable (Acuses de recibo)
- ✓ Comunicación basada en Publicación/Suscripción de mensajes
- ✓ OASIS estándar
- ✓ Código abierto
- ✓ 'Seguro'



MQTT (Message Queue Telemetry Transport)

- Topología en *Estrella*
- Nodo central: *Broker*
- *Publicación/Suscripción*
- Basada en mensajes asíncronos
- Topics
- Sintaxis de niveles (/,#,+)



Conectividad

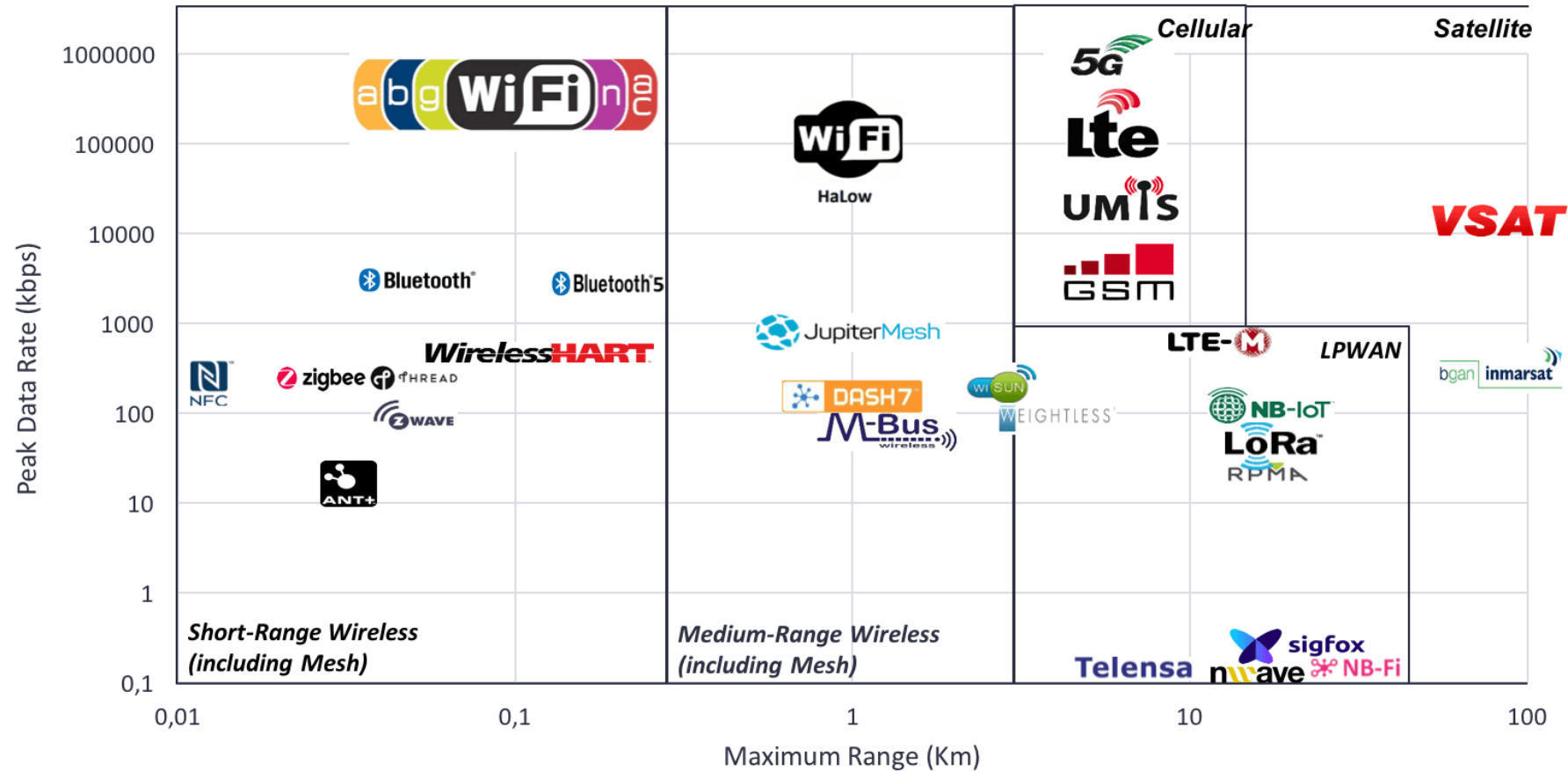
- Los datos recolectados necesitan ser enviados/transmitidos
- Existen varias tecnologías para dotar conectividad IOT
- Cada una posee ventajas y desventajas
- Su elección dependerá de la aplicación

Aspectos importantes para su elección:

- ✓ Consumo de energía 
- ✓ Rango de operación (distancia) 
- ✓ Ancho de banda 

Comparison Wireless technologies

Peak Data Rate vs Maximum Range



Please note that this chart is meant to show the maximum theoretical range and data rate for each technology, but this does not mean that the two can be achieved at the same time. On the contrary, no wireless technology can achieve the maximum range while transmitting at its peak data rate, but rather the higher is the used data rate, the lower is the achievable communication range.

Procesamiento de datos

- Una vez los datos son recibidos el software realiza algún tipo de procesamiento con ellos (ML, IA, Analítica...)
- Las plataformas IOT son la base para que los dispositivos estén interconectados y se genere un ecosistema
- Software que conecta hardware, puntos de acceso y redes de datos

APPLICATION



Data
storage/analytics



Consumer
application



Industrial
application



Business
application



Your
application

MIDDLEWARE



HARDWARE

Mobility



Tags / beacons



Sensors



Health and
fitness devices



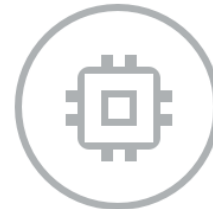
Consumer
electronics



Automotive



Embedded
hardware



The “Things”

*Primarily
analog data
sources*

*Devices,
machines,
people, tools,
cars, animals,
clothes, toys,
environment,
buildings, etc.*

Stage 1

Sensors/Actuators
(wired, wireless)



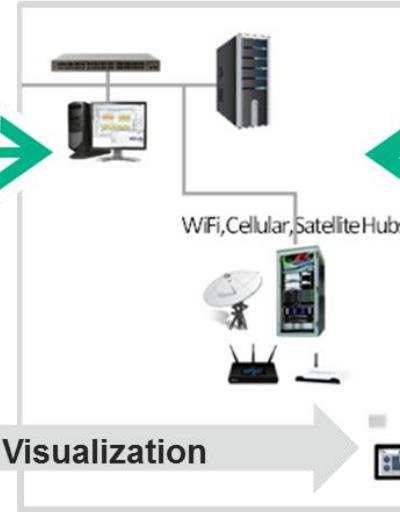
Stage 2

**Internet Gateways,
Access Points,
DAQ, Control**
(data aggregation, A/D,
measurement, control)



Stage 3

Edge IT
(analytics, pre-
processing)



Stage 4

Data Center / Cloud
(analytics,
management, archive)









The Edge

Visualization

SW Stacks:
Data Flow:
Control Flow:



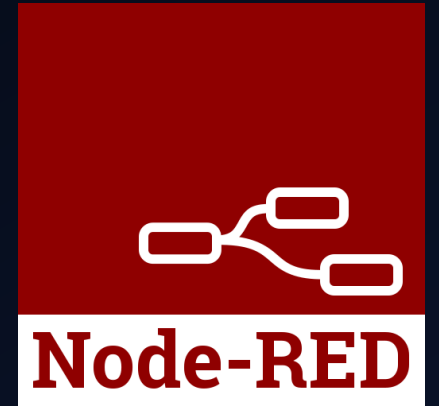
Interfaz de usuario

- La información es poder 
- El usuario final recibe ésta información de alguna manera
 - Notificaciones 
 - Alertas 
 - Correo electrónico 
 - SMS 
 - Dashboards 
- El usuario final puede configurar el sistema remotamente o realizar alguna acción sobre el mismo (OTA)

Node-RED

- Herramienta de programación visual para el IoT
- Desarrollada por IBM
- Opensource
- Basado en Node.js
- Ecosistema basado en paquetes (npm)
- Editor de flujos basado en navegador Web
- Permite conectar Hardware, APIs, y servicios online rápidamente
- Desarrollo social (flujos guardados como JSON)

“Pensada para pasar más tiempo pensando en cómo se comunican los dispositivos, más que preocuparme del código que necesitaría”



filter nodes

Termostato

Node-RED-Worldmap Ex

Flow 4

+

info

debug

dashboard

input

inject

catch

status

link

mqtt

http

websocket

tcp

udp

alexa home

Watson IoT

serial

Information

Flow "e98205dc.523138"

Name Flow 4

Status Enabled

Flow Description

None

PLUG 'EM ALL



WORKSHOP



APRENDIENDO EL INTERNET DE LAS
COSAS...



...A BASE DE PRÁCTICA!!!

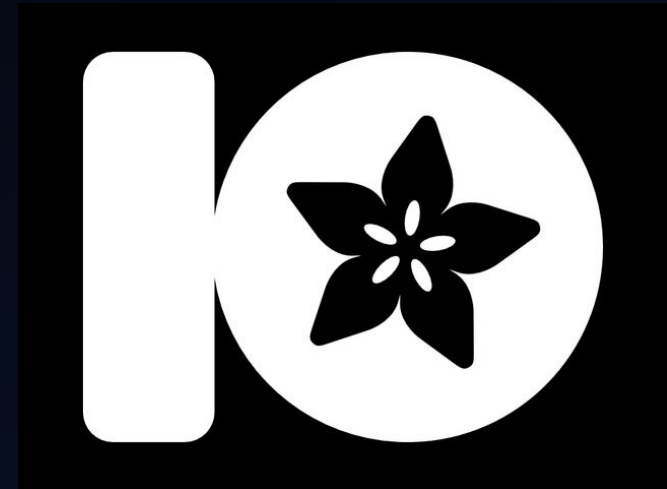


Adafruit IO

¿Qué es y qué nos permite?

Es un servicio en la nube (Cloud Service) de Adafruit Industries. Pretende ser una solución para la construcción de aplicaciones IoT.

- Recepción de datos online en tiempo real (**Feeds**)
- Visualización de datos en tiempo real (**Dashboards**)
- Los datos son **privados** (por defecto) y **seguros**
- Hacer que tu proyecto esté conectado a **Internet**: controlar motores, lectura de datos de sensores...
- **Conectar** proyectos a **servicios web** de terceros (**Triggers**)
- Conectar tu proyecto a otros **proyectos conectados** a Internet
- Servicio **gratuito** con posibilidad de adherirse a un plan de pago



IO Free

30 data points per minute

30 days of data storage

10 feeds

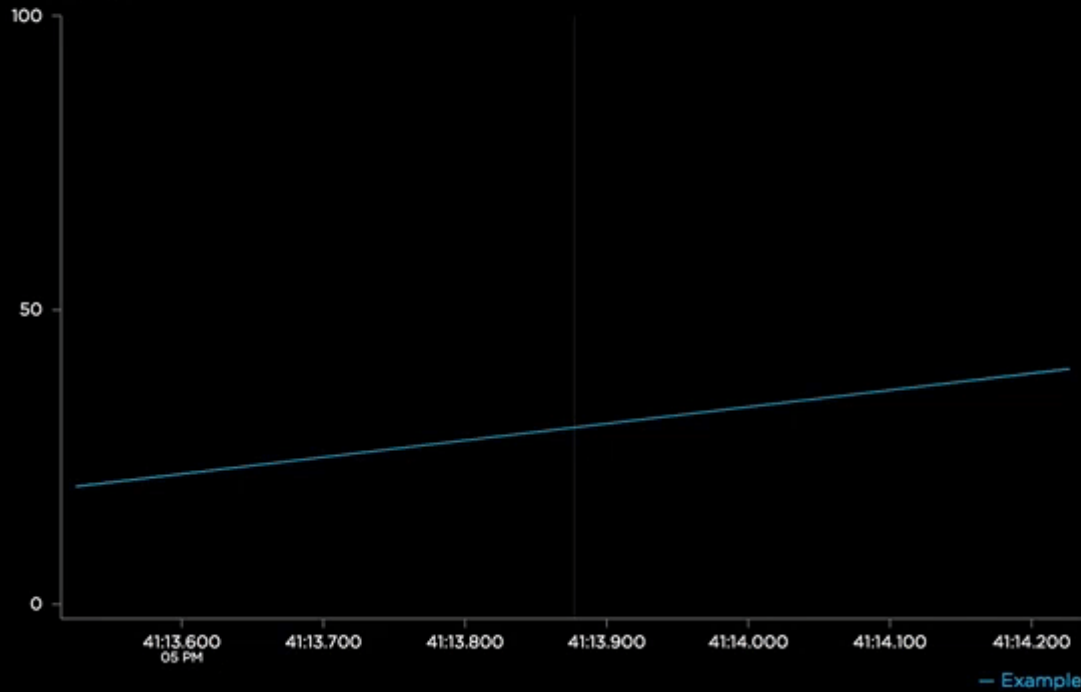
5 dashboards

Community support

Projects and guides



Example



Example



Example



February 13th 2017, 5:41:14 pm

Software

Descargar e instalar Arduino IDE

<https://www.arduino.cc/en/Main/Software>



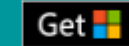
ARDUINO 1.8.9

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10



Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits

Linux 64 bits

Linux ARM 32 bits

Linux ARM 64 bits

[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)

Software

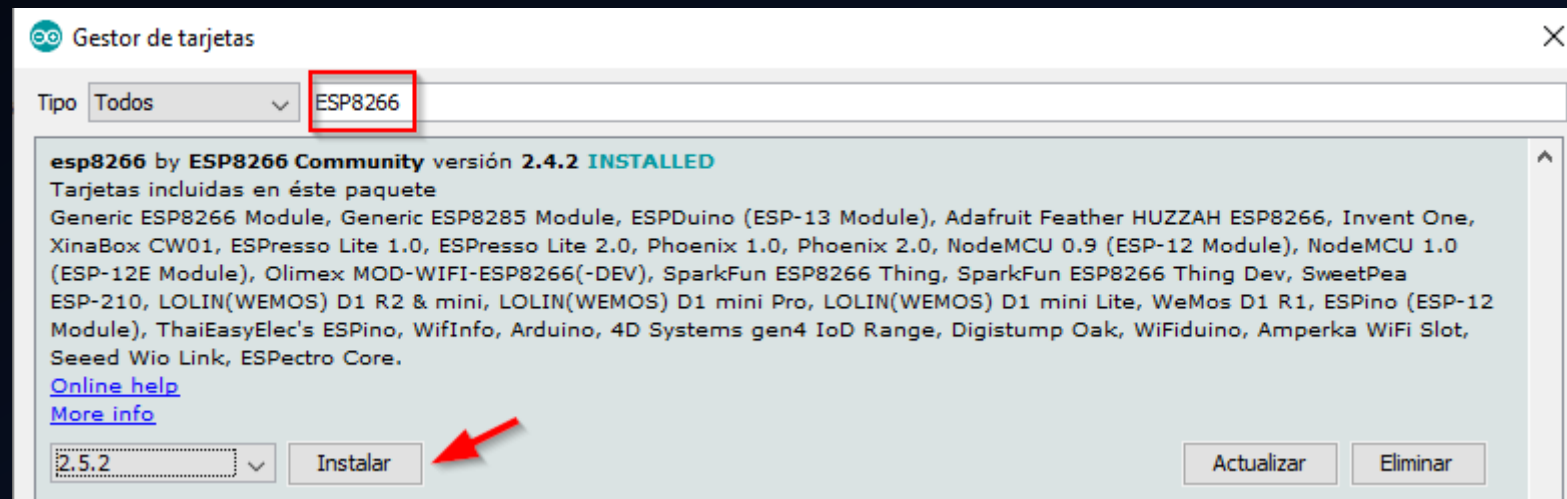
Añadir soporte para ESP8266 en el Arduino IDE

1. Archivo-> Preferencias-> Gestor de URLs adicionales de tarjetas
2. Copiar y pegar la siguiente URL -> http://arduino.esp8266.com/stable/package_esp8266com_index.json

Gestor de URLs Adicionales de Tarjetas:



3. Herramientas-> Placa-> Gestor de tarjetas...
4. Buscar por ESP8266 e instalar '**esp8266 by ESP8266 Community**'



Software

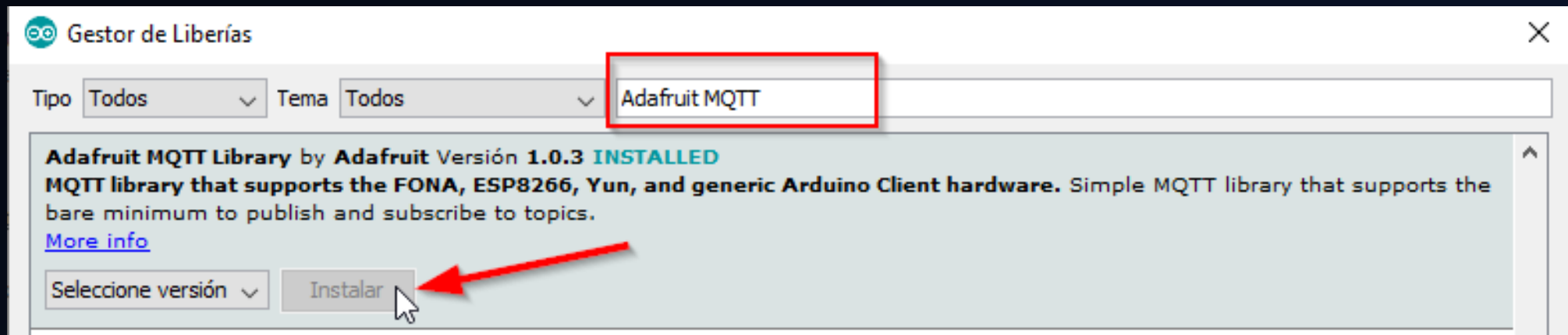
Añadir librerías al IDE de Arduino

1. Programa-> Incluir Librerías-> Administrar Bibliotecas...
2. Buscar las siguientes **librerías** e **Instalarlas**:

[ArduinoHttpClient](#)

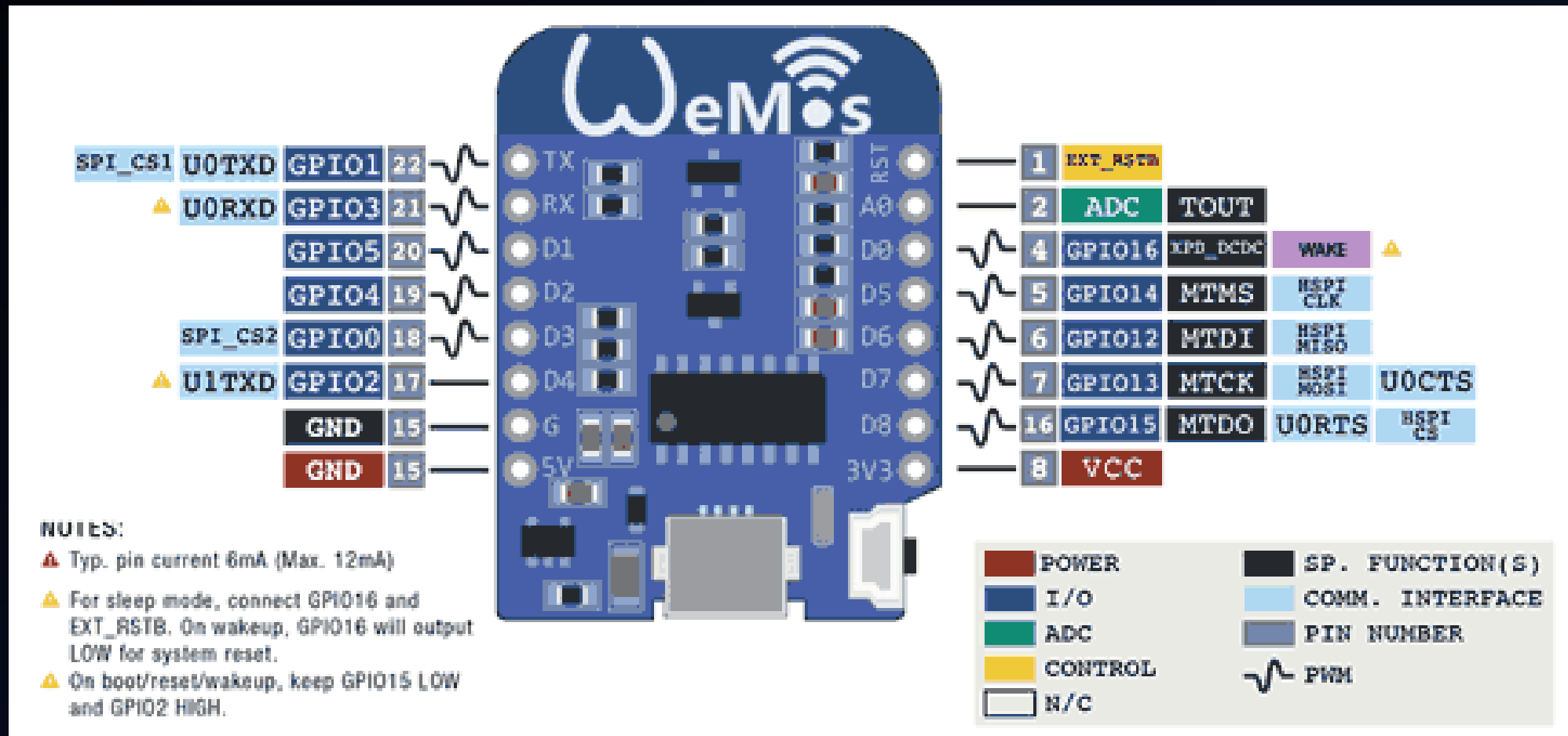
[Adafruit IO Arduino](#)

[Adafruit MQTT](#)



3. Crear cuentas de usuario en [AdafruitIO](#) y en [IFTTT](#)

Hardware: Wemos D1 Mini

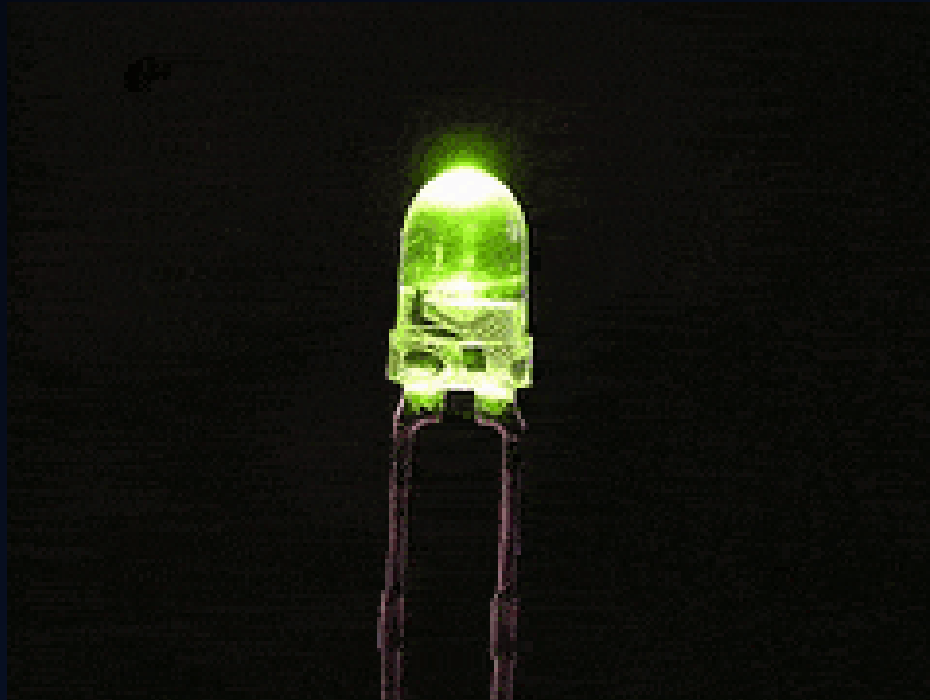


Hardware

Probar que todo ha ido bien... (i)

Cargar nuestro primer... 'Hello World' 🙌🌐

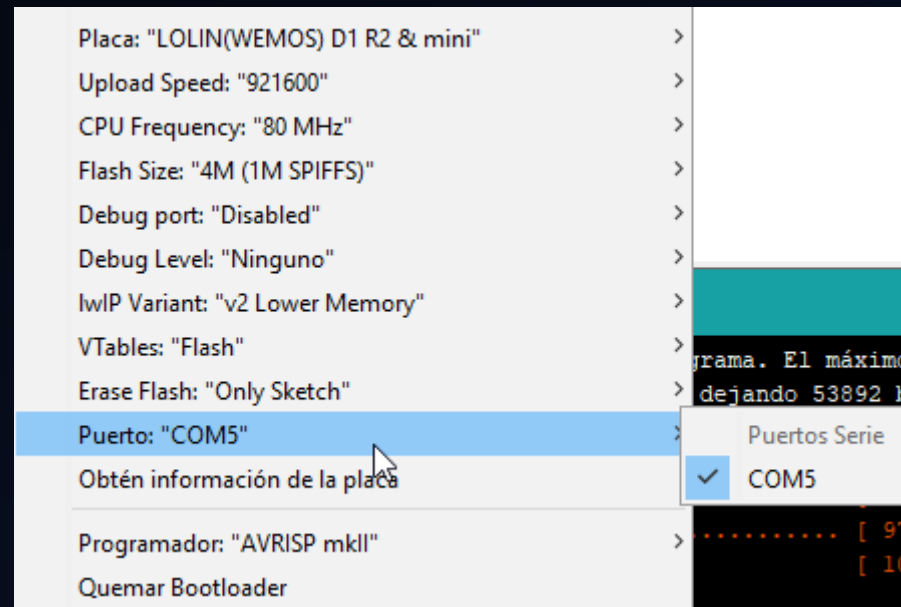
Archivo-> Ejemplos-> 01.Basics-> Blink



Hardware

Probar que todo ha ido bien... (ii)



1. Conectar la placa con el cable USB y seleccionar la placa **LOLIN(WEMOS) D1 R2 & mini**



2. Seleccionar el **Puerto Serie** que aparezca donde se ha detectado la placa (COMX) Siendo X: número
3. Verificar ✓ y Subir → el código a la placa para que el código se ejecute

Hardware

Conectando la placa a Internet

1. Archivo-> Ejemplos-> ESP8266WiFi-> WiFiClientBasic
2. Editar valores “SSID” y “passpasspass” por los de la red WiFi a conectar. Editar variable **host** por una URL
3. Verificar ✓ y Subir  el código a la placa para que el código se ejecute. Abrir Monitor Serie 

```
WiFiClientBasic $  
  
#include <ESP8266WiFi.h>  
#include <ESP8266WiFiMulti.h>  
  
ESP8266WiFiMulti WiFiMulti;  
  
void setup() {  
  Serial.begin(115200);  
  delay(10);  
  
  // We start by connecting to a WiFi network  
  WiFi.mode(WIFI_STA);  
  WiFiMulti.addAP("SSID", "passpasspass");
```

```
COM5  
  
WiFi connected  
IP address:  
192.168.1.39  
connecting to www.google.es  
  
closing connection
```

PLUG 'EM ALL



WORKSHOP

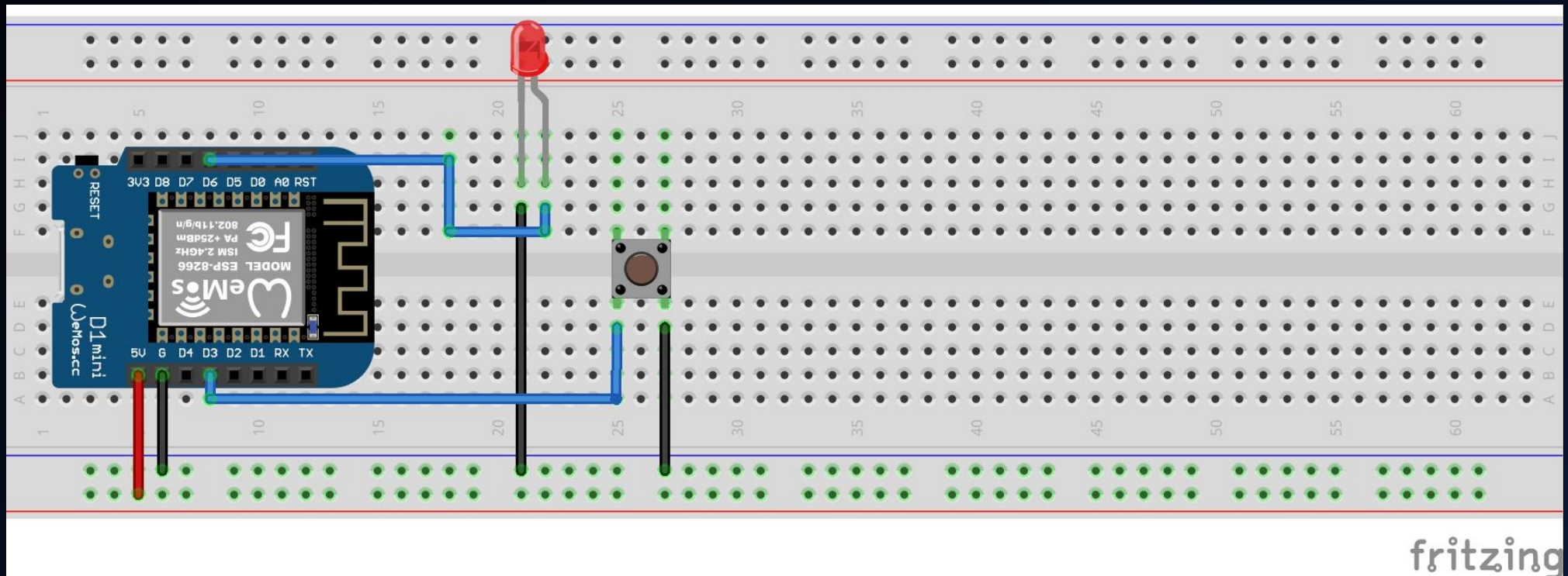
PRÁCTICAS



Práctica 01 – Entrada Digital

Montaje Botón + Feed (i)

1. Realizar montaje del circuito según el siguiente esquema



Práctica 01 – Entrada Digital

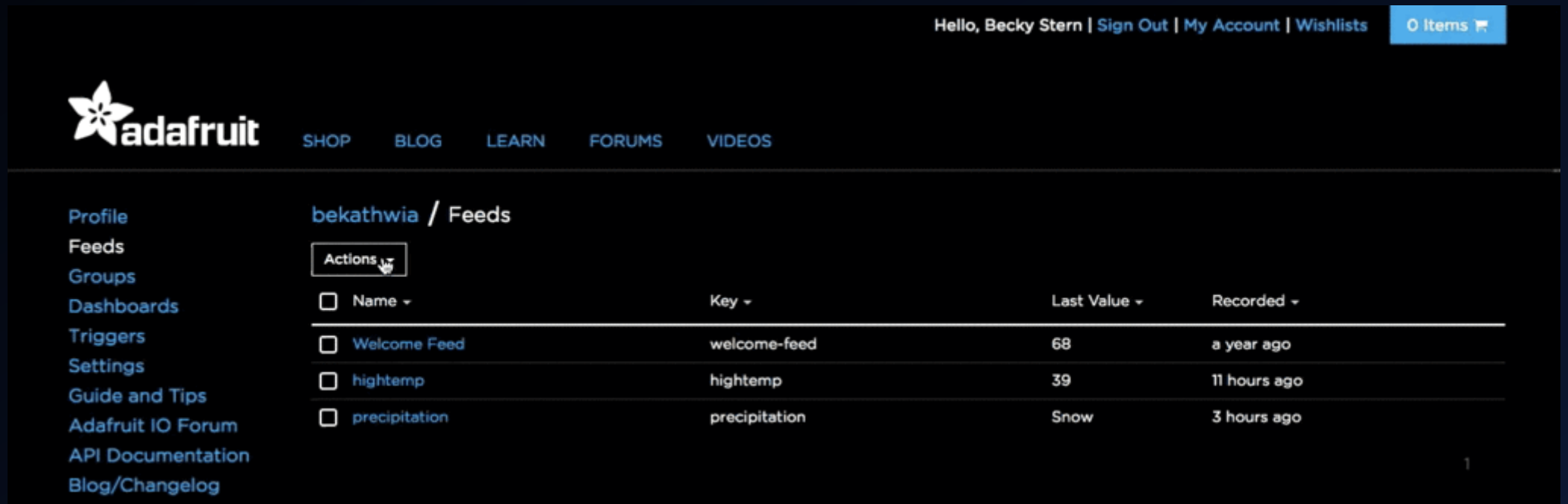
Montaje Botón + Feed (ii)

2. Descargar el código Practica_01 y abrirlo con el Arduino IDE
3. Actualizar la configuración para que coincidan tus credenciales
 - Adafruit IO **Username** y **Active Key**
 - WiFi **SSID** y **Password**
4. Verificar ✓ y Subir → el código a la placa para que el código se ejecute
5. Comprobar el correcto funcionamiento del circuito
6. Si no funciona, revisar la siguiente lista de errores comunes:
 - Conexionado incorrecto o ausente de alguna de las conexiones
 - LED conectado al revés
 - Código no cargado correctamente
 - Tipo de placa incorrecto
 - Puerto COM no seleccionado o incorrecto
 - Velocidad Monitor Serie

Práctica 01 – Entrada Digital

Montaje Botón + Feed (iii)

7. Crear un nuevo Feed en AdafruitIO y llamarlo 'Digital'



The screenshot shows the AdafruitIO website interface. At the top, the user is logged in as 'Becky Stern' with links for 'Sign Out', 'My Account', and 'Wishlists'. A shopping cart icon shows '0 Items'. The main navigation bar includes the Adafruit logo and links for 'SHOP', 'BLOG', 'LEARN', 'FORUMS', and 'VIDEOS'. On the left, a sidebar menu lists various options: Profile, Feeds, Groups, Dashboards, Triggers, Settings, Guide and Tips, Adafruit IO Forum, API Documentation, and Blog/Changelog. The main content area is titled 'bekathwia / Feeds' and features an 'Actions' button. Below this is a table listing the user's feeds.

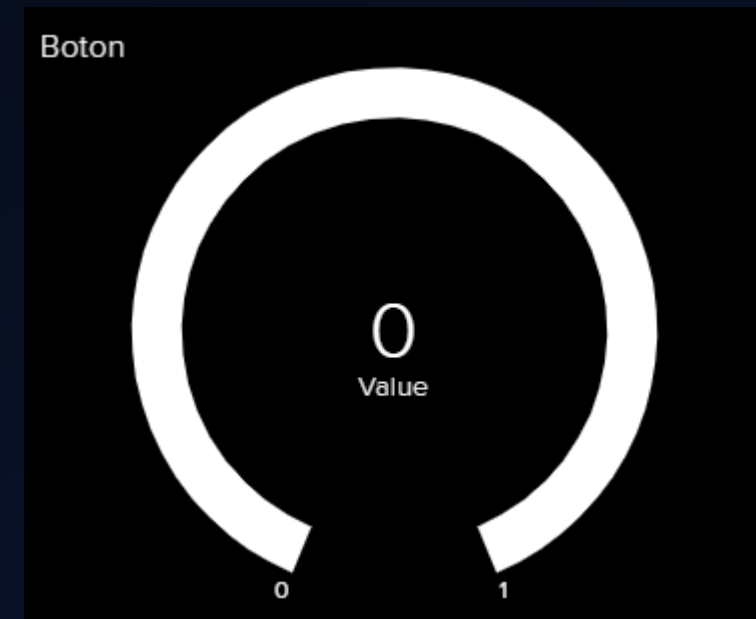
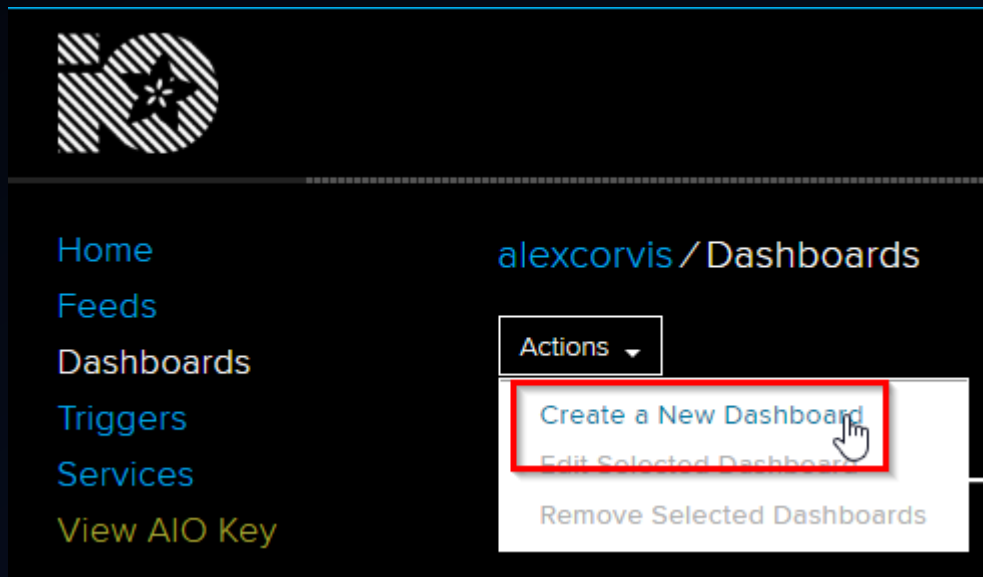
<input type="checkbox"/> Name ▾	Key ▾	Last Value ▾	Recorded ▾
<input type="checkbox"/> Welcome Feed	welcome-feed	68	a year ago
<input type="checkbox"/> hightemp	hightemp	39	11 hours ago
<input type="checkbox"/> precipitation	precipitation	Snow	3 hours ago

1

Práctica 01 – Entrada Digital

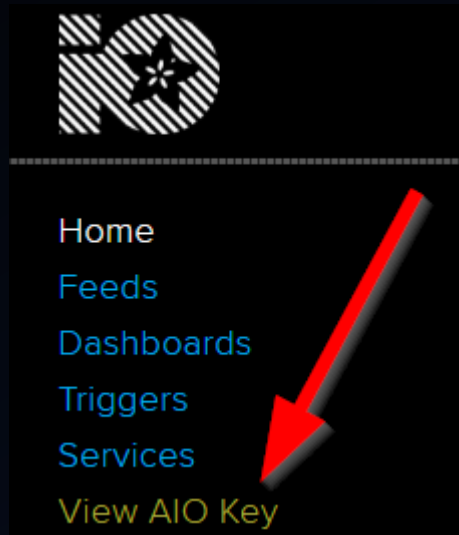
Montaje Botón + Feed (iv)

8. Crear un **Dashboard** en **AdafruitIO** y llamarlo '**Medialab**'
9. Añadir un bloque tipo '**Gauge**' asociado al Feed '**Digital**' con los valores 0 y 1 como mínimo y máximo.
10. Comprobar el **correcto** funcionamiento del circuito. Presionar botón y ver la señal recibida en el **Gauge**.



Práctica 01 – Entrada Digital

Anexo I



YOUR AIO KEY

Your Adafruit IO key should be kept in a safe place and treated with the same care as your Adafruit username and password. People who have access to your AIO key can view all of your data, create new feeds for your account, and manipulate your active feeds.

If you need to regenerate a new AIO key, all of your existing programs and scripts will need to be manually changed to the new key.



Username

Active Key

REGENERATE AIO KEY

[Hide Code Samples](#)

Arduino

```
#define IO_USERNAME "alexcorvis"
#define IO_KEY "ffef62b3e5e44"
```


```
// visit io.adafruit.com if you need to create an account,
// or if you need your Adafruit IO key.
#define IO_USERNAME "alexcorvis"
#define IO_KEY "ffef62b3e5e44"

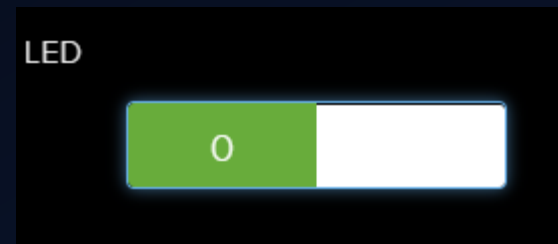
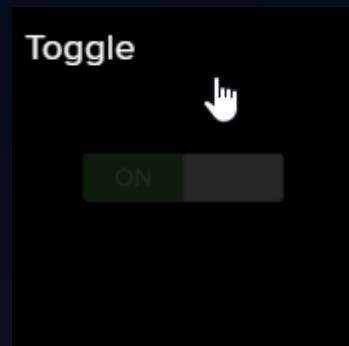
/***** WIFI Configuration *****/

#define WIFI_SSID "adafruit"
#define WIFI_PASS "adafruit"
```

Práctica 02 – Salida Digital

Montaje LED (i)

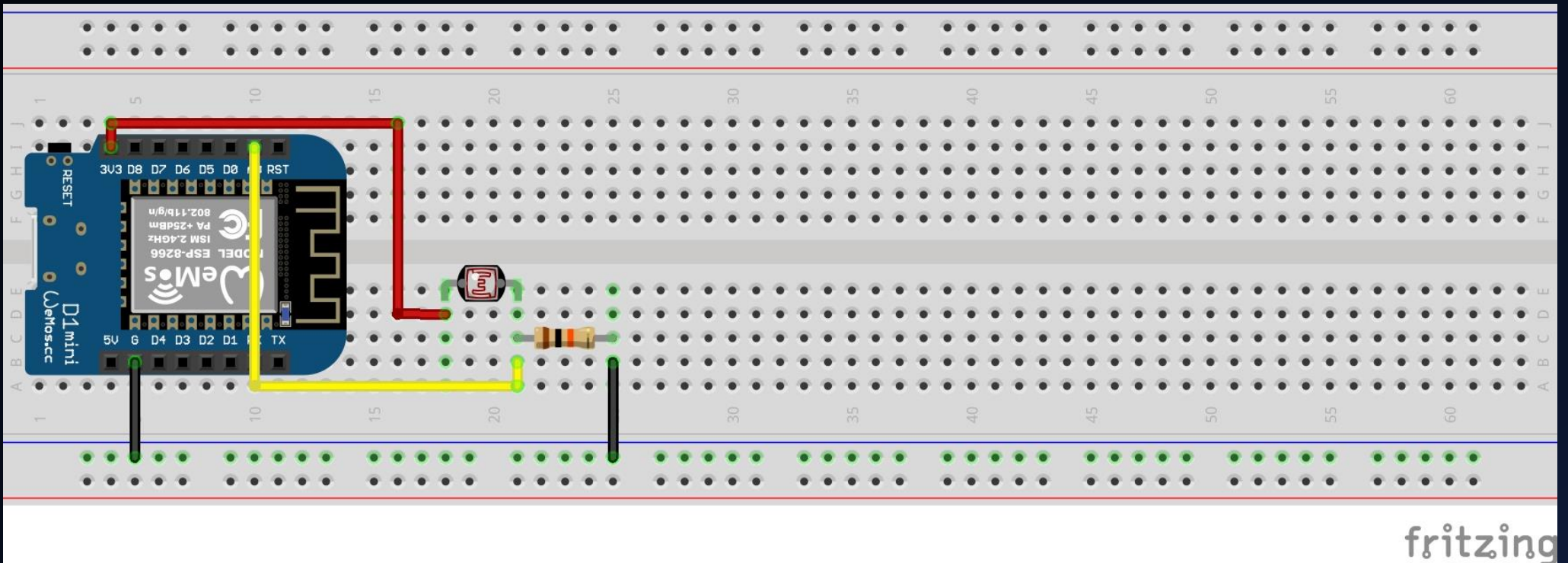
1. Crear un nuevo **Feed** llamado '**Led**'
2. Añadir un bloque tipo '**Toggle**' asociado al Feed '**Led**'
3. Editar los campos **Button On Text** y **Button Off Text** con los valores **1** y **0** respectivamente
4. Descargar el código **Practica_02** y abrirlo con el Arduino IDE
5. Actualizar la configuración para que coincidan tus credenciales
 - Adafruit IO **Username** y **Active Key**
 - WiFi **SSID** y **Password**
6. Verificar ✓ y Subir  el código a la placa para que el código se ejecute
7. Comprobar el correcto funcionamiento del circuito



Práctica 03 – Entrada Analógica


Montaje LDR (Fotoresistencia) (i)

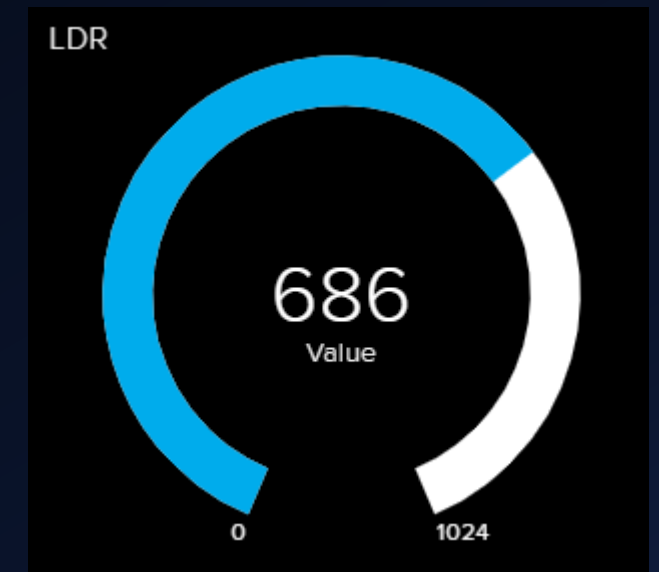
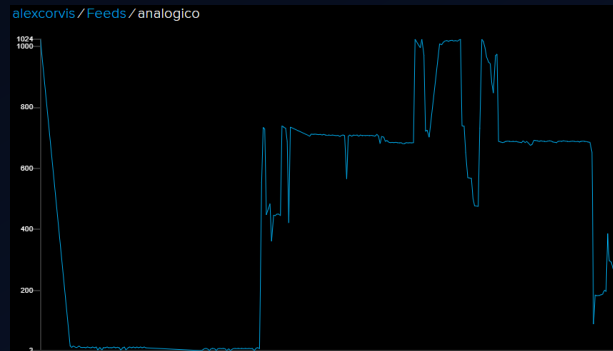
1. Realizar montaje del circuito según el siguiente esquema



Práctica 03 – Entrada Analógica

Montaje LDR (Fotoresistencia) (ii)

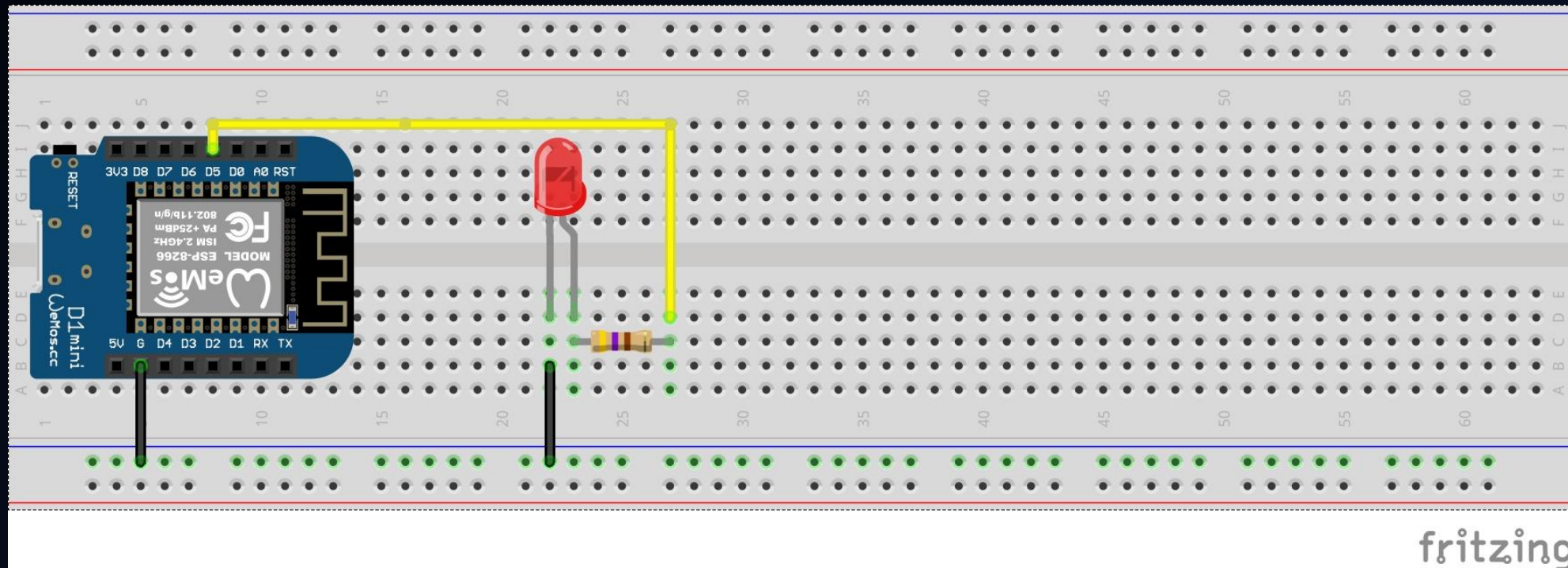
2. Crear un nuevo **Feed** llamado '**analogico**'
3. Añadir un bloque tipo '**Gauge**' asociado al Feed '**analogico**'
4. Editar los campos **Gauge Min Value** y **Gauge Max Value** con los valores **0** y **1024** respectivamente
5. Descargar el código **Practica_03** y abrirlo con el Arduino IDE
6. Actualizar la configuración para que coincidan tus credenciales
 - Adafruit IO **Username** y **Active Key**
 - WiFi **SSID** y **Password**
7. Verificar ✓ y Subir  el código a la placa para que el código se ejecute
8. Comprobar el correcto funcionamiento del circuito



Práctica 04 – Salida Analógica


Montaje LED (i)

1. Realizar montaje del circuito según el siguiente esquema



Práctica 04 – Salida Analógica

Montaje LED (ii)

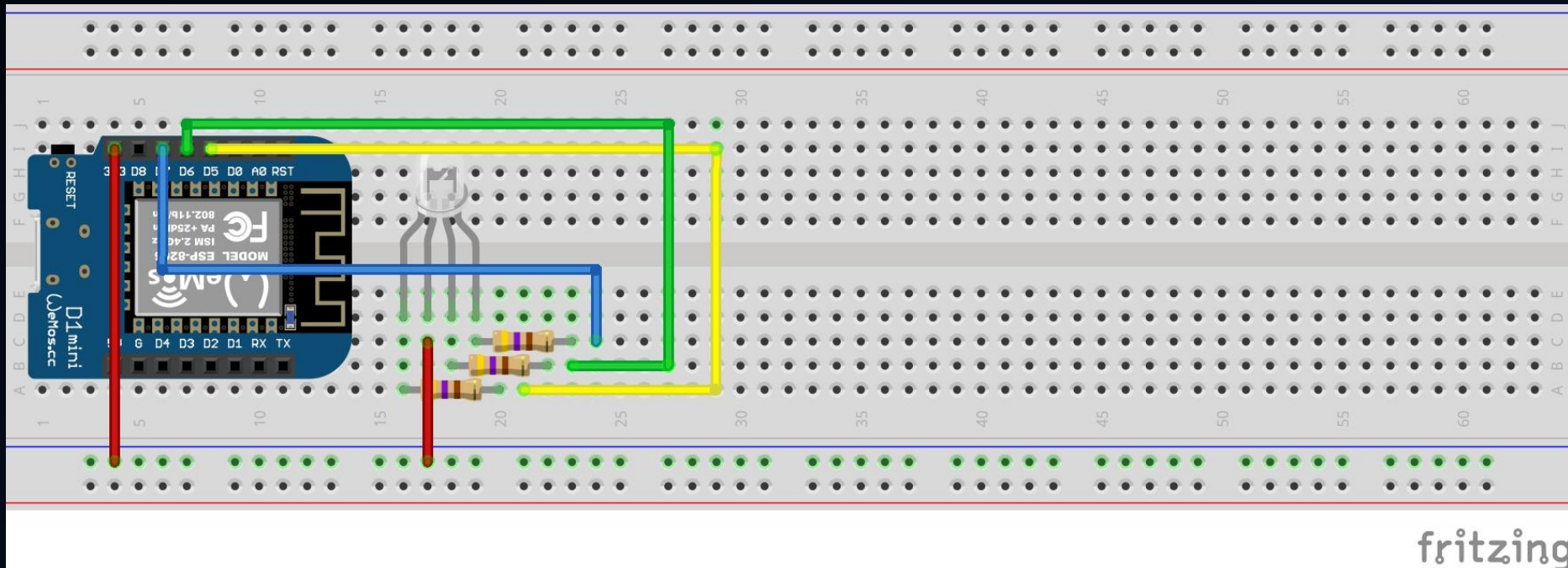
2. Crear un nuevo **Feed** llamado '**intensidad**'
3. Añadir un bloque tipo '**Slider**' asociado al Feed '**intensidad**'
4. Editar los campos **Slider Min Value** y **Slider Max Value** con los valores **0** y **1024** respectivamente
5. Descargar el código **Practica_04** y abrirlo con el Arduino IDE
6. Actualizar la configuración para que coincidan tus credenciales
 - Adafruit IO **Username** y **Active Key**
 - WiFi **SSID** y **Password**
7. Verificar ✓ y Subir  el código a la placa para que el código se ejecute
8. Comprobar el correcto funcionamiento del circuito



Práctica 05 – Color


Montaje LED RGB (i)

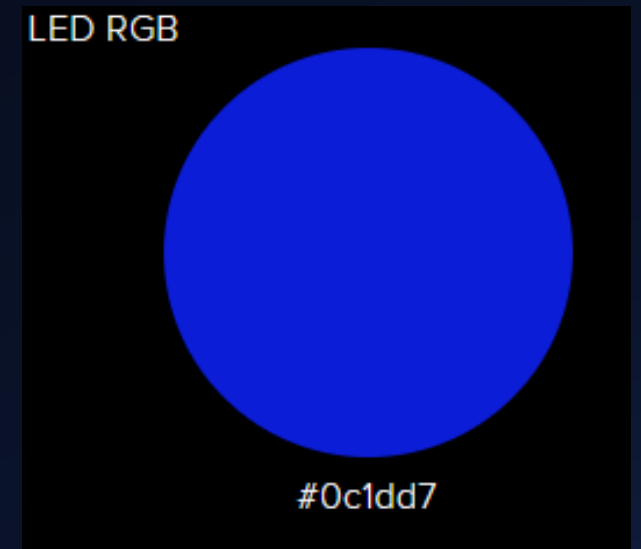
1. Realizar montaje del circuito según el siguiente esquema



Práctica 05 – Color

Montaje LED RGB (ii)

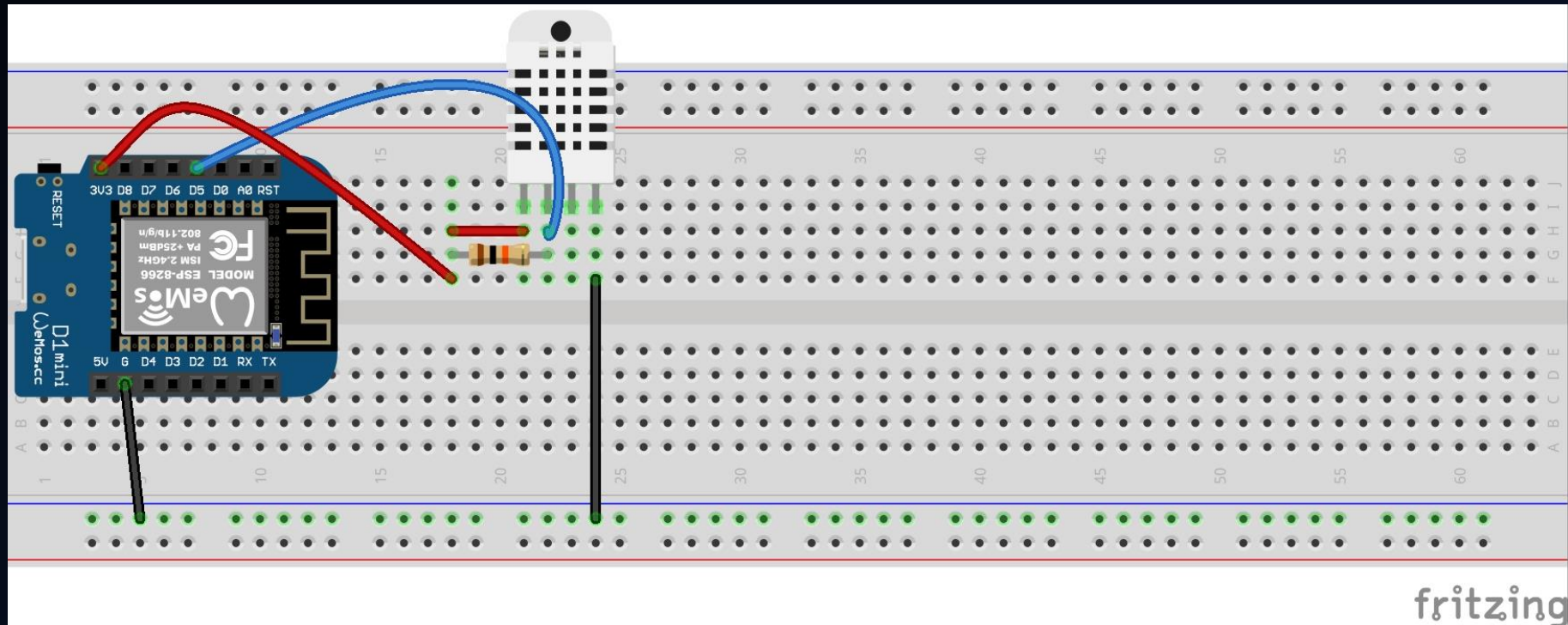
2. Crear un nuevo **Feed** llamado '**color**'
3. Añadir un bloque tipo '**Color Picker**' asociado al Feed '**color**'
4. Descargar el código **Practica_05** y abrirlo con el Arduino IDE
5. Actualizar la configuración para que coincidan tus credenciales
 - Adafruit IO **Username** y **Active Key**
 - WiFi **SSID** y **Password**
6. Verificar ✓ y Subir  el código a la placa para que el código se ejecute
7. Comprobar el correcto funcionamiento del circuito



Práctica 06 – Temperatura y Humedad

Montaje Sensor DHT22 (i)

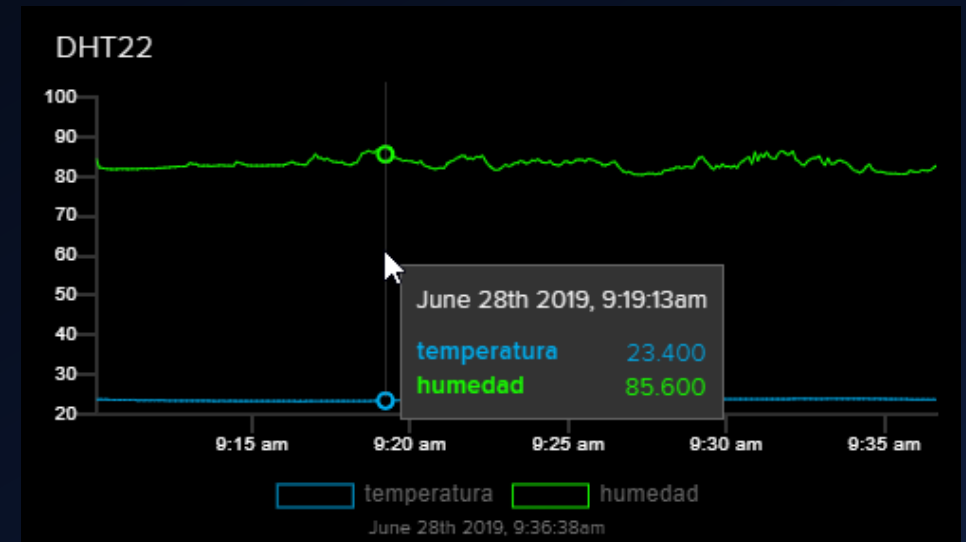
1. Realizar montaje del circuito según el siguiente esquema



Práctica 06 – Temperatura y Humedad

Montaje Sensor DHT22 (ii)

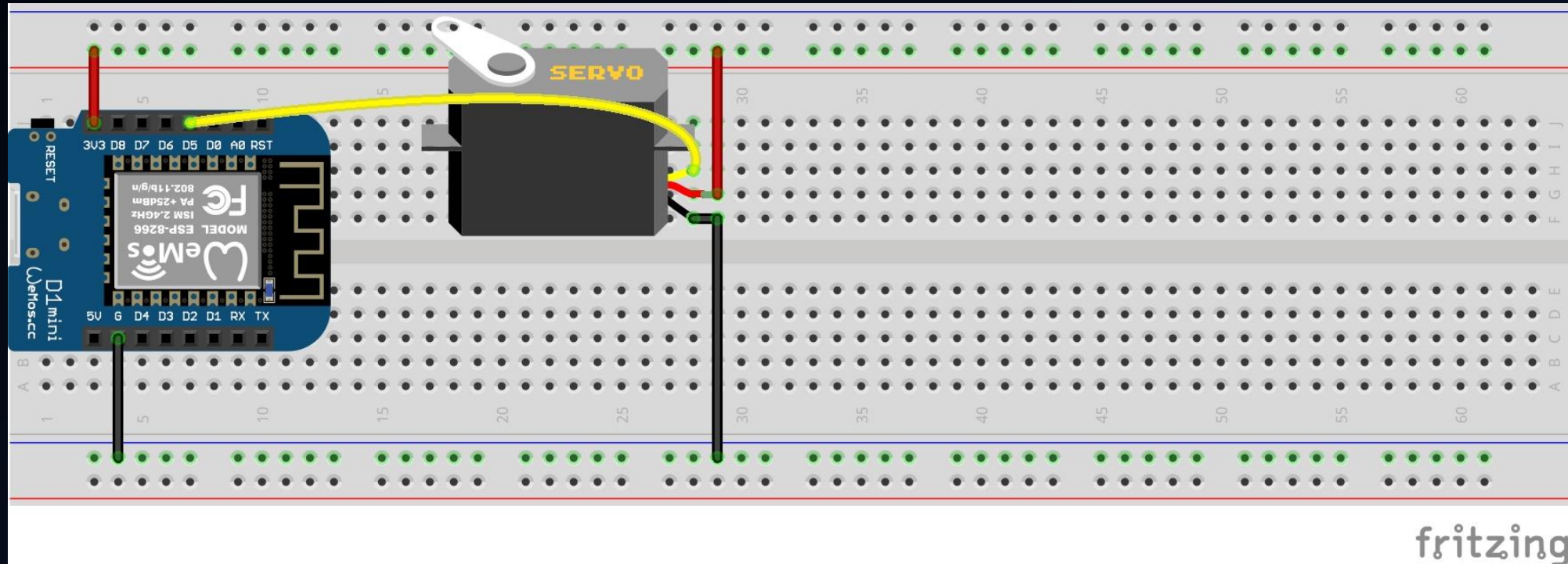
2. Instalar librerías 'Adafruit Unified Sensor' y 'DHT Sensor Library'
3. Crear dos nuevos Feeds llamados 'temperatura' y 'humedad'
4. Añadir un bloque tipo 'Line Chart' asociado al Feeds 'temperatura' y 'humedad'
5. Asignar los valores Y-Axis Minimum y Y-Axis Maximum como 0 y 100 respectivamente.
6. Descargar el código Practica_06 y abrirlo con el Arduino IDE
7. Actualizar la configuración para que coincidan tus credenciales
 - Adafruit IO Username y Active Key
 - WiFi SSID y Password
8. Verificar ✓ y Subir → el código a la placa
9. Comprobar el correcto funcionamiento del circuito



Práctica 07 – Servo

Montaje Servo (i)

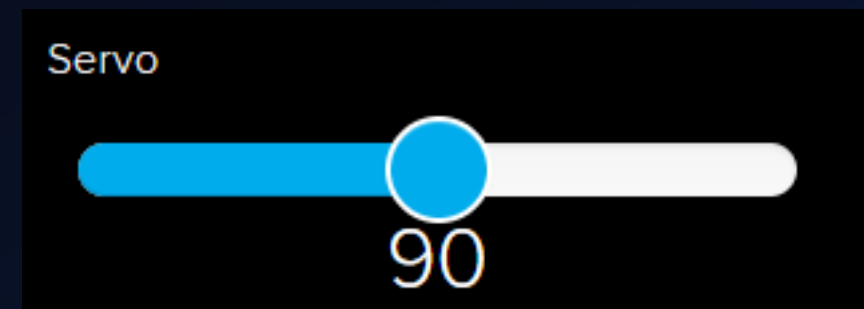
1. Realizar montaje del circuito según el siguiente esquema



Práctica 07 – Servo

Montaje Servo (ii)

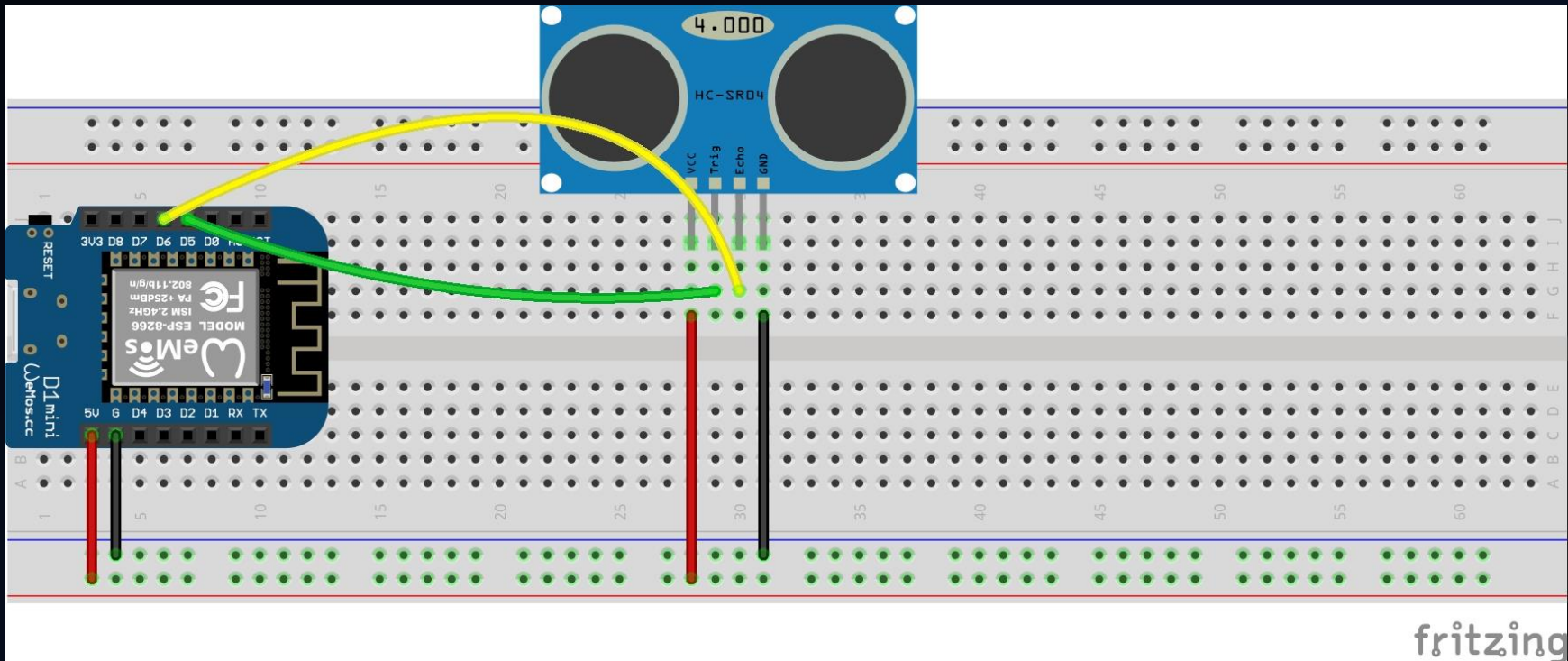
2. Crear un **Feed** llamado '**Servo**'
3. Añadir un bloque tipo '**Slider Block**' asociado al Feed '**Servo**'
4. Asignar los valores **Slider Min value** y **Slider Max Value** como **0** y **180** respectivamente.
5. Descargar el código **Practica_07** y abrirlo con el Arduino IDE
6. Actualizar la configuración para que coincidan tus credenciales
 - Adafruit IO **Username** y **Active Key**
 - WiFi **SSID** y **Password**
7. Verificar ✓ y Subir → el código a la placa
8. Comprobar el correcto funcionamiento del circuito



Práctica 08 – Distancia con ultrasonidos

Montaje Sensor HC-SR04 (i)

1. Realizar montaje del circuito según el siguiente esquema



Práctica 08 – Distancia con ultrasonidos

Montaje Sensor HC-SR04 (ii)

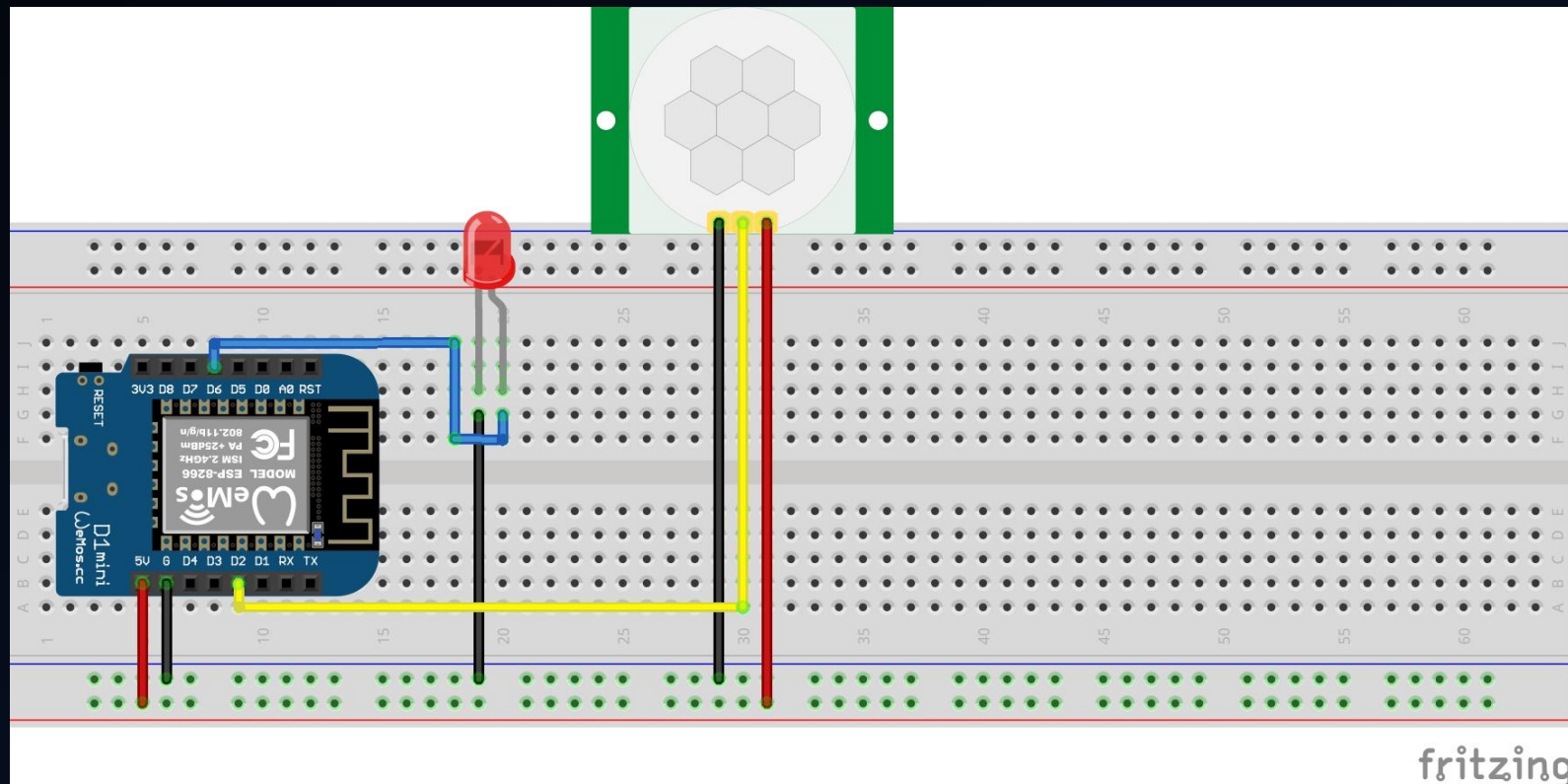
2. Crear un **Feed** llamado '**Distancia**'
3. Añadir un bloque tipo '**Gauge**' asociado al Feed '**Distancia**'
4. Asignar los valores **Gauge Min Value** y **Gauge Max Value** como **0** y **200** respectivamente.
5. Descargar el código **Practica_08** y abrirlo con el Arduino IDE
6. Actualizar la configuración para que coincidan tus credenciales
 - Adafruit IO **Username** y **Active Key**
 - WiFi **SSID** y **Password**
7. Verificar ✓ y Subir → el código a la placa
8. Comprobar el correcto funcionamiento del circuito



Práctica 09 – Sensor PIR


Montaje Sensor PIR HC-SR501 (i)

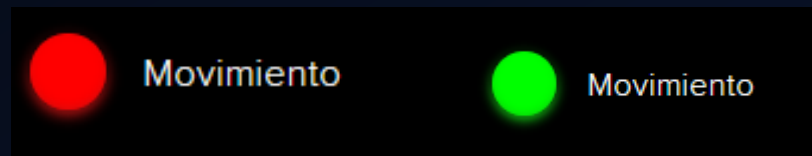
1. Realizar montaje del circuito según el siguiente esquema



Práctica 09 – Sensor PIR

Montaje Sensor PIR HC-SR501 (ii)

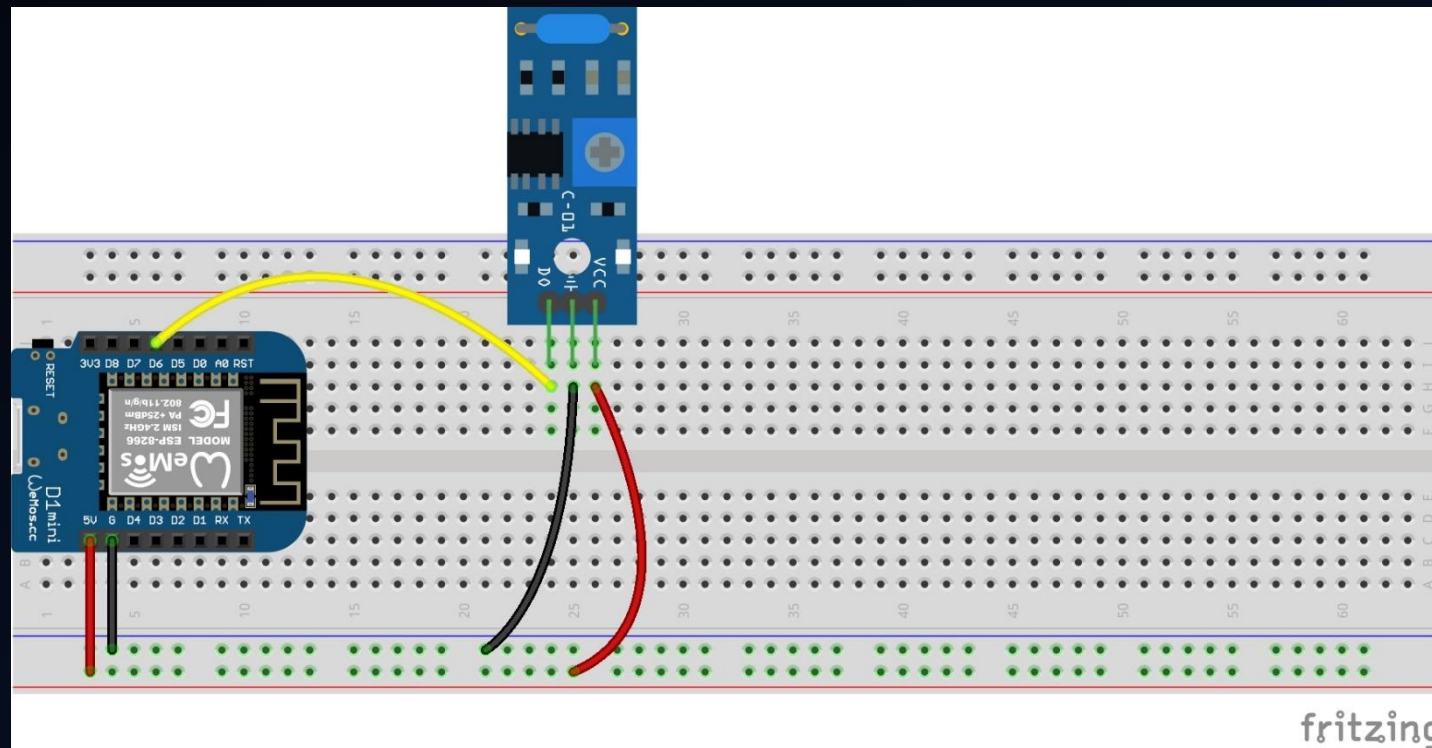
2. Crear un **Feed** llamado '**Pir**'
3. Añadir un bloque tipo '**Indicator**' asociado al Feed '**Pir**'
4. Asignar los valores **On Color** **#00ff00**, **Off Color** **#ff0000** y **Conditions** = **1**
5. Descargar el código **Practica_09** y abrirlo con el Arduino IDE
6. Actualizar la configuración para que coincidan tus credenciales
 - Adafruit IO **Username** y **Active Key**
 - WiFi **SSID** y **Password**
7. Verificar ✓ y Subir  el código a la placa
8. Comprobar el correcto funcionamiento del circuito



Práctica 10 – Sensor vibración

Montaje Sensor SW-420 (i)

1. Realizar montaje del circuito según el siguiente esquema



Práctica 10 – Sensor vibración

Montaje Sensor SW-420 (i)

2. Crear un **Feed** llamado 'Tilt'
3. Añadir un bloque tipo '**Indicator**' asociado al Feed 'Tilt'
4. Asignar los valores **On Color** #00ff00, **Off Color** #ff0000 y **Conditions** = 1
5. Descargar el código **Practica_10** y abrirlo con el Arduino IDE
6. Actualizar la configuración para que coincidan tus credenciales
 - Adafruit IO **Username** y **Active Key**
 - WiFi **SSID** y **Password**
7. Verificar ✓ y Subir → el código a la placa
8. Comprobar el correcto funcionamiento del circuito



Práctica 11 - Triggers

¿Qué son los Triggers?

- Son una característica de Adafruit IO que nos permite **añadir lógica** entre la colección de datos
- Un **Trigger** es una respuesta a un evento que es recibido, por ejemplo, un nuevo dato es publicado en un Feed
- Cuando un dato es añadido a un Feed, cualquier Trigger asociado compara el nuevo valor sobre una constante o el valor actual de otro feed
- Si la condición se satisface, se pueden tomar varias decisiones:
 - Enviar un correo electrónico
 - Enviar un mensaje Webhook
 - Enviar un valor específico a un Feed

Práctica 11 - Triggers

Edit Trigger ✕

If

tilt

Is

equal to

Comparison Value or F...

0

Then

publish a message to:

Feed

Servo

With

0

Cancel

Save

Edit Trigger ✕

If

tilt

Is

equal to

Comparison Value or F...

1

Then

publish a message to:

Feed

Servo

With

180

Cancel

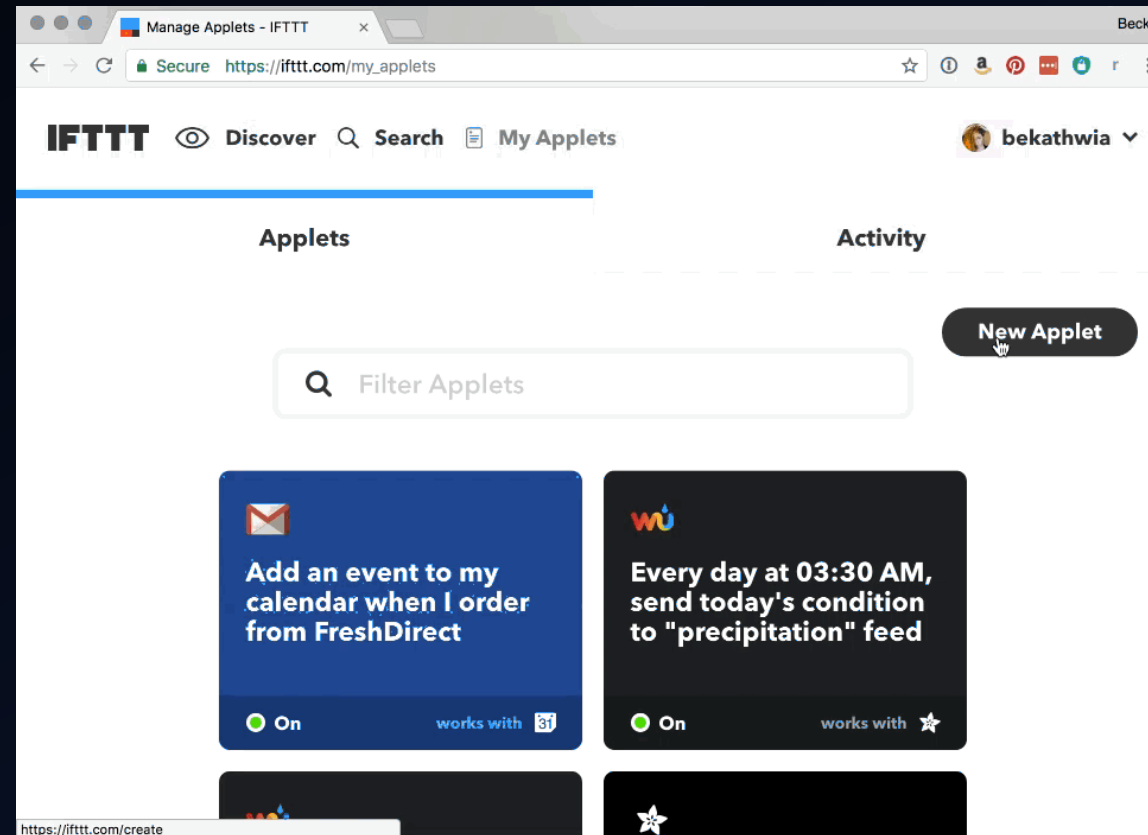
Save

<input type="checkbox"/>	ID ▾	Created ▾	Last Run ▾	Description	Action ▾	Run Count ▾	
<input type="checkbox"/>	15...	about 23 hours	2019/07/03 7:20pm	If tilt is equal to "0" then set Servo to "0".	Feed	32	✎
<input type="checkbox"/>	15...	about 23 hours	2019/07/03 7:20pm	If tilt is equal to "1" then set Servo to "180".	Feed	31	✎

Práctica 12 - IFTTT

Añadir servicio online IFTTT para toma de decisiones (i)

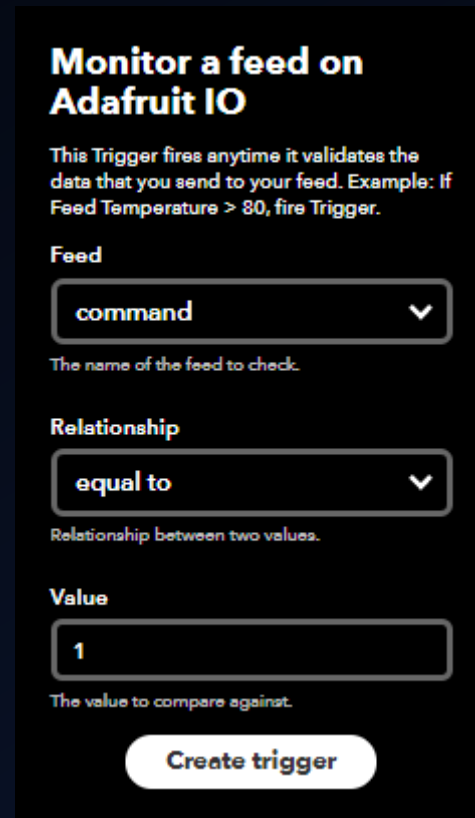
1. Conectar el Feed a IFTTT creando un nuevo Applet



Práctica 12 - IFTTT

Añadir servicio online IFTTT para toma de decisiones (ii)

2. Configurar el Applet seleccionando: Feed '**command**' previamente creado, relación y valor
3. Crear **trigger**



Monitor a feed on Adafruit IO

This Trigger fires anytime it validates the data that you send to your feed. Example: If Feed Temperature > 80, fire Trigger.

Feed

command

The name of the feed to check.

Relationship

equal to

Relationship between two values.

Value

1

The value to compare against.

Create trigger

Práctica 12 - IFTTT

Añadir servicio online IFTTT para toma de decisiones (iii)

4. Ahora elegir **+That** para elegir un servicio de salida



5. Elegir un servicio de la lista de disponible (E-mail, Twitter, Telegram, etc.)
6. Configurar el servicio acorde a las necesidades
7. Crear Acción



Complete action fields

Step 5 of 6

Send message

This action will send you a message from @IFTTT on Facebook Messenger.

Message text

FeedName FeedValue is
Operator TriggerValue !

Add ingredient


Create action



If Data on command feed is
equal to 1, then Send
message

56/140

by alexcorvis84

works with 



command 1 is equal to 1!



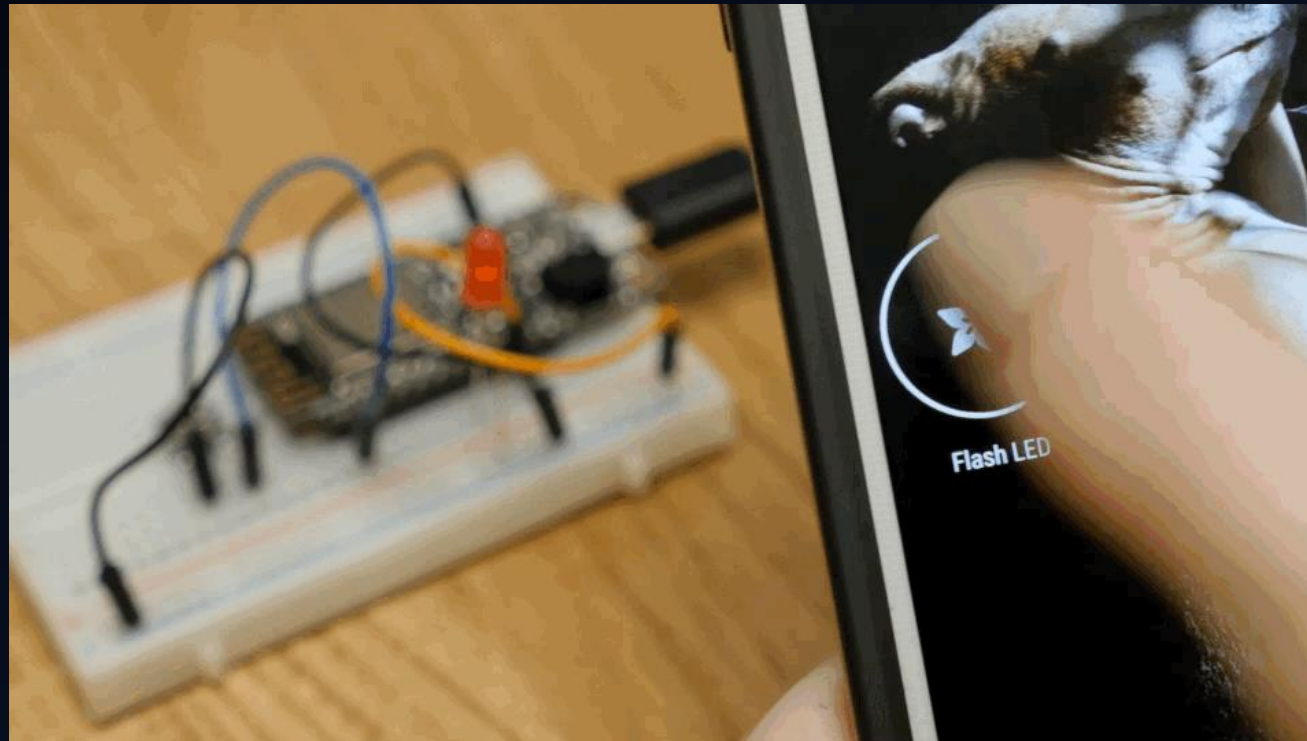
Escribe un mensaje...



Práctica 13 – IFTTT App Móvil

Mostrando datos recibidos de Internet

- Instalar aplicación IFTTT móvil
- Crear **Button widget (+this)** que envíe datos a un feed



Resumen Adafruit IO API

- Crear un Feed
 - `AdafruitIO_Feed *newFeed = io.feed("newfeed");`
- Añadir datos al Feed (Publicación) → `void loop()`
 - `io.run();`
 - `feedName → save(value);`
- Recibir datos de un Feed (Suscripción) → `void setup()`
 - `io.connect();`
 - `newFeed → onMessage(handleMessage);`
 - `void handleMessage(AdafruitIO_Data *data)`
- Conversión de datos

<code>data → toPinLevel()</code>	<code>data → toString()</code>	<code>data → toInt()</code>	<code>data → isTrue()</code>
<code>data → value()</code>	<code>data → toDouble()</code>	<code>data → toLong()</code>	<code>data → isFalse()</code>
<code>data → toChar()</code>	<code>data → toFloat()</code>	<code>data → toBool()</code>	...