

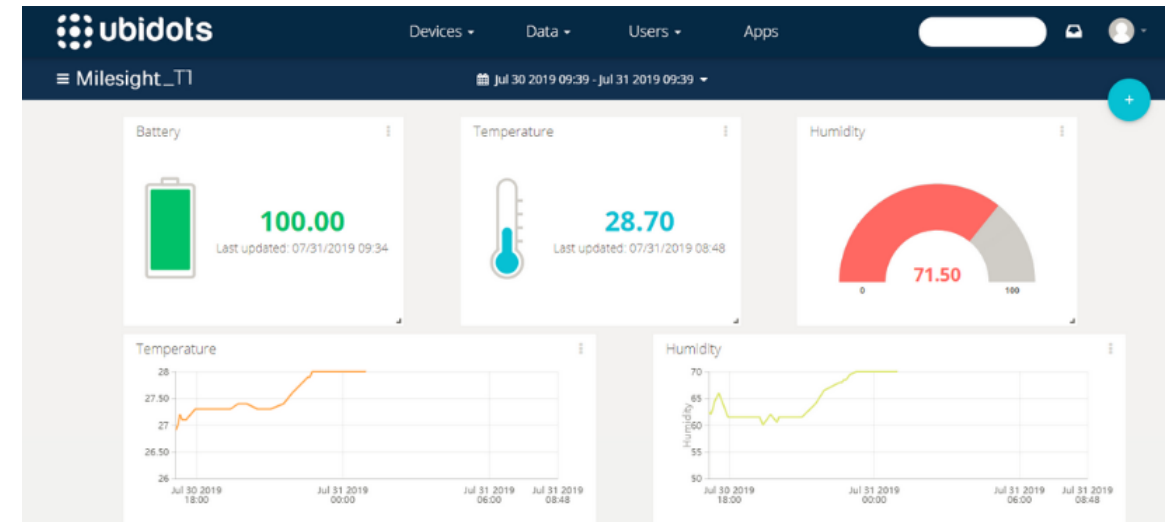
Comunicación wifi con NodeMCU ESP8266 y Ubidots Automatization con Microcontroladores

INTRODUCCIÓN UBIDOTS
FUNCIONALIDADES
IMPLEMENTACION
EJEMPLO DE APLICACIÓN



➤ ¿Qué Ubidots?

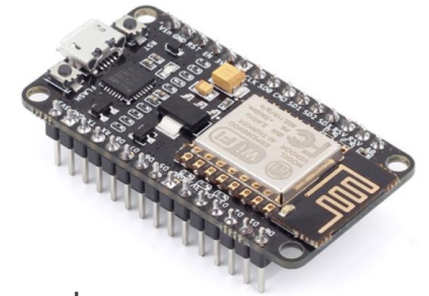
- Software de análisis de información y toma de decisiones(alertas)
- Fácil de usar.
- Interfaz de interacción entre usuarios y sistemas IoT (Dashboard)
- Mapas de datos en tiempo real



Datos ➔ Puntos

Que es es una tarjeta de Desarrollo

NodeMCU ESP8266
Dev kit



ESP8266
MICROCONTROLADOR

CARACTERÍSTICAS

Protocolos de comunicación:
HTTP, MQTT y TCP/UDP
115200 baudios

Voltaje de operación:
2.5 V a 3.6 V

Dispositivo
electrónico de
Integración

Módulo wifi integrado
WIFI

IDE Arduino

Puede programarse mediante el
IDE de Arduino

Node MCU → Firmware

Soporte lógico inalterable que controla los
circuitos electrónicos de cualquier dispositivo

► Ubidots STEM (depende del plan)

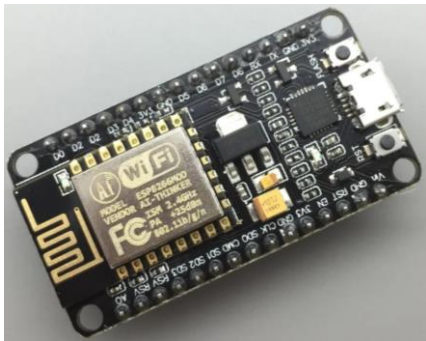
- Hasta 10 dispositivos (primeros 3 gratis).
- Hasta 10 variables por dispositivo
- Lectura de datos: Hasta 4000 puntos al día (sumando todos los dispositivos).
- Extracción de datos: Hasta 500,000 puntos al día
- Tasa de procesamiento de datos: 1 solicitud/segundo.
- Hasta 10 dashboards con 10 ventanas de visualización (widgets) cada uno.
- Hasta 10 SMS, 100 correos electrónicos y una llamada gratis al mes (solo en USA y Canadá).
- Almacenamiento de datos por 1 mes

► Ubidots

- De 1 a miles de dispositivos.
- Hasta 20 variables por dispositivo.
- Lectura de datos: Desde 2 millones de puntos al día (sumando todos los dispositivos).
- Extracción de datos: Desde 20 millones de puntos al día.
- Tasa de procesamiento de datos: 4 solicitudes/segundo.
- Dashboards y ventanas de visualización ilimitados.
- SMS, correos electrónicos y llamadas ilimitados.
- Almacenamiento de datos por 2 años.

Tarjeta de desarrollo

Módulo Wifi Node
MCU ESP8266 Dev Kit

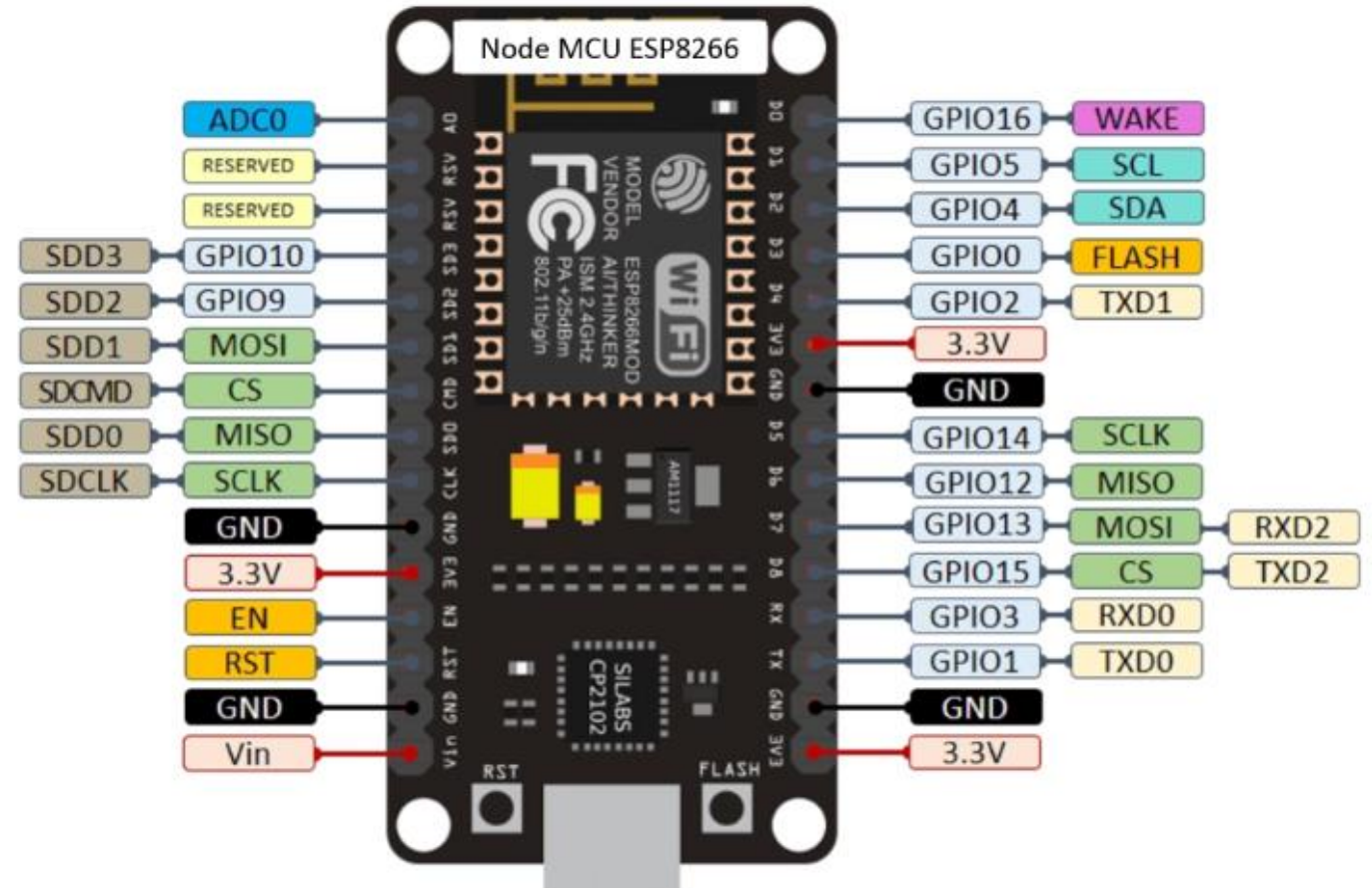


1 Pin analógico

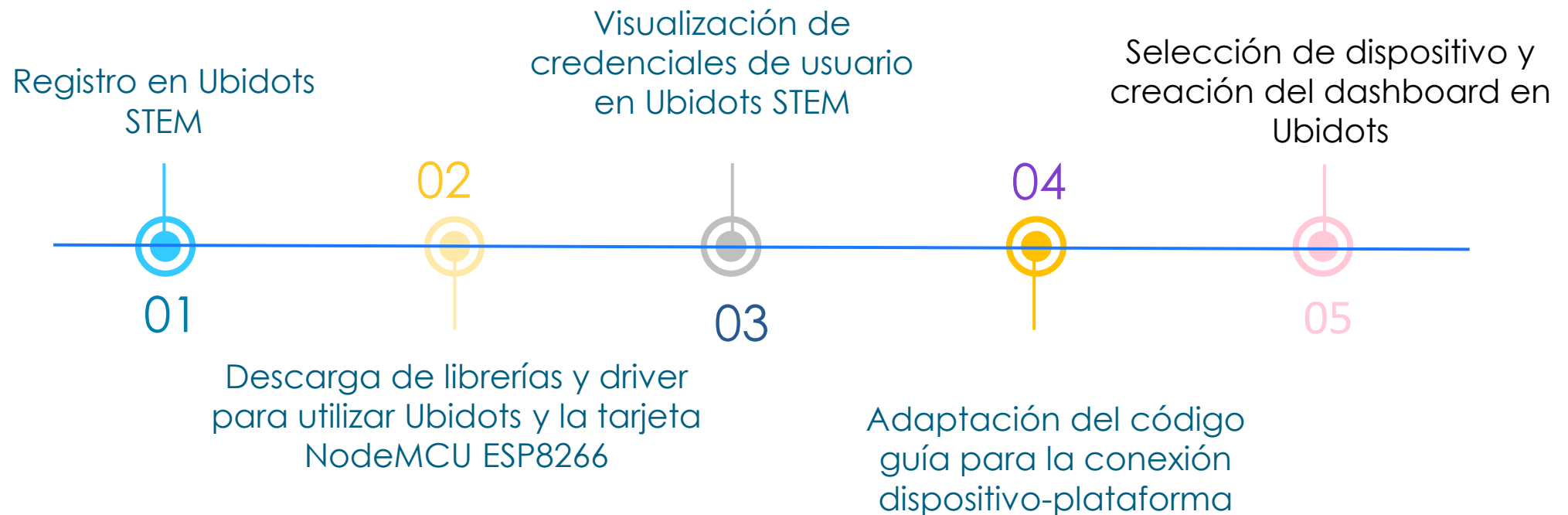
Voltaje máximo de entrada: 3,1V

Resolución: $3,1V/2014 = 3,17mV$

16 Pines GPIO



Implementación




Pasos implementación

► Registro en Ubidots STEM

How would you like to use Ubidots?

Whether you're prototyping or going to market, Ubidots is designed to accelerate your IoT initiatives.


For Educational or Personal Use



Join **60,000+** students, makers and professors using our **FREE** Ubidots STEM platform to prototype, learn, or teach IoT.

[TAKE ME TO UBIDOTS STEM](#)


For Business



Join **1,000+** System Integrators, OEMs and IoT Entrepreneurs building connected products and services with Ubidots.

[CONTINUE →](#)

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



Devices ▾Data ▾

Plans and Billing

My account

API Credentials

Plans and Billing

Usage

Plans and Billing

Account balance ⓘ

\$0

Add balance

Your plan ⓘ

STEM

Change plan

Billing Information ⓘ

Payment methods

NAME	TYPE	DESCRIPTION	DEFAULT	ACTIONS
No payment methods found				

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

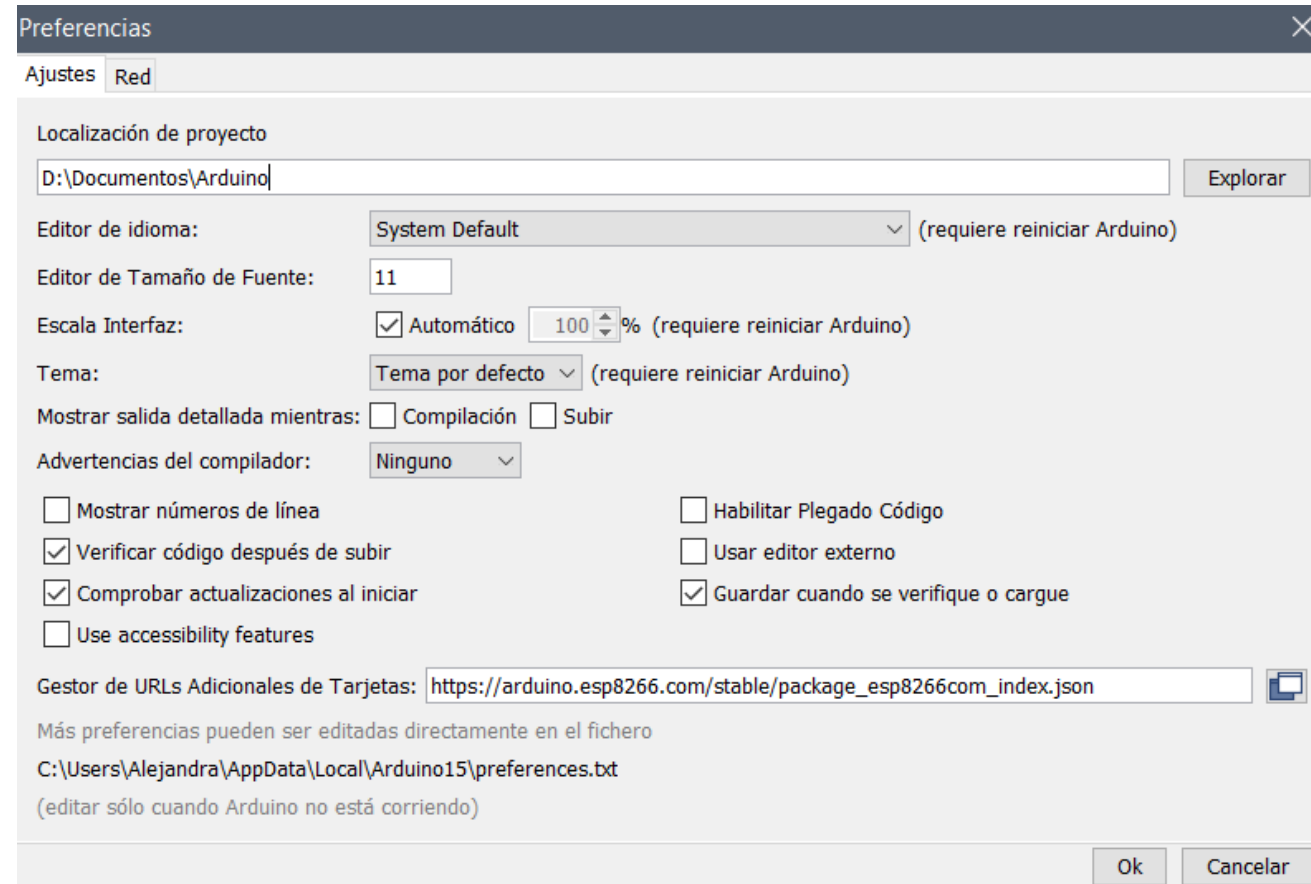
02 Instalación de librerías y driver

Paso 1. Agregar vínculo al gestor de tarjetas

- IDE de Arduino > Archivo > Preferencias > Gestor de URLs Adicionales de Tarjetas
- Pegar:
http://arduino.esp8266.com/stable/package_esp8266com_index.json
- Ok

¿Qué se necesita?

- Tarjeta NodeMCU ESP8266
- Cable micro USB a USB
- IDE de Arduino
- Driver CH340G/CH341SER de la tarjeta NodeMcu



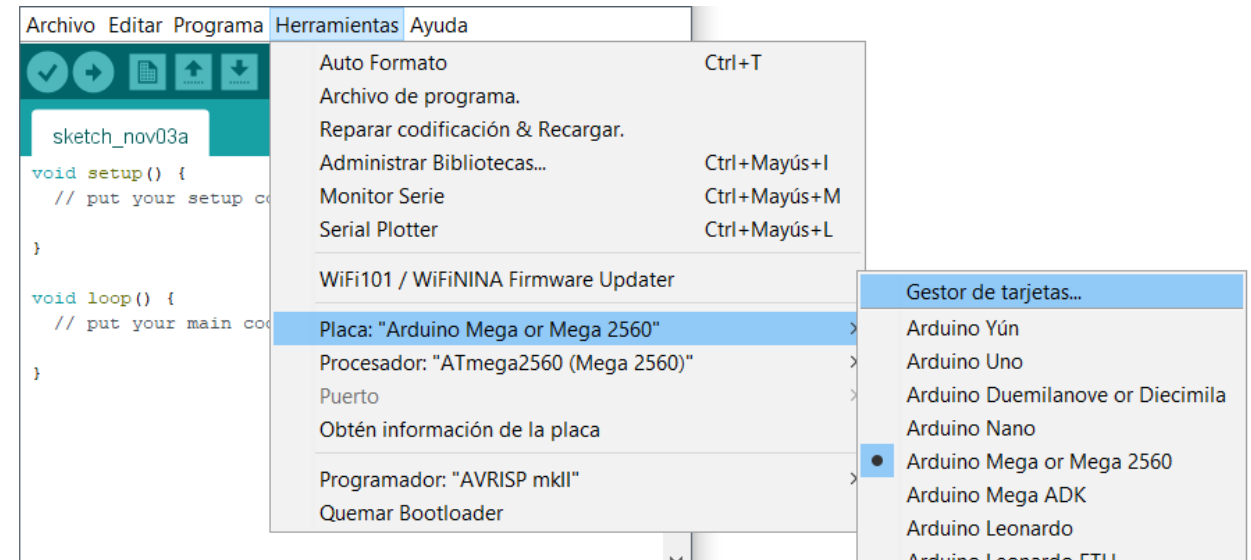
Referencia repositorio GitHub Ubidots ESP8266 <https://github.com/ubidots/ubidots-esp8266/>

02 Instalación de librerías y driver

Paso 2. Descargar librería ESP8266

- IDE de Arduino > Herramientas > Placa > Gestor de Tarjetas
- Buscar ESP8266
- Seleccionar la versión 2.7.4*
- Instalar

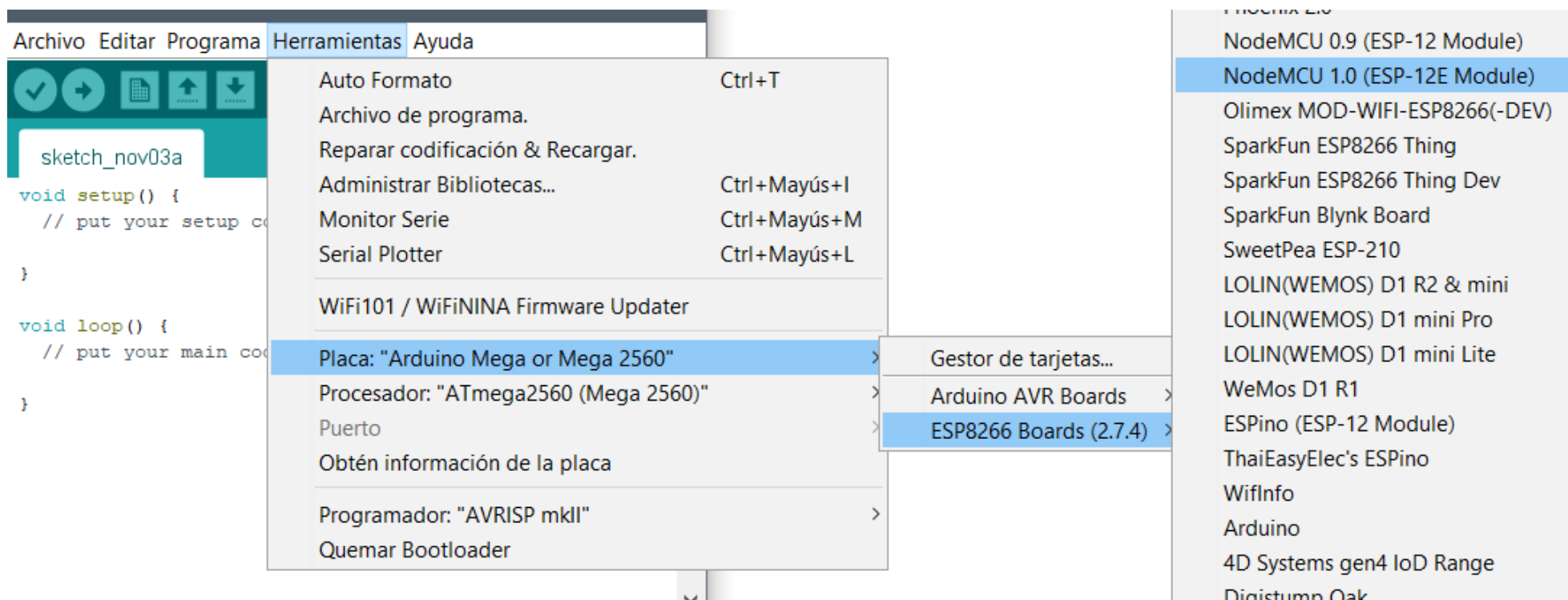
*En ocasiones instalar la última versión puede generar errores en el reconocimiento de la tarjeta, por eso en diferentes foros se recomienda utilizar la versión 2.7.4. Puede ser necesario desactivar el antivirus si se presenta algún error en la descarga de la librería



02 Instalación de librerías y driver

Paso 3. Verificar disponibilidad de la tarjeta

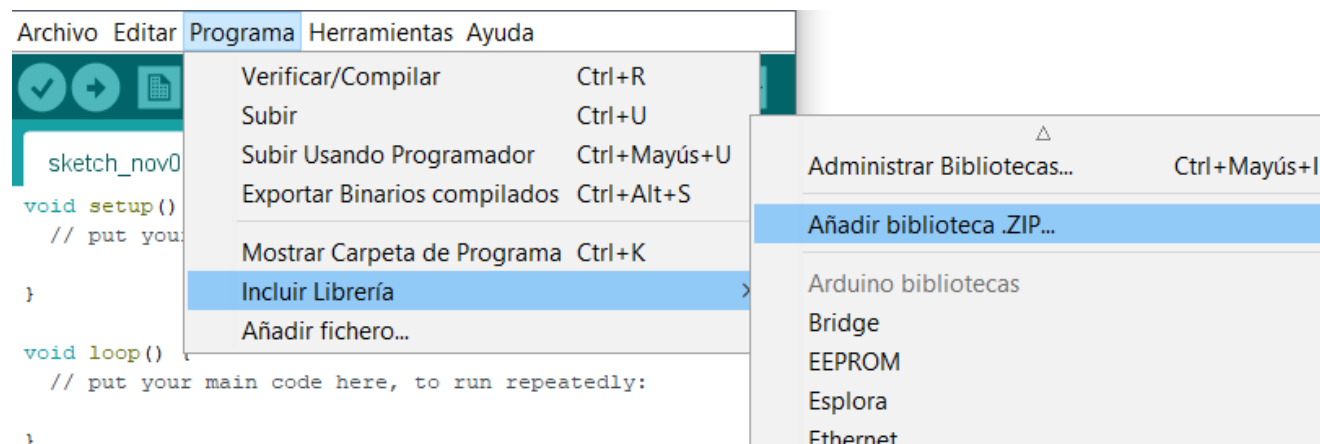
- IDE de Arduino > Herramientas > Placa
- Verificar que aparezca en ESP8266 Boards la placa NodeMCU 1.0 (ESP-12E Module)



02 Instalación de librerías y driver

Paso 4. Descargar librería Ubidots

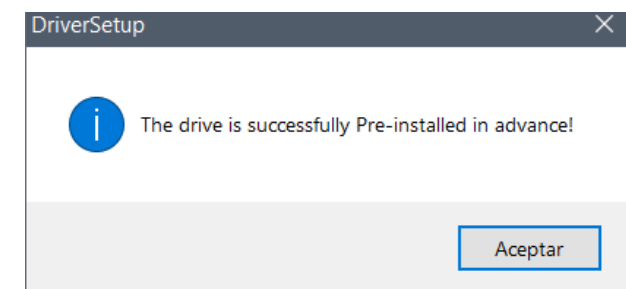
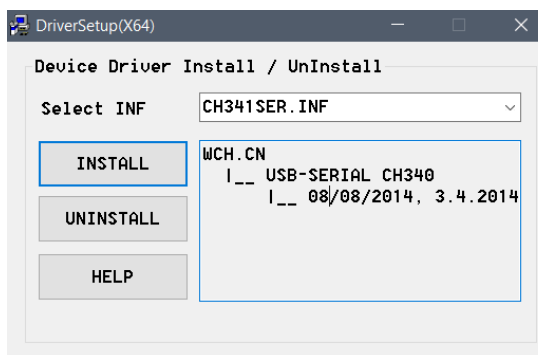
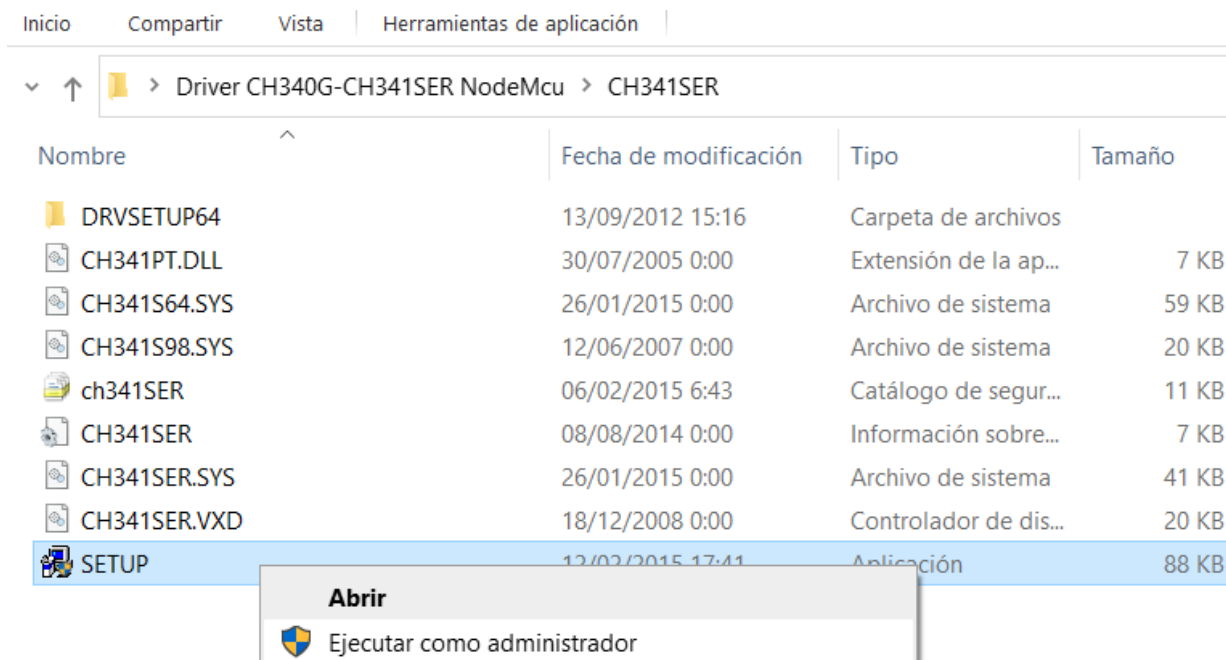
- <https://github.com/ubidots/ubidots-nodemcu/archive/master.zip>
- Descargar el archivo .ZIP
- IDE de Arduino > Programa > Incluir Librería > Añadir biblioteca .ZIP
- Seleccionar el archivo .ZIP descargado



02 Instalación de librerías y driver

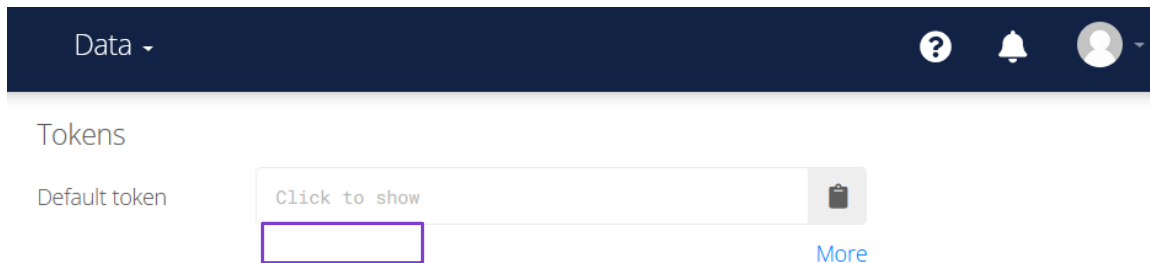
Paso 5. Descargar el driver CH340G-CH341SER

- Ir a la página: <https://drive.google.com/file/d/1eDnXFbN-INOkGhKhTHKXnOZ1wQ0xIYwJ/view>
- Descargar y extraer el archivo .ZIP
- CH341SER > Clic derecho en SETUP > Ejecutar como administrador
- Install
- Aceptar

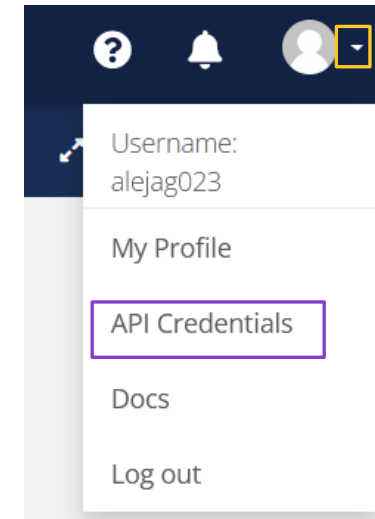


03 Obtención de credenciales de usuario en Ubidots STEM

- Ir a Ubidots STEM e iniciar sesión.
- Desplegar opciones de usuario > API Credentials.
- En Tokens, clic en mostrar.
- El código mostrado corresponde al Token asignado al usuario y se utilizará posteriormente.



Token asignado



04 Adaptación del código guía para la conexión dispositivo-plataforma (enviar datos)

- Conectar la tarjeta y en el IDE de Arduino seleccionar la placa NodeMCU 1.0 (ESP-12E Module)
- Ir a la página: <https://help.ubidots.com/en/articles/513312-connect-a-nodemcu-esp8266-to-ubidots-over-http>
- Copiar el código para enviar datos a Ubidots
- Definir el tipo de variable asociada al valor a medir según sea el caso.
- Agregar el Token, nombre de la red wifi (WIFI_SSID) y contraseña (WIFI_PASS).

Código Guía

```

/*****
 * Include Libraries
 *****/

#include "Ubidots.h"

/*****
 * Define Instances and Constants
 *****/

const char* UBIDOTS_TOKEN = "..."; // Put here your Ubidots TOKEN
const char* WIFI_SSID = "..."; // Put here your Wi-Fi SSID
const char* WIFI_PASS = "..."; // Put here your Wi-Fi password

Ubidots ubidots(UBIDOTS_TOKEN, UBI_HTTP);

```

Ejemplo a partir del código guía

```

/*****
 * Include Libraries
 *****/

#include "Ubidots.h"

/*****
 * Define Instances and Constants
 *****/

int push_button = D1;

const char* UBIDOTS_TOKEN = "BBFF-8RkyJSM7QpyAea461ytLSGWQdz82c5";
const char* WIFI_SSID = "Aleja G";
const char* WIFI_PASS = "pechilomas";

Ubidots ubidots(UBIDOTS_TOKEN, UBI_HTTP);

```

04 Adaptación del código guía para la conexión dispositivo-plataforma (enviar datos)

Agregar el void setup para:

- Inicializar la comunicación a 115200 baudios
- Establecer la conexión wifi mediante el comando `ubidots.wifiConnect(WIFI_SSID, WIFI_PASS);`
- Definir pines como de entrada o salida según sea el caso.

```

/*****
* Main Functions
*****/

void setup() {

    Serial.begin(115200);
    ubidots.wifiConnect(WIFI_SSID, WIFI_PASS);
    // ubidots.setDebug(true); // Uncomment
this line for printing debug messages
}

```

```

/*****
* Main Functions
*****/

void setup() {

    Serial.begin(115200);
    ubidots.wifiConnect(WIFI_SSID, WIFI_PASS);
    // ubidots.setDebug(true); // Uncomment this line
for printing debug messages

    pinMode(push_button, INPUT);
}

```

04 Adaptación del código guía para la conexión dispositivo-plataforma (enviar datos)

Agregar el void loop para:

- Especificar la lectura u obtención de cada variable.
- Agregar la variable a Ubidots mediante el comando `ubidots.add("Variable_Name", value)`.
- Enviar los datos a Ubidots

```
void loop() {

    float value1 = random(0, 9) * 10;
    float value2 = random(0, 9) * 100;
    float value3 = random(0, 9) * 1000;
    ubidots.add("Variable_Name_One", value1); // Change for
    your variable name
    ubidots.add("Variable_Name_Two", value2);
    ubidots.add("Variable_Name_Three", value3);

    bool bufferSent = false;
    bufferSent = ubidots.send(); // Will send data to a
    device label that matches the device Id

    if (bufferSent) {
        // Do something if values were sent properly
        Serial.println("Values sent by the device");
    }
    delay(5000);
}
```

```
void loop() {

    push_button = digitalRead(D1); // Lectura pulsador 1
    if (push_button == LOW)
    {
        push_button = 0;    //
    }
    else if (push_button == HIGH)
    {
        push_button = 1;    //
    }

    ubidots.add("push_button", push_button);

    bool bufferSent = false;
    bufferSent = ubidots.send();

    if (bufferSent)
    {
        Serial.println("Values sent by the device");
    }
    delay(10000);
}
```


04 Adaptación del código guía para la conexión dispositivo-plataforma (enviar datos)

De forma similar se pueden recibir datos desde Ubidots utilizando el comando `ubidots.get("device_name", "variable_name_one")` en vez de `ubidots.add`, siendo el primer argumento el nombre del dispositivo y el segundo el nombre de la variable (el resto del código guía es igual).

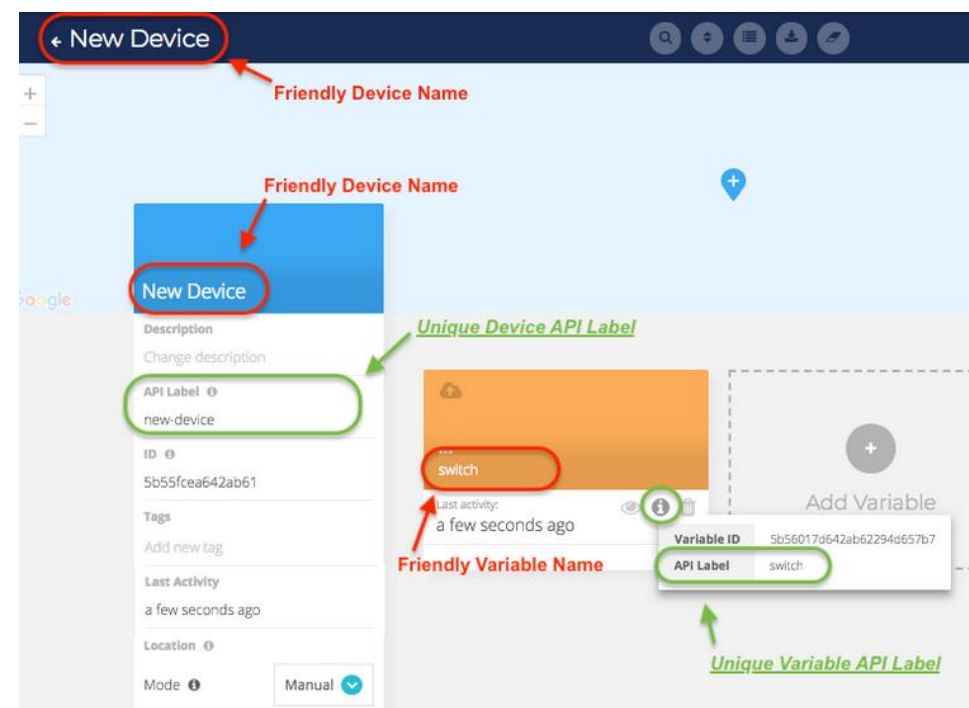
¿Dónde encontrar el nombre del dispositivo y de la variable?

```
void setup() {

  Serial.begin(115200);
  ubidots.wifiConnect(WIFI_SSID, WIFI_PASS);
  // ubidots.setDebug(true); //Uncomment this line for printing debug
  messages
}

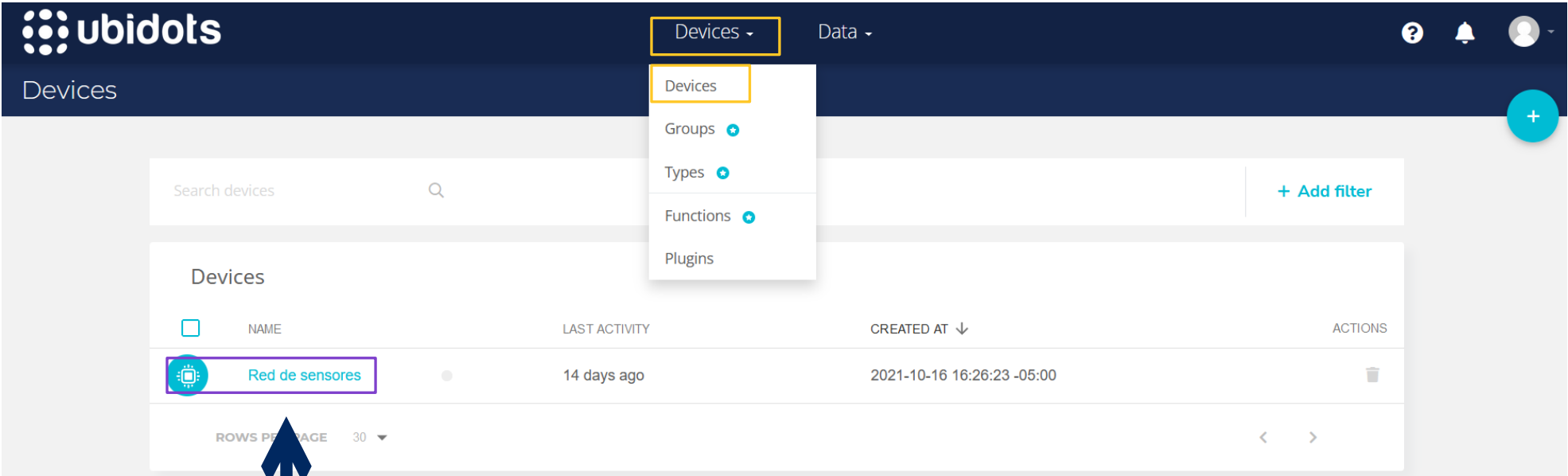
void loop() {
  /* Obtain last value from a variable as float using HTTP */
  float value = ubidots.get("device_name", "variable_name_one");

  // Evaluates the results obtained
  if (value != ERROR_VALUE) {
    Serial.print("Value:");
    Serial.println(value);
  }
  delay(5000);
}
```



05 Selección de dispositivo y creación de dashboard en Ubidots

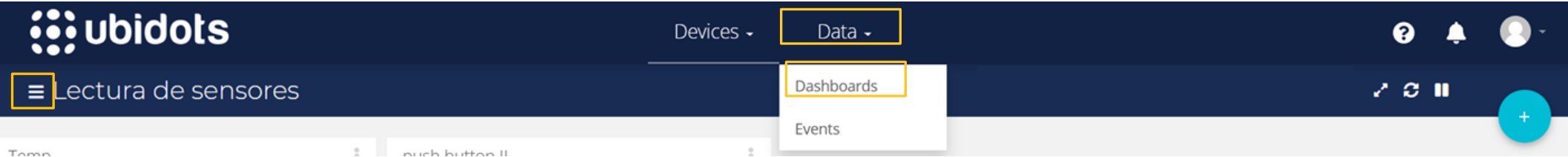
- Para visualizar los dispositivos conectados: Ubidots > Devices > Devices

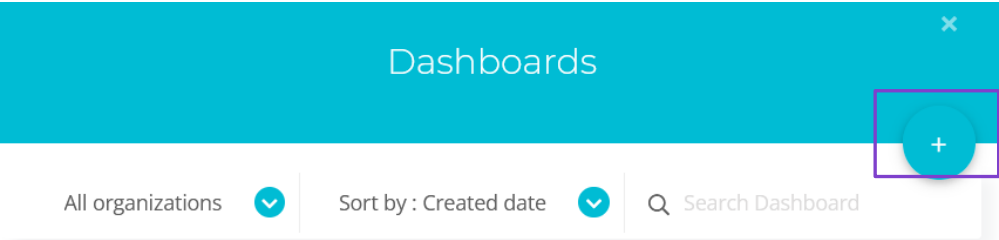


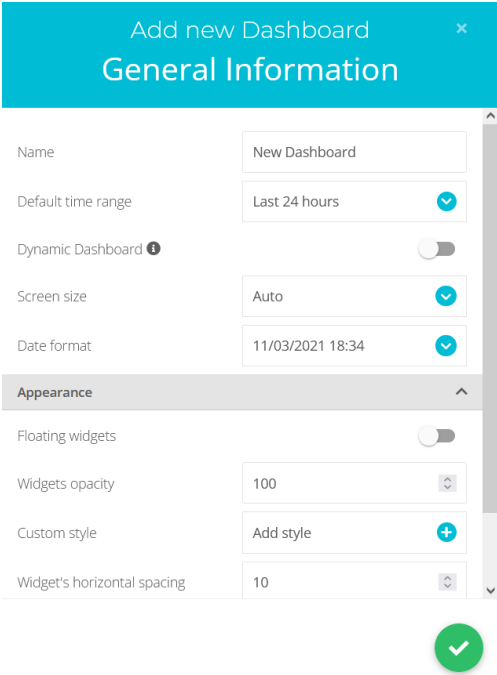
Dispositivo asociado a la tarjeta de desarrollo

05 Selección de dispositivo y creación de dashboard en Ubidots

- Para crear el dashboard: Data > Dashboards > Menú () >







Crear nuevo dashboard

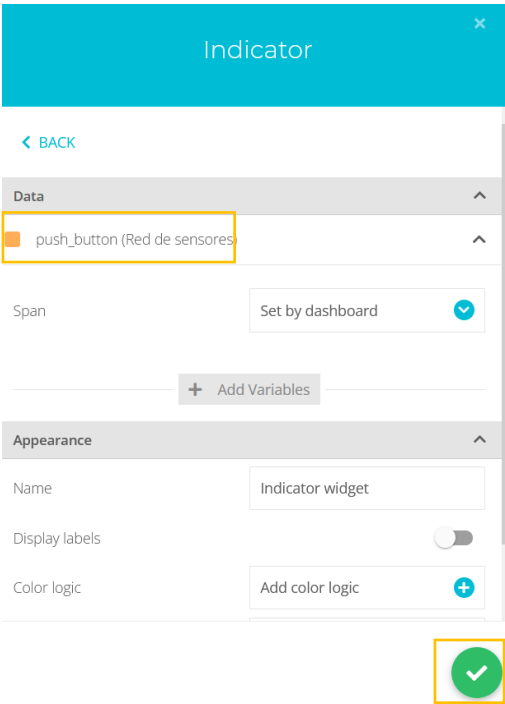
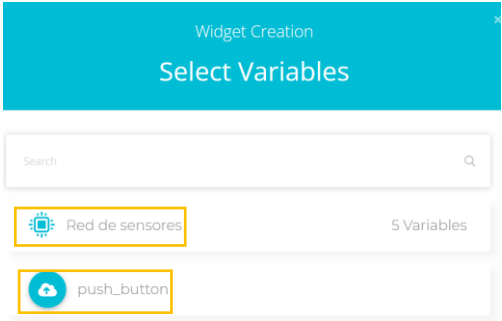
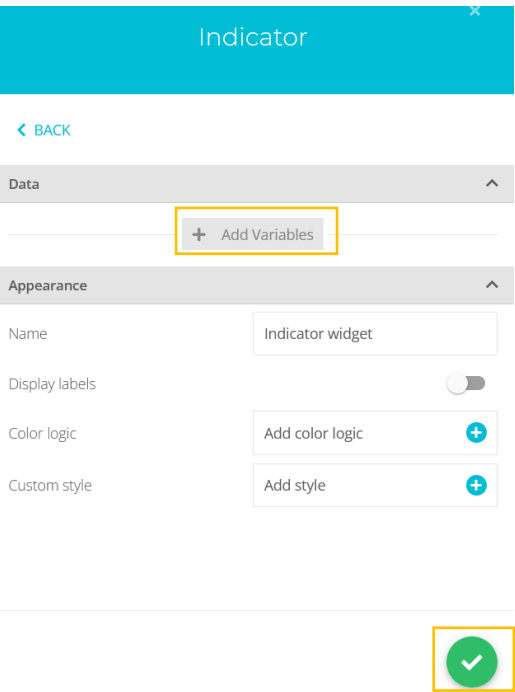
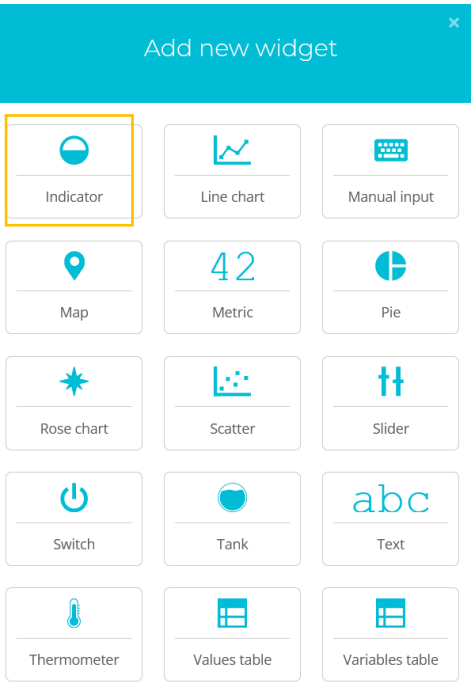
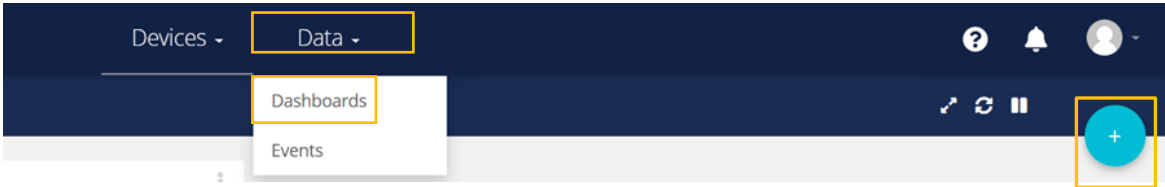
➔

Llenar los campos

Aparecen los dashboards ya creados

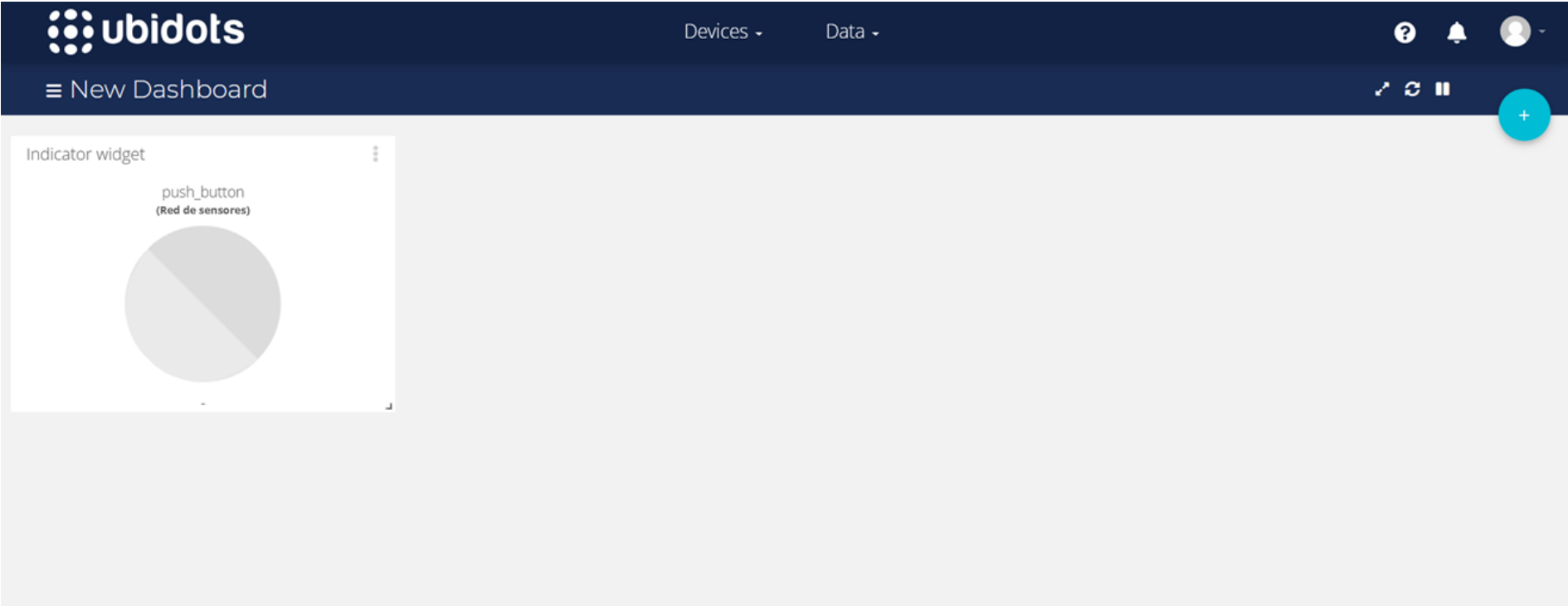
05 Selección de dispositivo y creación de dashboard en Ubidots

- Para crear ventanas de visualización o widgets: Data > Dashboards > +
- Seleccionar tipo de gráfico deseado > + Add Variables> seleccionar el dispositivo y la variable a agregar



05 Selección de dispositivo y creación de dashboard en Ubidots

- Dashboard



























05 Selección de dispositivo y creación de dashboard en Ubidots

- Para crear ventanas de visualización o widgets: Data > Dashboards >
- La selección de los widgets se realiza de modo que el gráfico sea lo más representativo posible de la variable

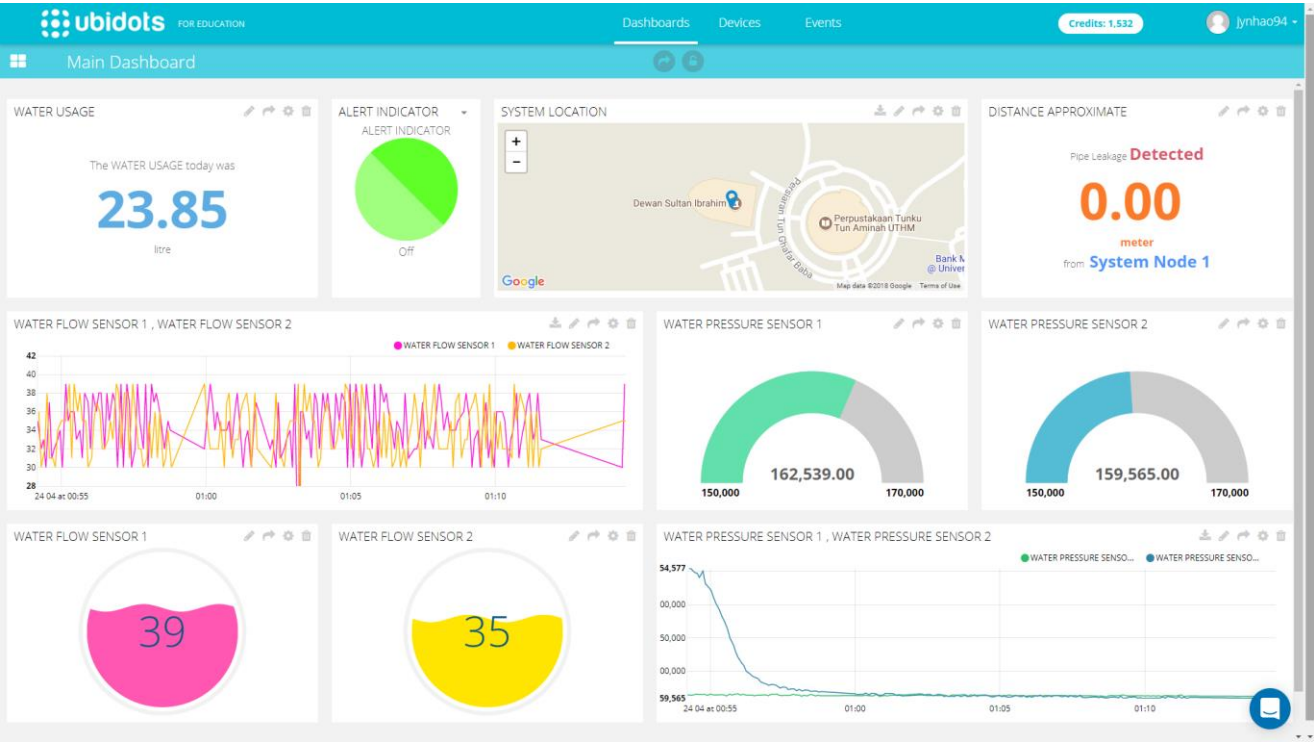


Add new widget

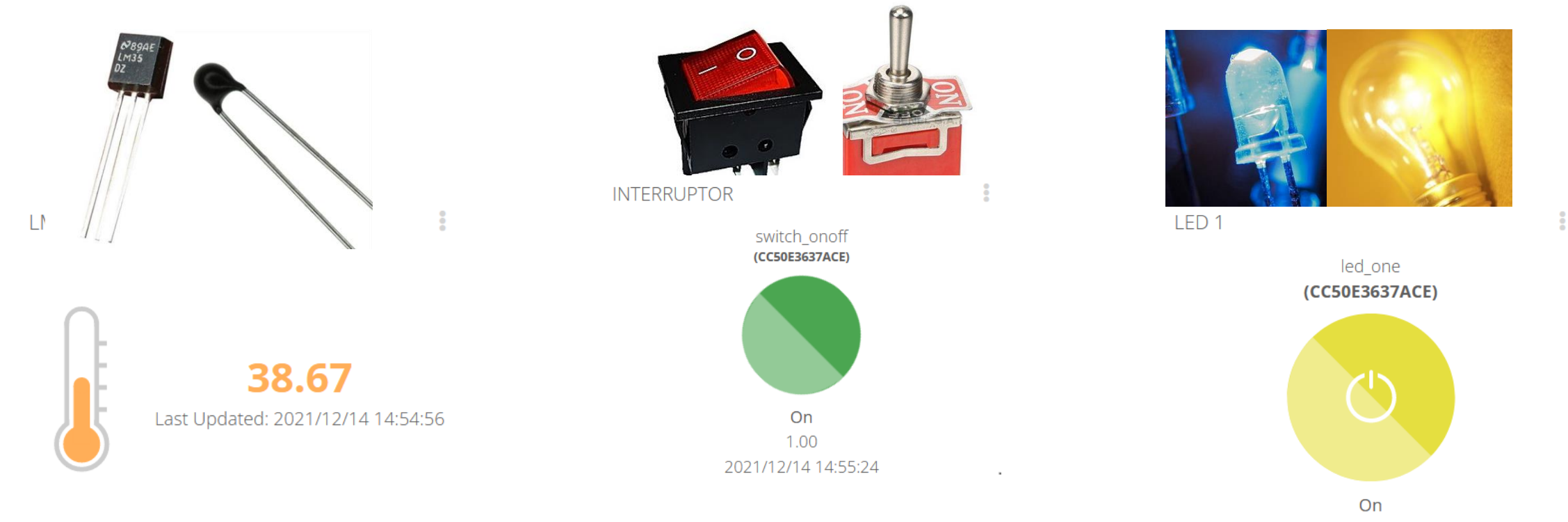
 Bar chart	 Battery	 Clock	 Map	 42 Metric	 Pie
 Devices Table	 Double Axis	 Gauge	 Rose chart	 Scatter	 Slider
 HTML Canvas	 Histogram	 Image	 Switch	 Tank	 abc Text
 Indicator	 Line chart	 Manual input	 Thermometer	 Values table	 Variables table

05 Selección de dispositivo y creación de dashboard en Ubidots

- Para crear ventanas de visualización o widgets: Data > Dashboards > +
- La selección de los widgets se realiza de modo que el gráfico sea lo más representativo posible de la variable



05 Selección de dispositivo y creación de dashboard en Ubidots



4. Lectura de sensores (ejemplo de aplicación)

Código

```

/*****
  Include Libraries
  *****/
#include "Ubidots.h"
/*****
  Define Instances and Constants
  *****/
#define LED_BUILTIN D0

int switch_onoff = D2; // Interruptor
int push_button = D1; // Pulsador
int led_y = D6; //Led amarillo (LED 1)
int led_r = D5; //Led rojo (LED 2)

int led_one;
int led_two;

double temperat;
double Vcd;
double V;

//Voltaje de alimentación puente de Wheatstone
double Vab = 3.1; //Voltaje de entrada

//Constantes ecuación R vs T del termistor
double A = 321234.942071026000000000;
double B = 0.04638645038113310; //17 cifras decimales

//Resistencias puente de Wheatstone
double R1 = 101400; //Ohmios Brazo Izq sup
double R2 = 47080; //Ohmios Brazo Izq inf
double R3 = 75260; //524740; //Ohmios Brazo Der sup

//Cálculo ganancia amplificador
double Rg = 98900; //Ohmios
double G; //Ganancia AD620
double Rt;

const char* UBIDOTS_TOKEN = "BBFF-8RkyJSM7QpyAea461ytLSGWQdz82c5";
const char* WIFI_SSID = "Aleja G"; //"CLARO_WIFIA95"// Put here your Wi-Fi SSID (Name)
const char* WIFI_PASS = "pechilomas"; //"CLAROI743"// Put here your Wi-Fi password

Ubidots ubidots(UBIDOTS_TOKEN, UBI_TCP);

void setup() {

  Serial.begin(115200);
  ubidots.wifiConnect(WIFI_SSID, WIFI_PASS);
  //ubidots.setDebug(true); // Uncomment this line for printing debug message

  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(switch_onoff, INPUT);
  pinMode(push_button, INPUT);
  pinMode(led_y, OUTPUT);
  pinMode(led_r, OUTPUT);

  led_one = 0;
  led_two = 0;

}

```

4. Lectura de sensores (ejemplo de aplicación)

Código – Lectura de variables

```
void loop() {
  /*****
  Lectura sensor de temperatura
  *****/
  V = analogRead(A0); // Lee entrada A1 y la asigna a variable Vcd
  Vcd = V * (3.1) / 1024;
  G = 1 + (49400 / Rg); // Cálculo de ganancia
  //Serial.println(V, 4);
  //Serial.println(Vcd, 4);
  Vcd = Vcd / G;
  Rt = ((R1 * R3 * Vab) - (Vcd * R1 * R3) - (Vcd * R2 * R3)) / (Vab * R2 + Vcd * R2 + Vcd * R1); //Considerando Vd - Vc
  temperat = (log(A) - log(Rt)) / B;
  Serial.print("Temperatura: ");
  Serial.println(temperat);

  /*****
  Lectura de datos enviados desde Ubidots
  *****/
  led_y = ubidots.get("c45bbe6d51d1", "led_one");
  if (led_y != ERROR_VALUE) {
    Serial.print("LED 1: ");
    Serial.println(led_y);
    digitalWrite(D6, led_y);
  }
  else {
    Serial.println("Error en lectura del LED 1");
  }
  led_r = ubidots.get("c45bbe6d51d1", "led_two");
  if (led_r != ERROR_VALUE) {
    Serial.print("LED 2: ");
    Serial.println(led_r);
    digitalWrite(D5, led_r);
  }
  else {
    Serial.println("Error en lectura del LED 2");
  }
}
```

```

/*****
Lectura sensores alto - bajo
*****/

push_button = digitalRead(D1); // Lectura pulsador 1
if (push_button == LOW)
{
  push_button = 0;    //
}
else if (push_button == HIGH)
{
  push_button = 1;    //
}
Serial.print("Pulsador: ");
Serial.println(push_button);

switch_onoff = digitalRead(D2); //Lectura swiche on - off
if (switch_onoff == LOW)
{
  switch_onoff = 0;    //
}
else if (switch_onoff == HIGH)
{
  switch_onoff = 1;    //
}
Serial.print("Interruptor: ");
Serial.println(switch_onoff);

```

4. Lectura de sensores (ejemplo de aplicación)

Código – Envío de datos a Ubidots

```
ubidots.add("Temperat", temperat); // Change for your variable name
ubidots.add("push_button", push_button); // Change for your variable name
ubidots.add("switch_onoff", switch_onoff); // Change for your variable name
bool bufferSent = false;
bufferSent = ubidots.send(); // Will send data to a device label that matches the device Id

if (bufferSent) //Do something if values were sent properly
{
    Serial.println("Values sent by the device");
}

}
```

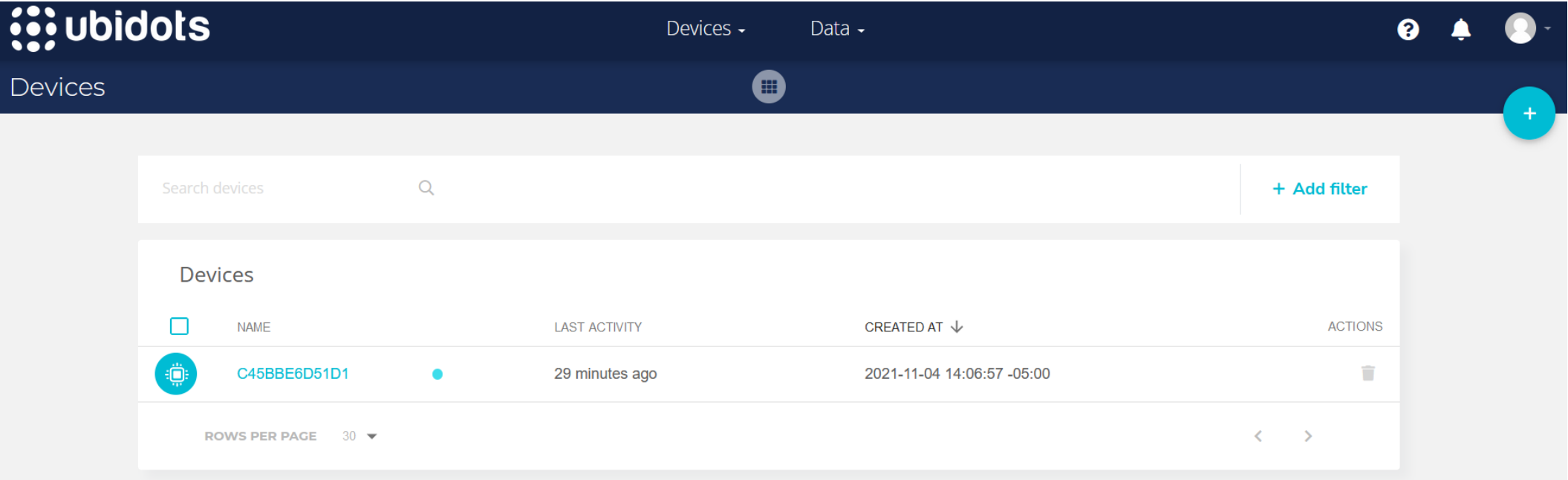
4. Lectura de sensores (ejemplo de aplicación)

Visualización del monitor serial



4. Lectura de sensores (ejemplo de aplicación)

Interfaz – Visualización del dispositivo



4. Lectura de sensores (ejemplo de aplicación)

Interfaz – Visualización de Variables y dispositivo

Nombre del dispositivo
➔

C45BBE6D51D1

Description

Change description

API Label ⓘ

c45bbe6d51d1

ID ⓘ

61842f511d84721e8137c544

Token

.....

Tags

Add new tag

Last activity

30 minutes ago

Device type ⓘ

Set Device Type

Location ⓘ

Mode ⓘ

Auto

1.00

led_one

Last activity:

36 minutes ago

1.00

led_two

Last activity:

36 minutes ago

0.00

push_button

Last activity:

30 minutes ago

1.00

switch_onoff

Last activity:

30 minutes ago

15.00

temperat

Last activity:

30 minutes ago

+
Add Variable

Variable ID

61842f521d84721e8137c546

API Label

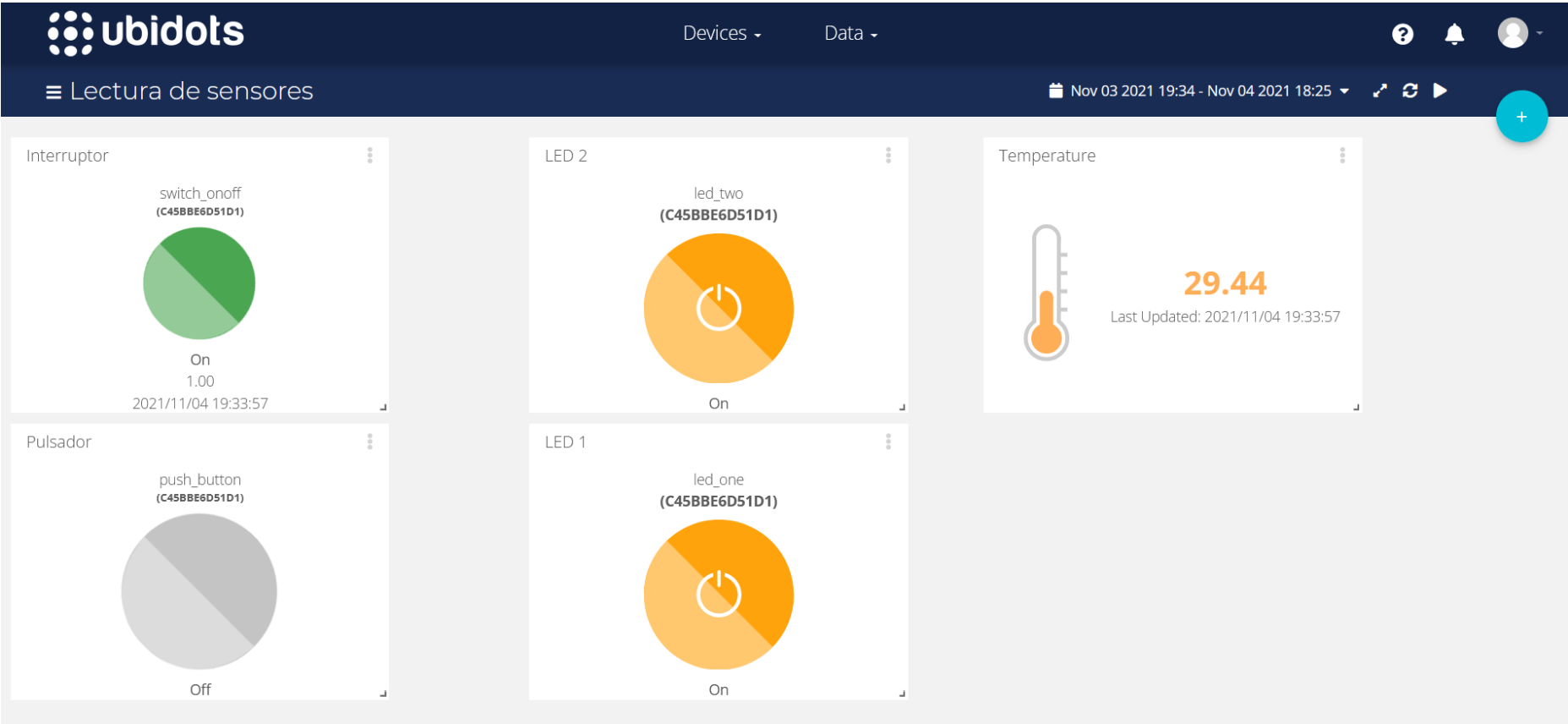
temperat

VARIABLES PER PAGE 30

Nombre de la variable
←

4. Lectura de sensores (ejemplo de aplicación)

Interfaz – Visualización del Dashboard



4. Lectura de sensores (ejemplo de aplicación)

Interfaz - Alertas

ubidots

DevicesData




























?

Events

Filter by: All organizations

Sort by: Created date

Search event

	Llamada por alerta de temperatura				    
	Alerta Interruptor				    
	Alerta de temperatura				    

MUCHAS GRACIAS