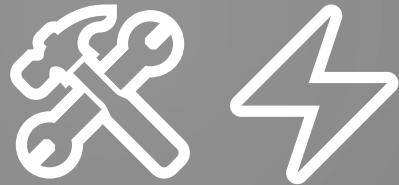


# PLUG 'EM ALL



Aprendiendo   
el Internet  de las cosas  
GUSTAVO REYNAGA

# DISCLAIMER

- ESTO NO ES MI TRABAJO 💰 ¡ES LO QUE ME APASIONA 😍!
- 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

## 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

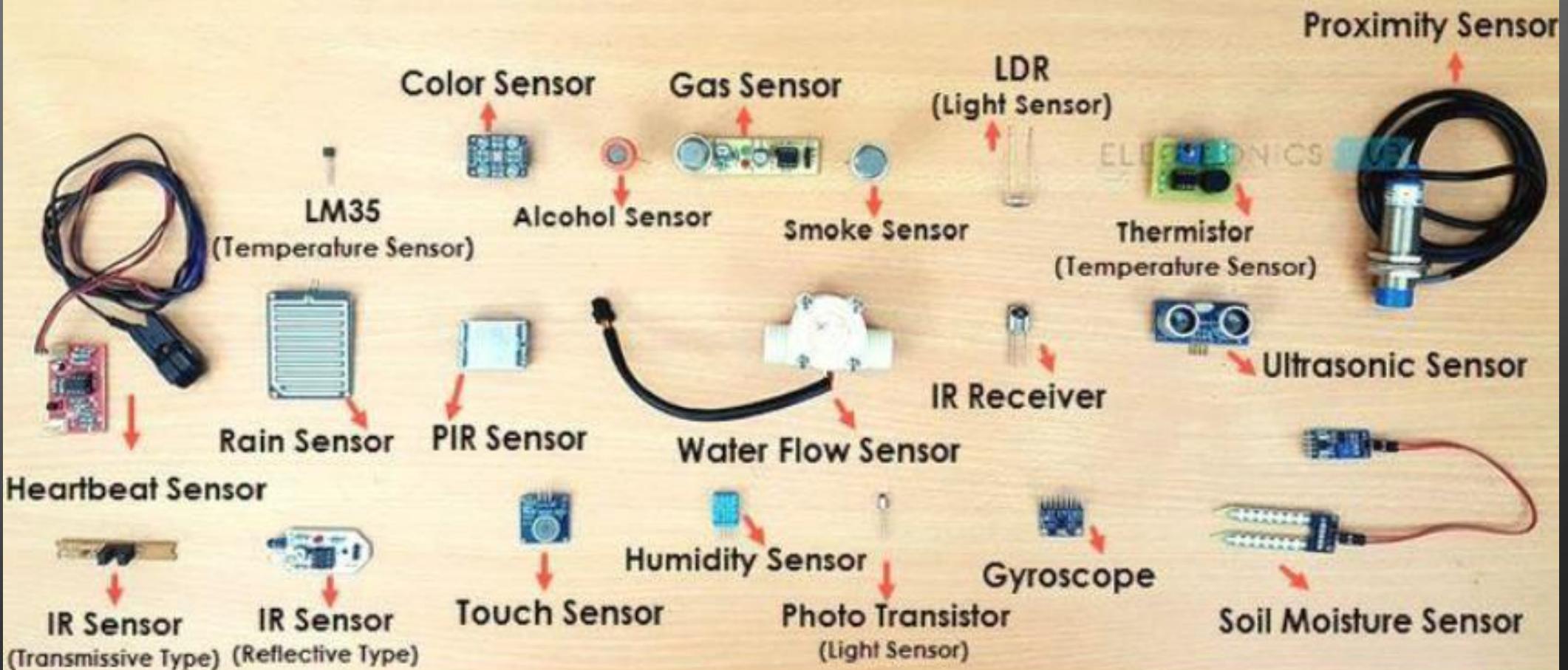


## Clasificación:

- Activos y pasivos (señal de excitación externa)
- Medios de detección
  - Eléctricos, químicos, biológicos, radioactivos...
- Fenómenos de conversión (E/S)
  - Fotoeléctricos, termoeléctricos, 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

CoAP

DDS

HTTP (REST/JSON)

MQTT

ZigBee

Z-Wave

Bluetooth

Wi-Fi

LoRaWAN

Sigfox

6LowPAN

NB-IoT

Broadcom

Digimesh

NFC

RFID

LoRa

...

# MQTT (MESSAGE QUEUE TELEMETRY TRANSPORT)

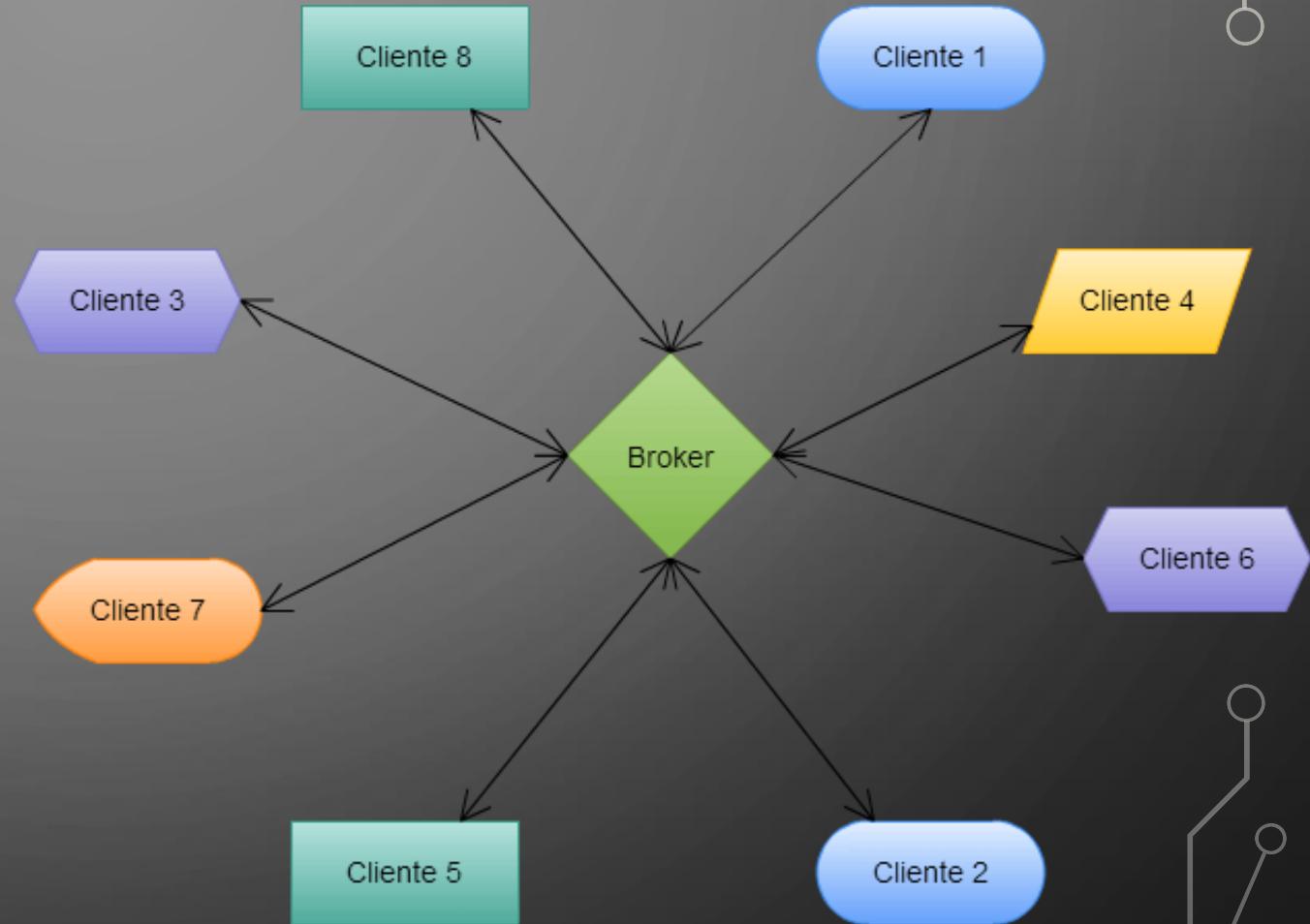
Protocolo de comunicación M2M ideal para el IOT

- ✓ 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:

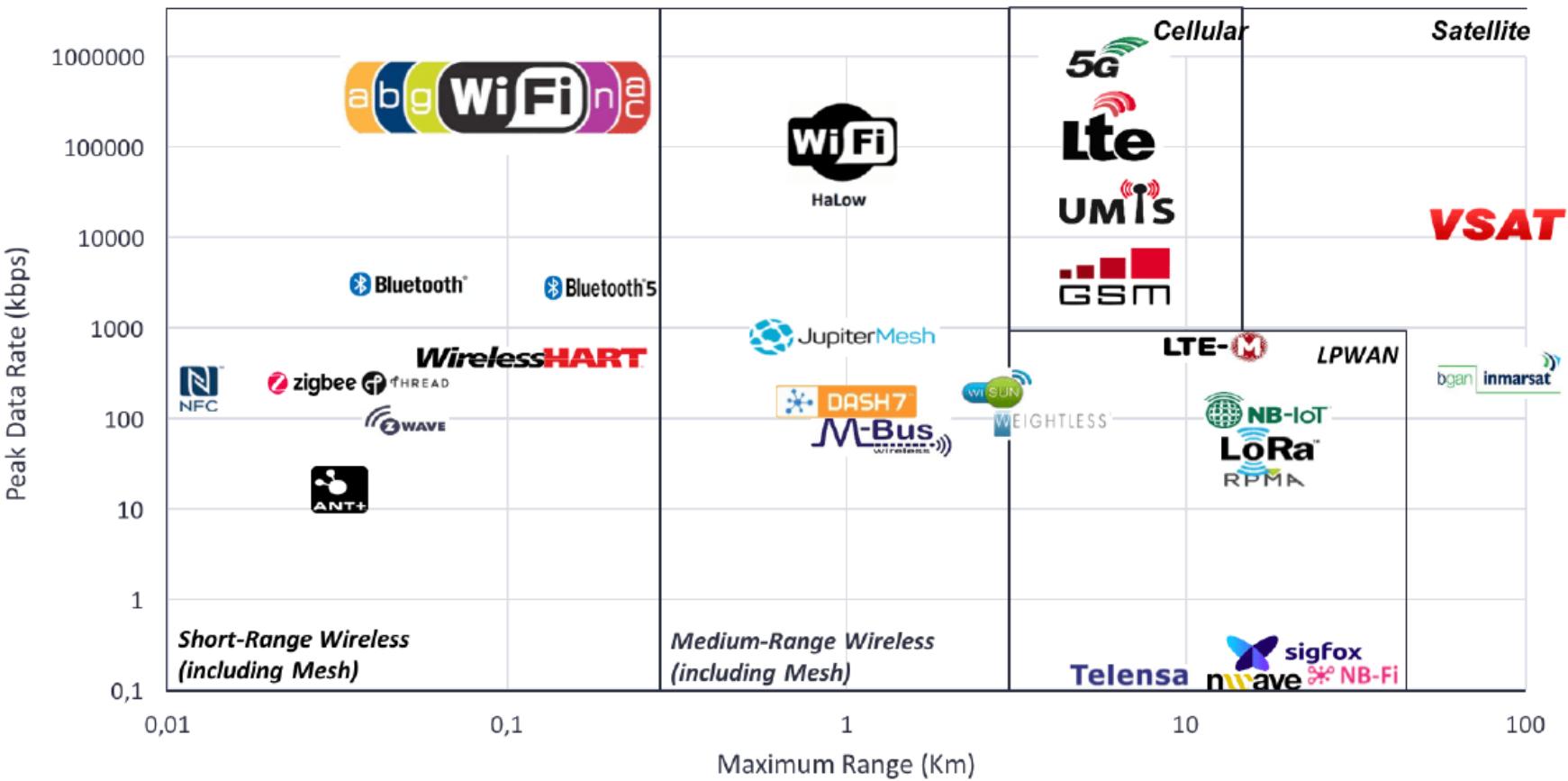
✓ Consumo de energía 

✓ Rango de operación (distancia) 

✓ Ancho de banda 

# Comparison Wireless technologies

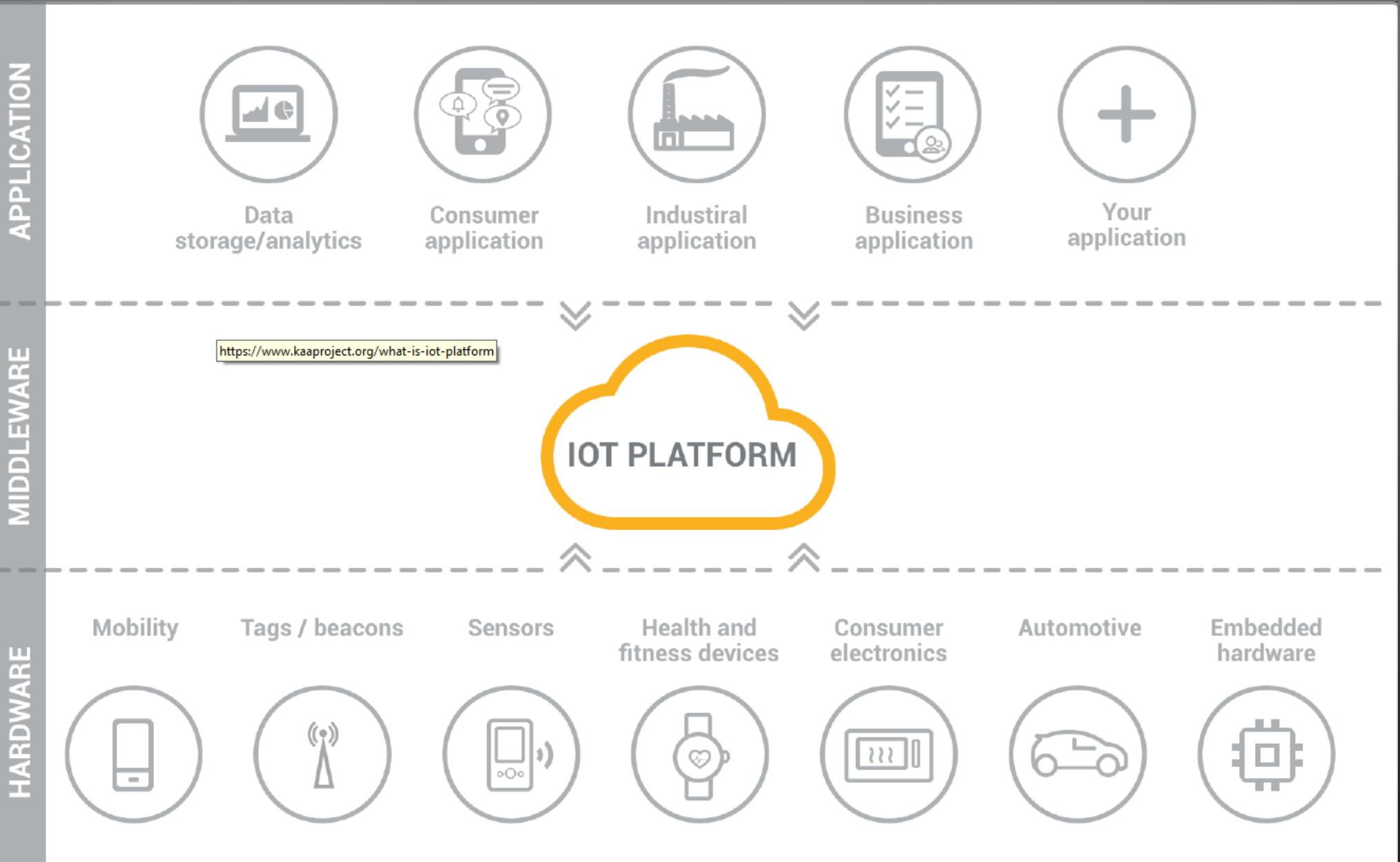
Peak Data Rate vs Maximum Range



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.



# 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).

# PLUG 'EM ALL



# WORKSHOP



APRENDIENDO EL INTERNET DE LAS  
COSAS...

...A BASE DE PRÁCTICA!!!



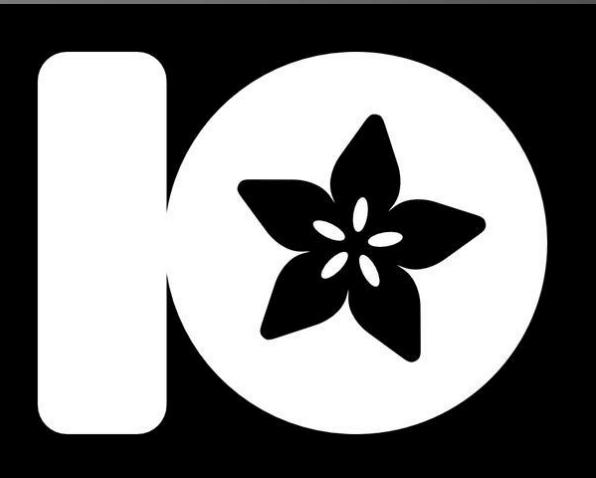
# ADAFRUIT IO

¿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

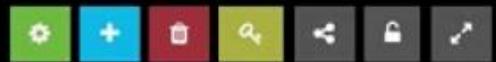


## IO Free

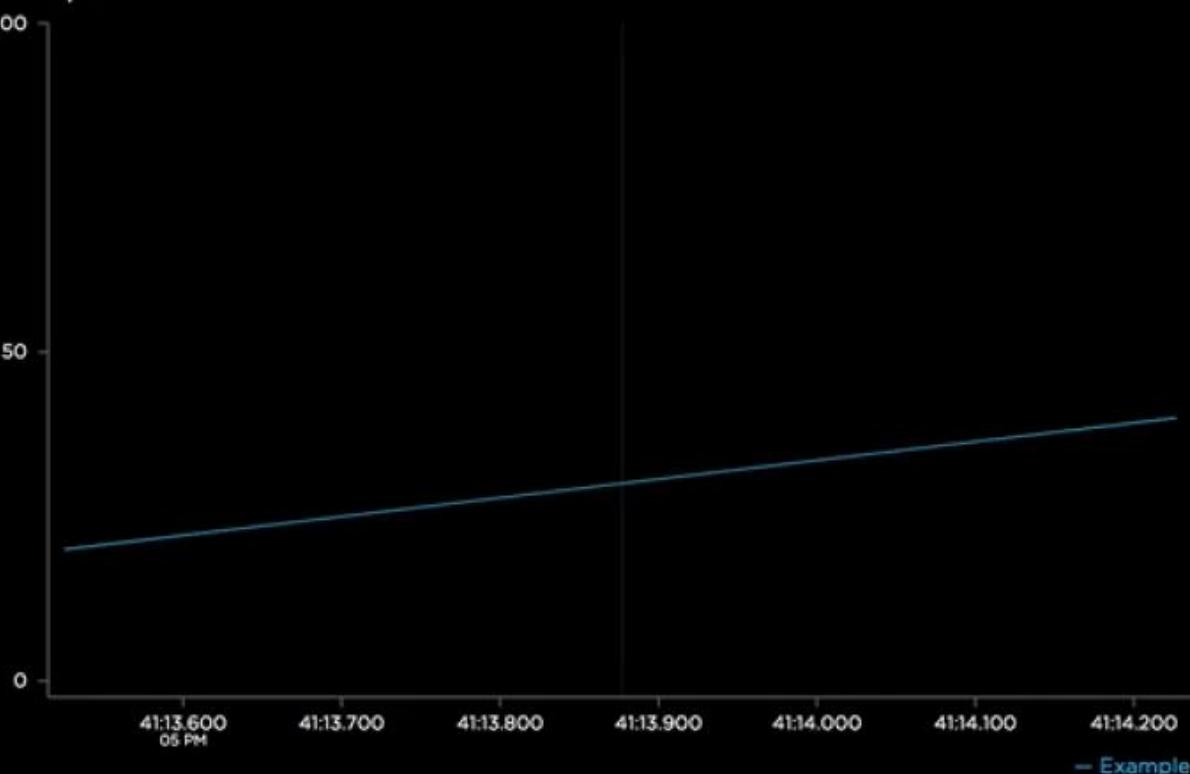
30 data points per minute  
30 days of data storage  
10 feeds  
5 dashboards  
Community support  
Projects and guides

[SHOP](#)[BLOG](#)[LEARN](#)[FORUMS](#)[VIDEOS](#)

uniontownlabs / Dashboards / Example Dashboard



Example



Example



Example



# 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  
[Get](#)

**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

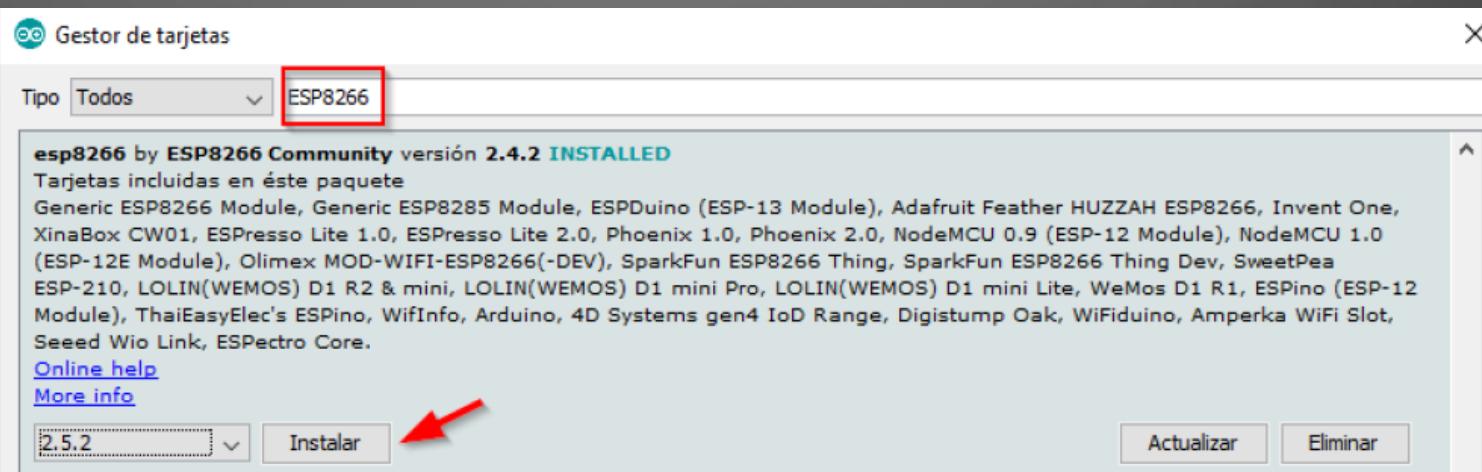
2. Copiar y pegar la siguiente URL ->

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)



3. Herramientas-> Placa-> Gestor de tarjetas...

4. Buscar por ESP8266 e instalar “esp8266 by ESP8266 Community”



# SOFTWARE

Añadir librerías al IDE de Arduino

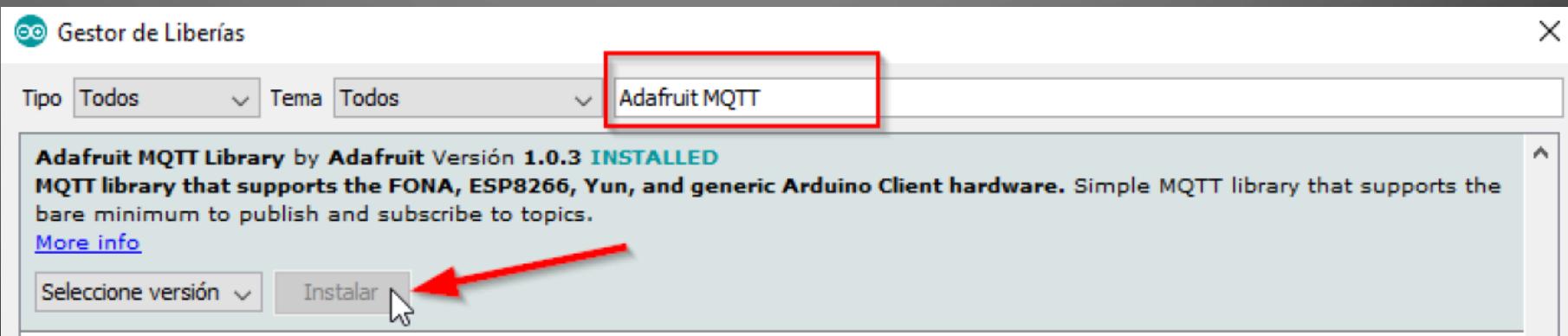
1. Programa-> Incluir Librerías-> Administrar Bibliotecas...

2. Buscar las siguientes librerías e Instalarlas:

“ArduinoHttpClient”

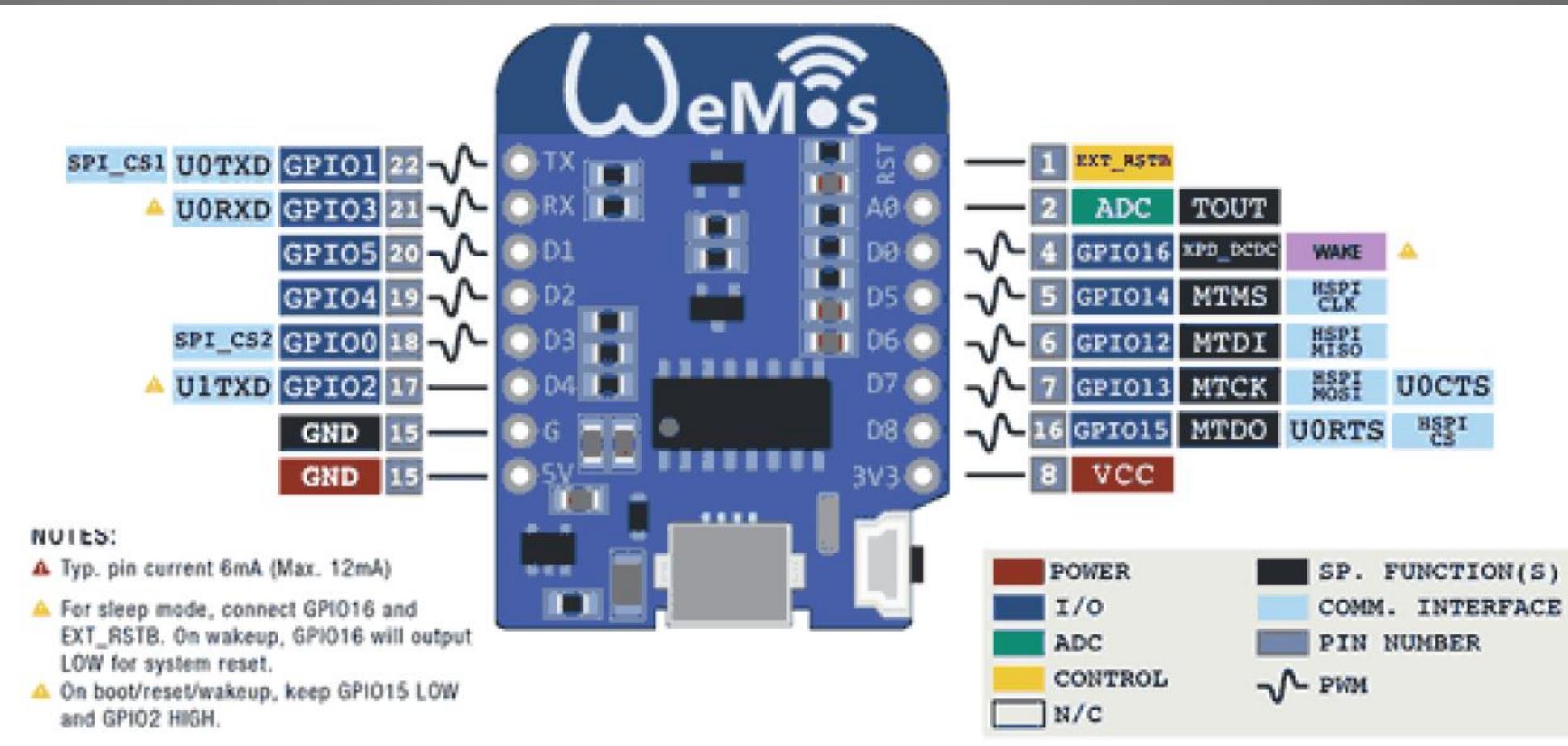
“Adafruit IO Arduino”

“Adafruit MQTT”



3.Crear cuentas de usuario en AdafruitIO

# HARDWARE: WEMOS D1 MINI

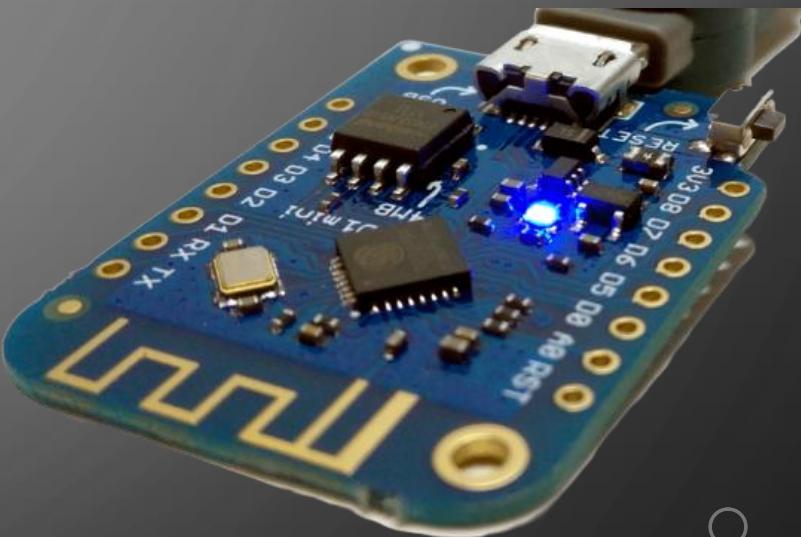


# HARDWARE

Probar que todo ha ido bien.. (i)

Cargar el clásico “Hola Mundo”

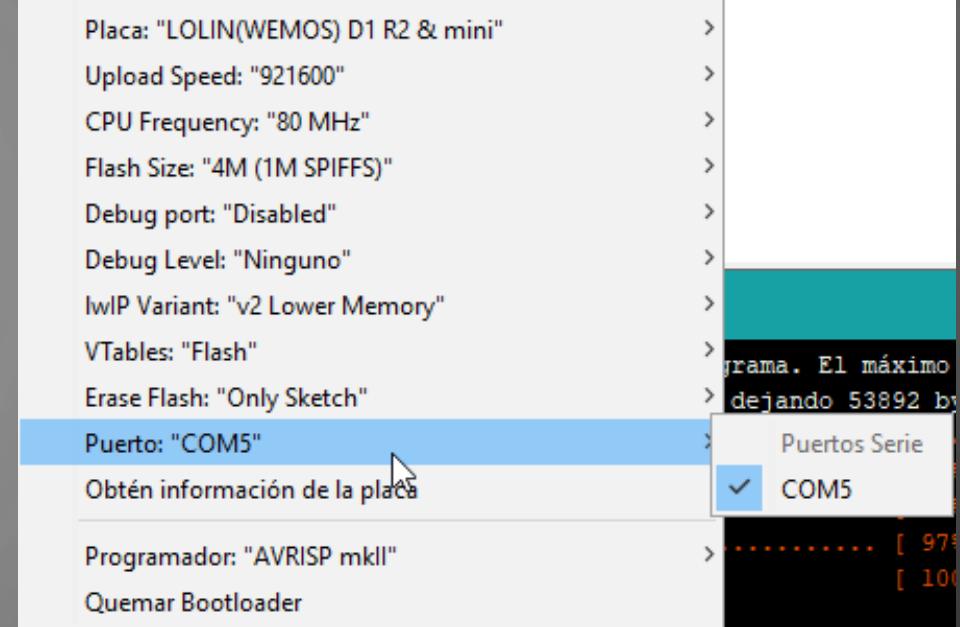
Archivo-> Ejemplos-> 01.Basics-> Blink



# 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-> WiFiClient

2. Editar valores “SSID” y “passpasspass” por los de la red WiFi a conectar.

3. Verificar  y Subir  el código a la Placa para que el código se ejecute.

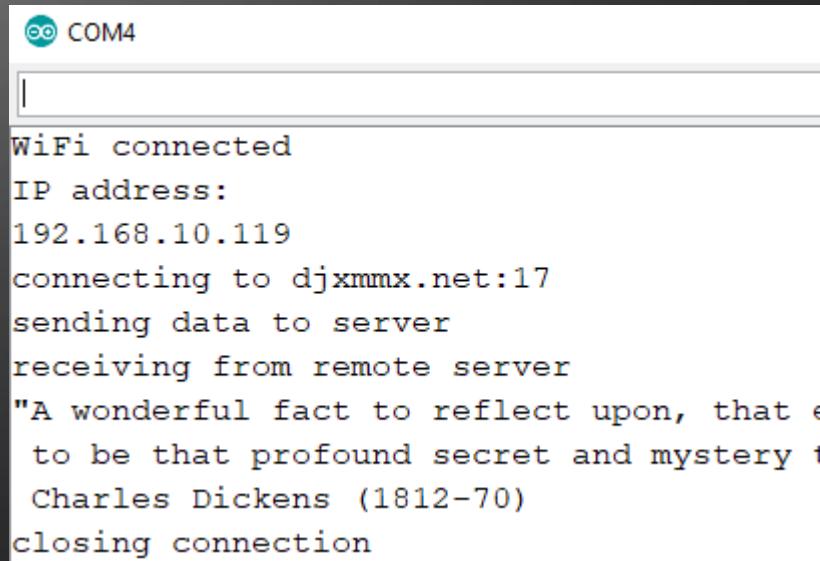
Abrir Monitor Serie 

```
WiFiClientBasic.h
#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");
```



COM4

WiFi connected

IP address:

192.168.10.119

connecting to djxmmx.net:17

sending data to server

receiving from remote server

"A wonderful fact to reflect upon, that e  
to be that profound secret and mystery t  
Charles Dickens (1812-70)

closing connection

# PLUG 'EM ALL



# WORKSHOP

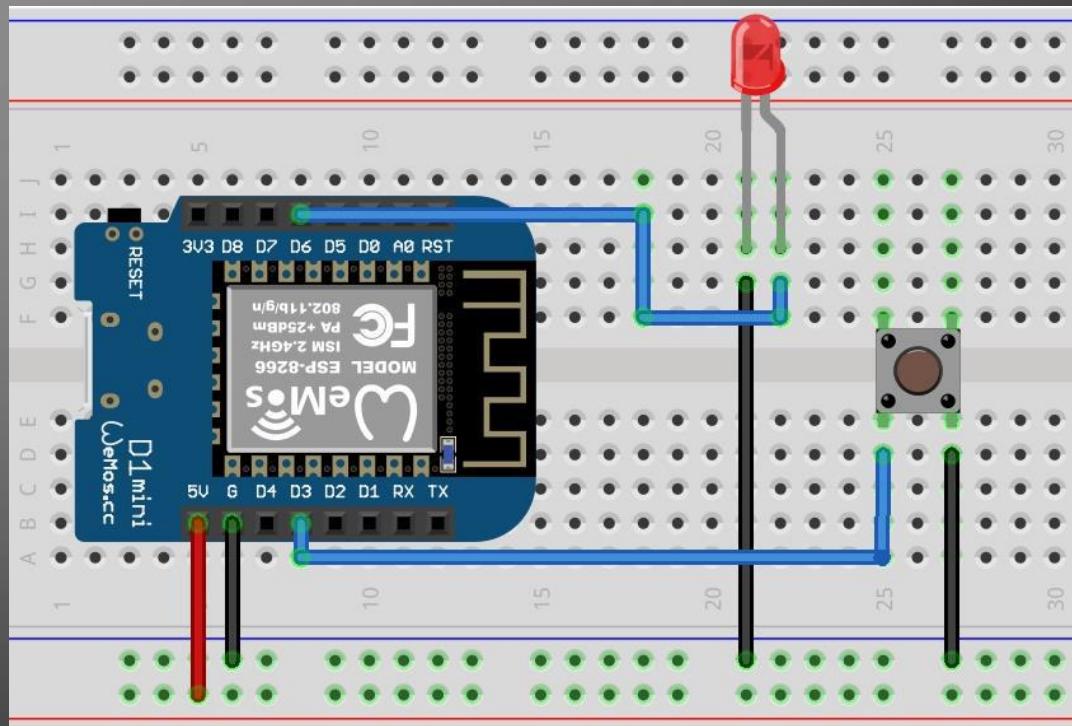
P R Á C T I C A S



# 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 Usernamey ActiveKey
  - WiFi SSIDy 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’

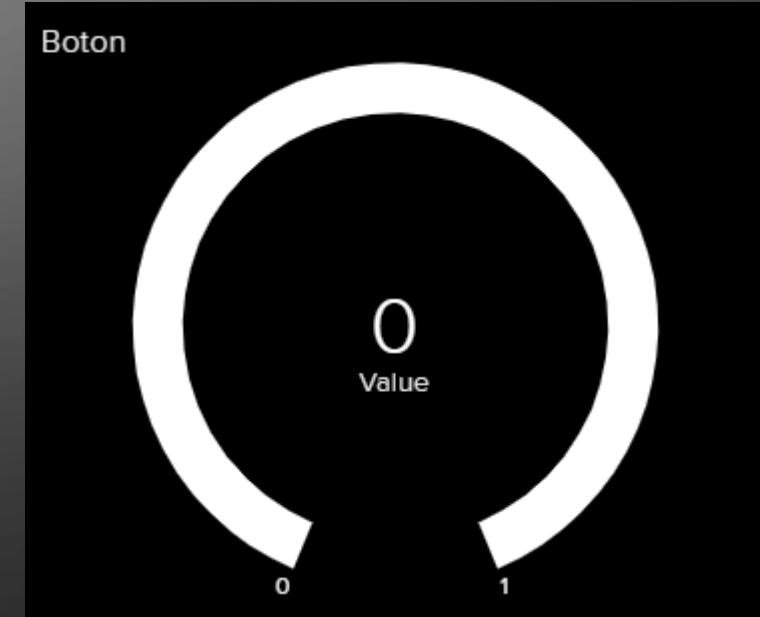
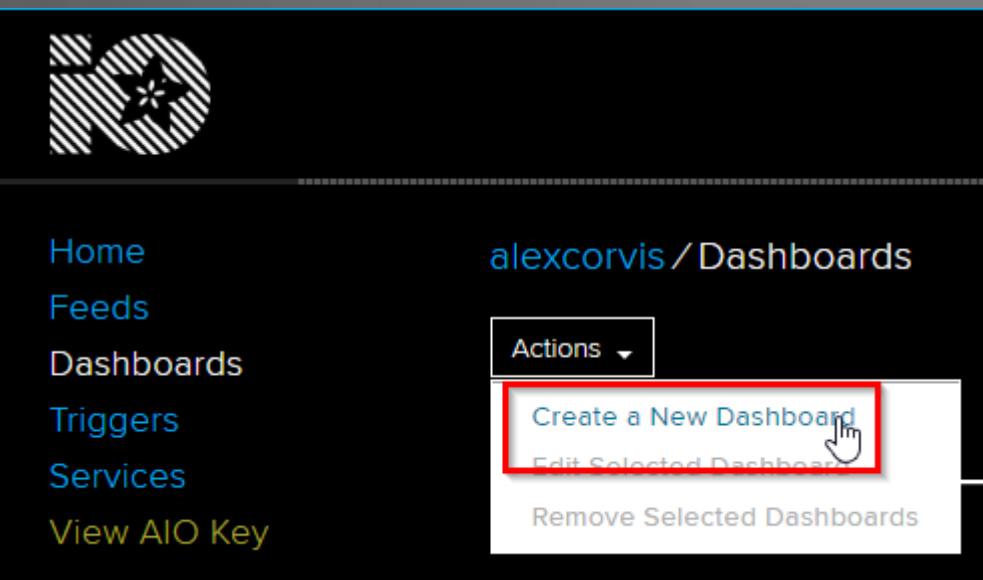
The screenshot shows the Adafruit IO web interface. At the top, there is a navigation bar with links for SHOP, BLOG, LEARN, FORUMS, and VIDEOS. On the far right, it shows 'Hello, Becky Stern | Sign Out | My Account | Wishlists' and '0 Items'. The main content area has a sidebar on the left with links for Profile, Feeds (which is highlighted in blue), Groups, Dashboards, Triggers, Settings, Guide and Tips, Adafruit IO Forum, API Documentation, and Blog/Changelog. The main panel displays a table titled 'bekathwia / Feeds'. The table has columns for Actions, Name, Key, Last Value, and Recorded. There are three entries:

Actions	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

# PRÁCTICA 01 –ENTRADA DIGITAL

## Montaje Botón + Feed (iv)

8. Crear un Dashboard en AdafruitIO y llamarlo “AGORA”
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 1



# 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**

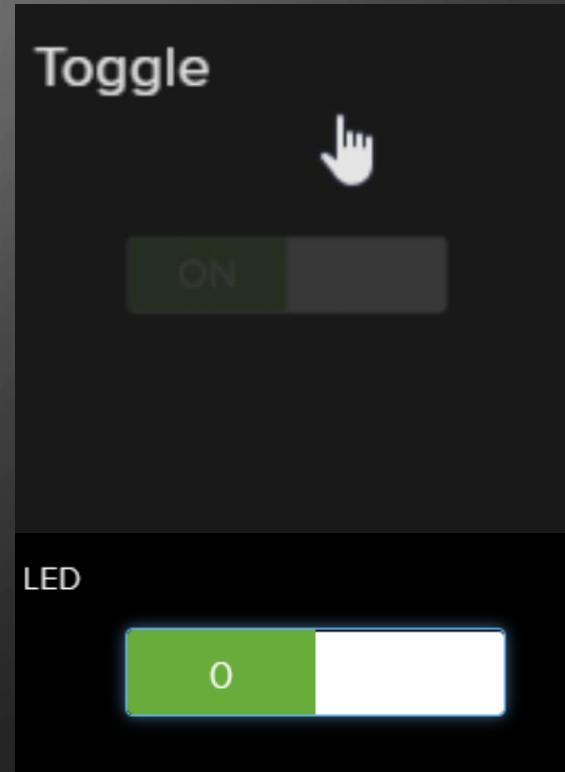
```
#define IO_USERNAME "alexcorvis"
#define IO_KEY "ffef62b3e5e44[REDACTED]"
```

```
// visit io.adafruit.com if you need to create an account,  
// or if you need your Adafruit IO key.  
  
#define IO_USERNAME    "XXXXXXXXXX"  
#define IO_KEY          "XXXXXXXXXXXXXXXXXXXXXX"  
  
/* ***** WIFI Configuration ***** /  
  
#define WIFI_SSID        "XXXXXXXXXXXX"  
#define WIFI_PASS        "XXXXXXXXXXXXXX"
```

# PRÁCTICA 02 –SALIDA DIGITAL

## Montaje LED (i)

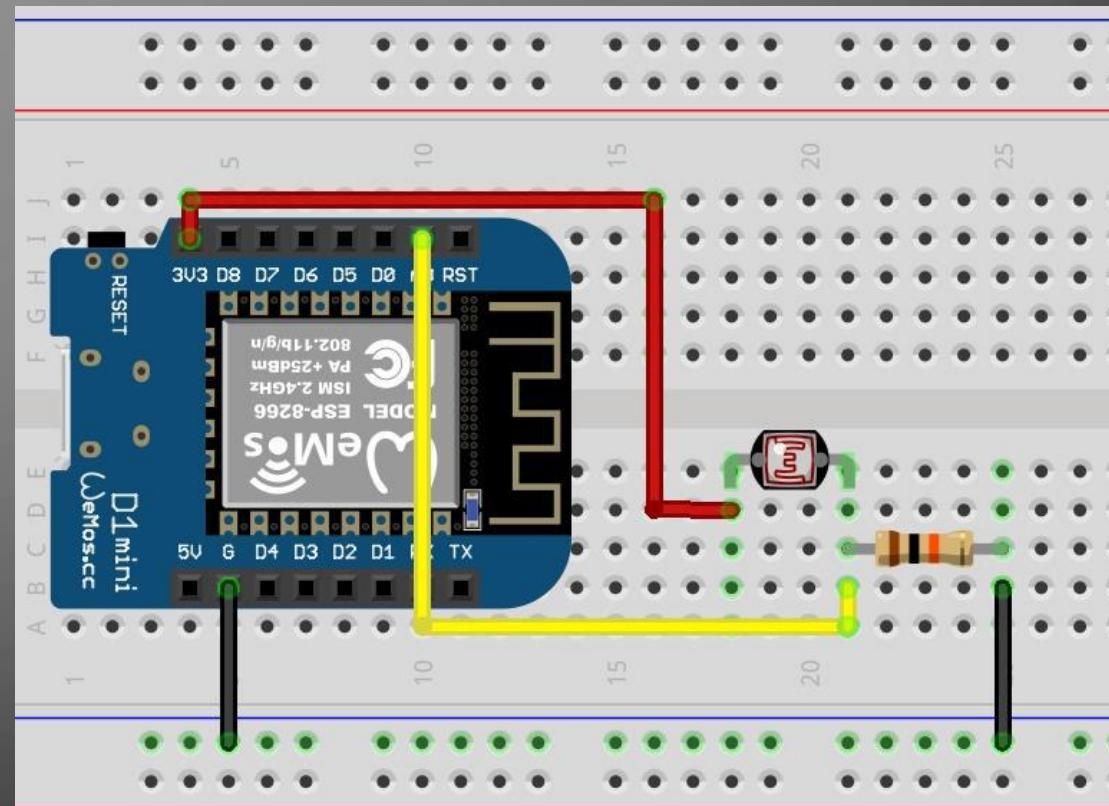
- 1.- Crear un nuevo Feed llamado ‘Led’
- 2.- Añadir un bloque tipo ‘Toggle’ asociado al Feed ‘Led’
- 3.- Editar los campos Button On Text y Button Off Text con los valores 1 y 0 respectivamente
- 4.- Descargar el código Practica\_02 y abrirlo con el Arduino IDE
- 5.- Actualizar la configuración para que coincidan tus credenciales
  - “Adafruit IO Username” y “ActiveKey”
  - “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)

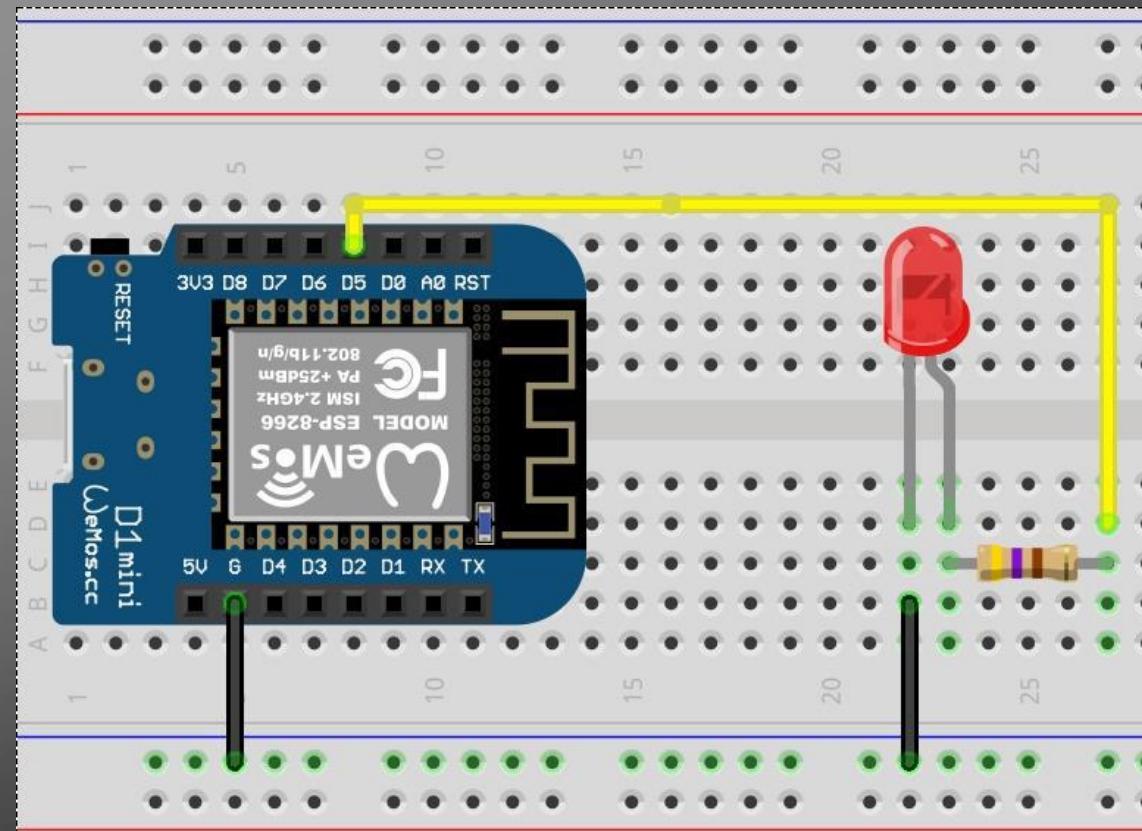
- 2.- Crear un nuevo **Feed** llamado “**análogico**”
- 3.- Añadir un bloque tipo “**Gauge**” asociado al **Feed** “**análogo**”
- 4.- Editar los campos “**Gauge Min Value**” y “**Gauge Max Value**” con los valores 0 y 1024 respectivamente
- 5.- Descargar el código Practica\_03 y abrirlo con el Arduino IDE
- 6.- Actualizar la configuración para que coincidan tus credenciales
  - Adafruit IO Usernamey ActiveKey
  - WiFi SSIDy 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)

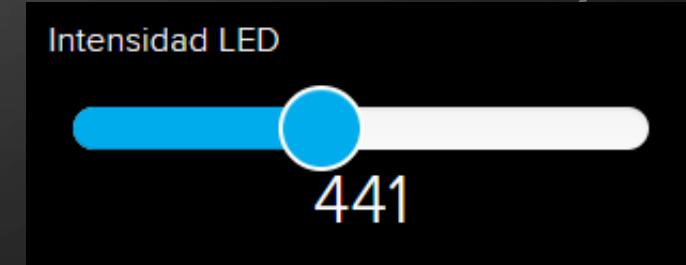
1. Realizar montaje del circuito según el siguiente esquema



# PRÁCTICA 04 – SALIDA ANALÓGICA

## Montaje Led (ii)

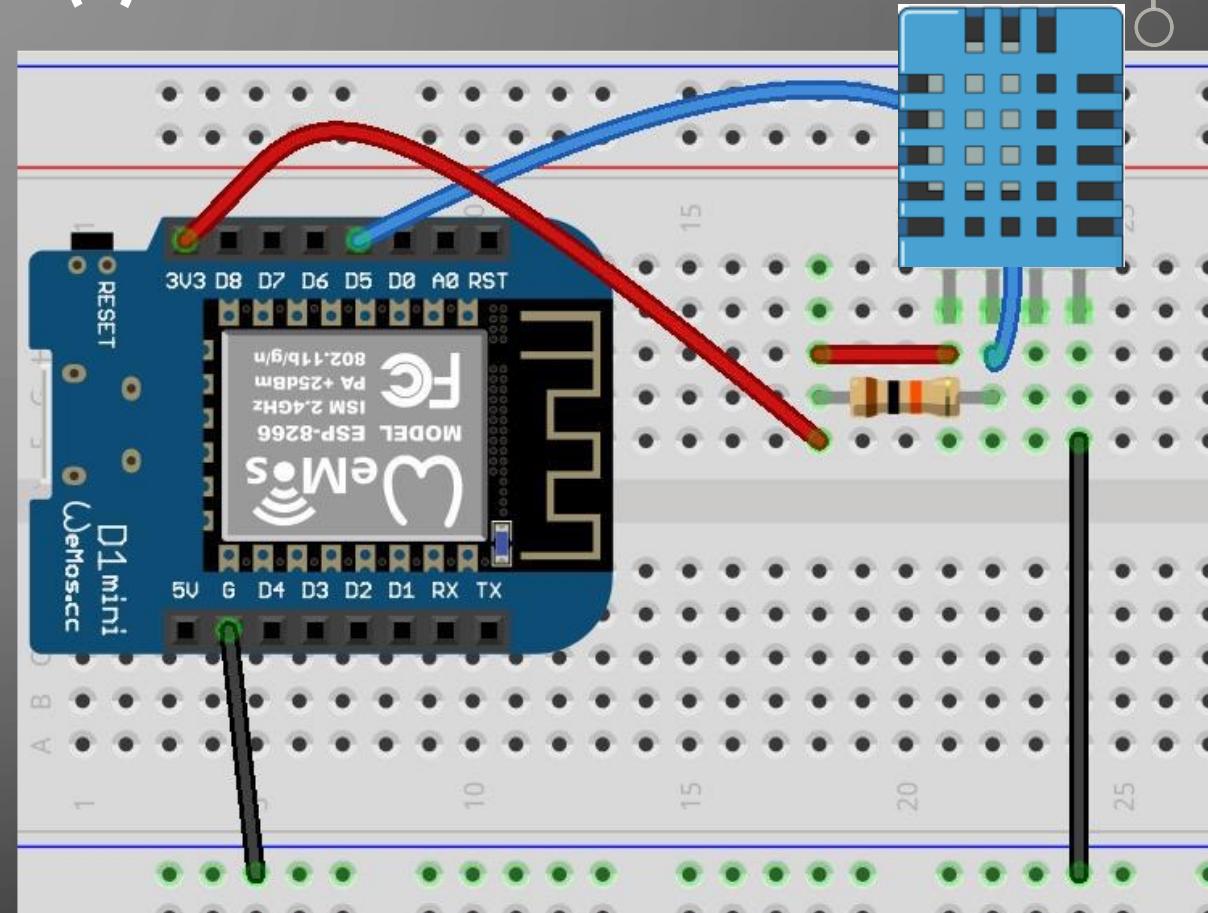
- 2.- Crear un nuevo Feed llamado ‘Intensidad’
- 3.- Añadir un bloque tipo ‘Slider’ asociado al Feed ‘Intensidad’
- 4.- Editar los campos “Slider Min Value” y “Slider Max Value” con los valores 0 y 1024 respectivamente
- 5.- Descargar el código Practica\_04 y abrirlo con el Arduino IDE
- 6.- Actualizar la configuración para que coincidan tus credenciales
  - Adafruit IO Usernamey ActiveKey
  - WiFi SSIDy 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 – TEMPERATURA Y HUMEDAD

## Montaje Sensor DHT11 (i)

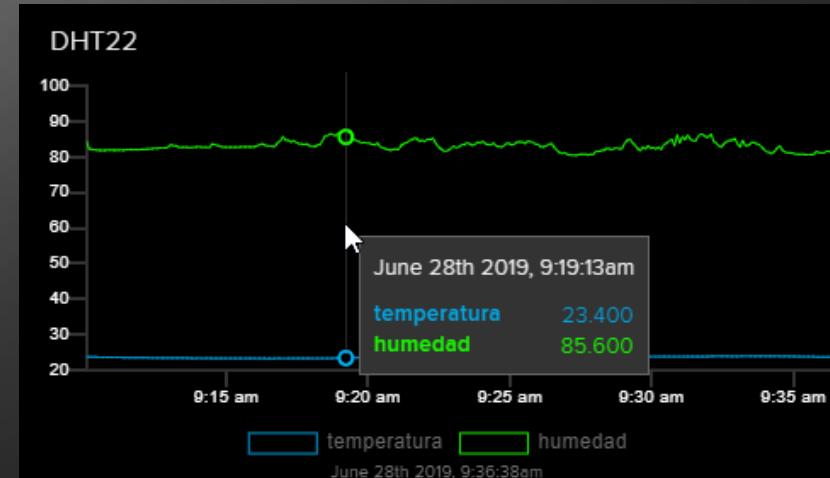
1. Realizar montaje del circuito según el siguiente esquema



# PRÁCTICA 05 – TEMPERATURA Y HUMEDAD

## Montaje Sensor DHT11 (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 “ActiveKey”
  - “WiFi SSID” y “Password”
- 8.- Verificar  y Subir  el código a la placa
- 9.- Comprobar el correcto funcionamiento del circuito



# RESUMEN ADAFRUIT IO API

## Montaje Sensor DHT11 (ii)

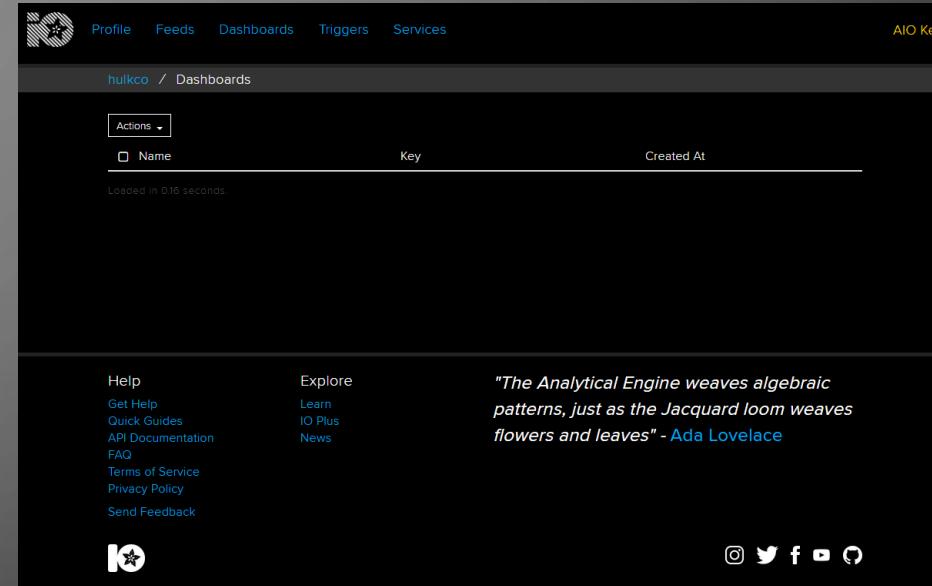
- Crear un Feed
  - AdafruitIO\_Feed\*newFeed= io.feed("newfeed");
- Añadir datos al Feed(Publicación) → voidloop()
  - io.run();
  - feedName→ save(value);
- Recibir datos de un Feed(Suscripción) → voidsetup()
  - io.connect();
  - newFeed→ onMessage(handleMessage);
  - voidhandleMessage(AdafruitIO\_Data\*data)
- Conversión de datos

**data** →**toPinLevel()**  
**data** →**value()**  
**data** →**toChar()**

**data** →**toString()**  
**data** →**toDouble()**  
**data** →**toFloat()**

**data** →**toInt()**  
**data** →**toLong()**  
**data** →**toBool()**

**data** →**isTrue()**  
**data** →**isFalse()**  
...



BASADO EN EL TRABAJO DE:

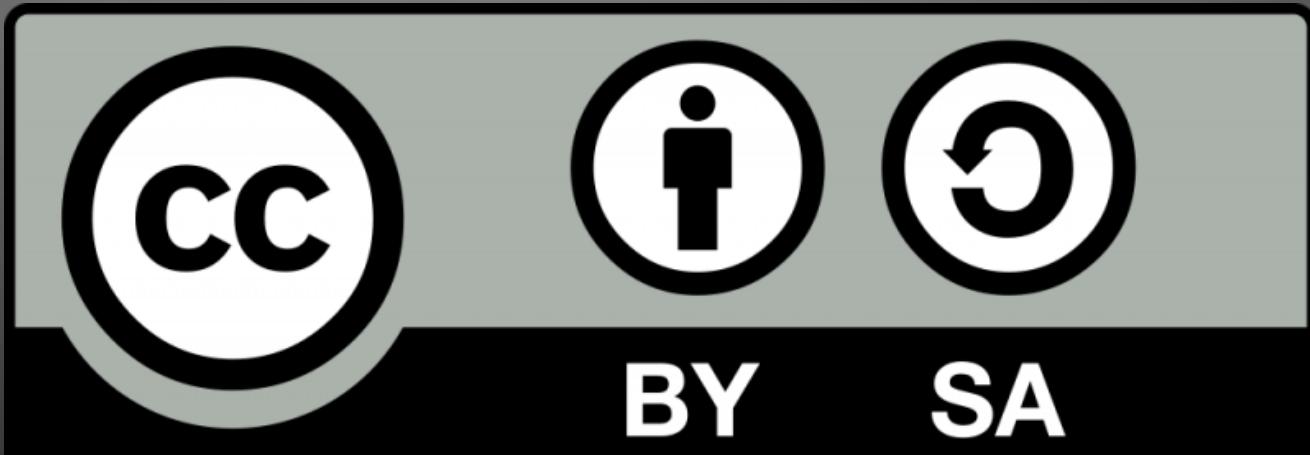


ALEJANDROJUANGARCÍA



@ALEXCORVIS84

@ MAKERSASTURIAS



# Gracias Totales!!!

greynaga@outloo.com

 @gsreynaga

