

Guía IoT Diseño y creación de Interfaces 2022

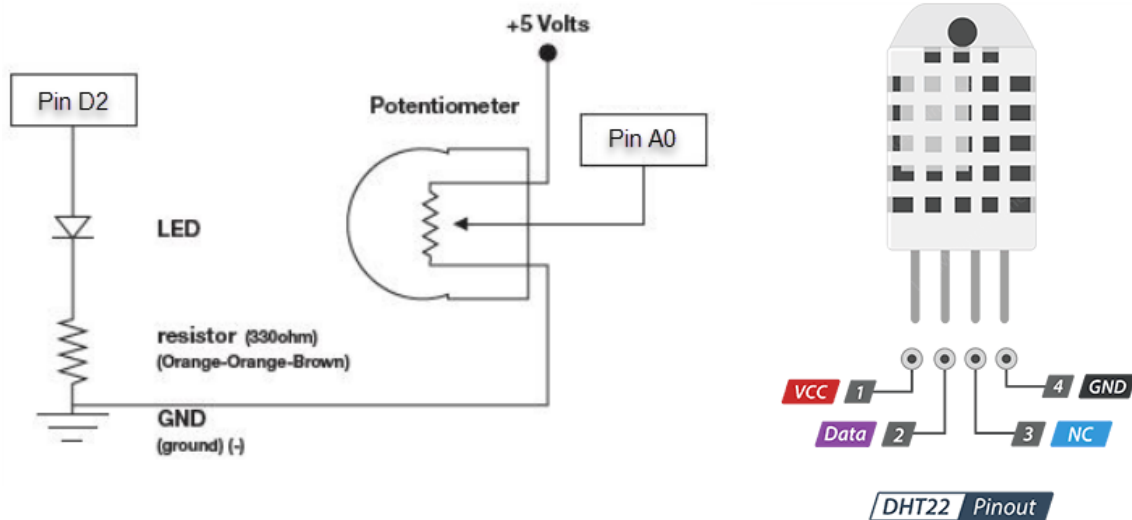
Preparado por: **Camilo Becerra** y **Paul Aguayo**

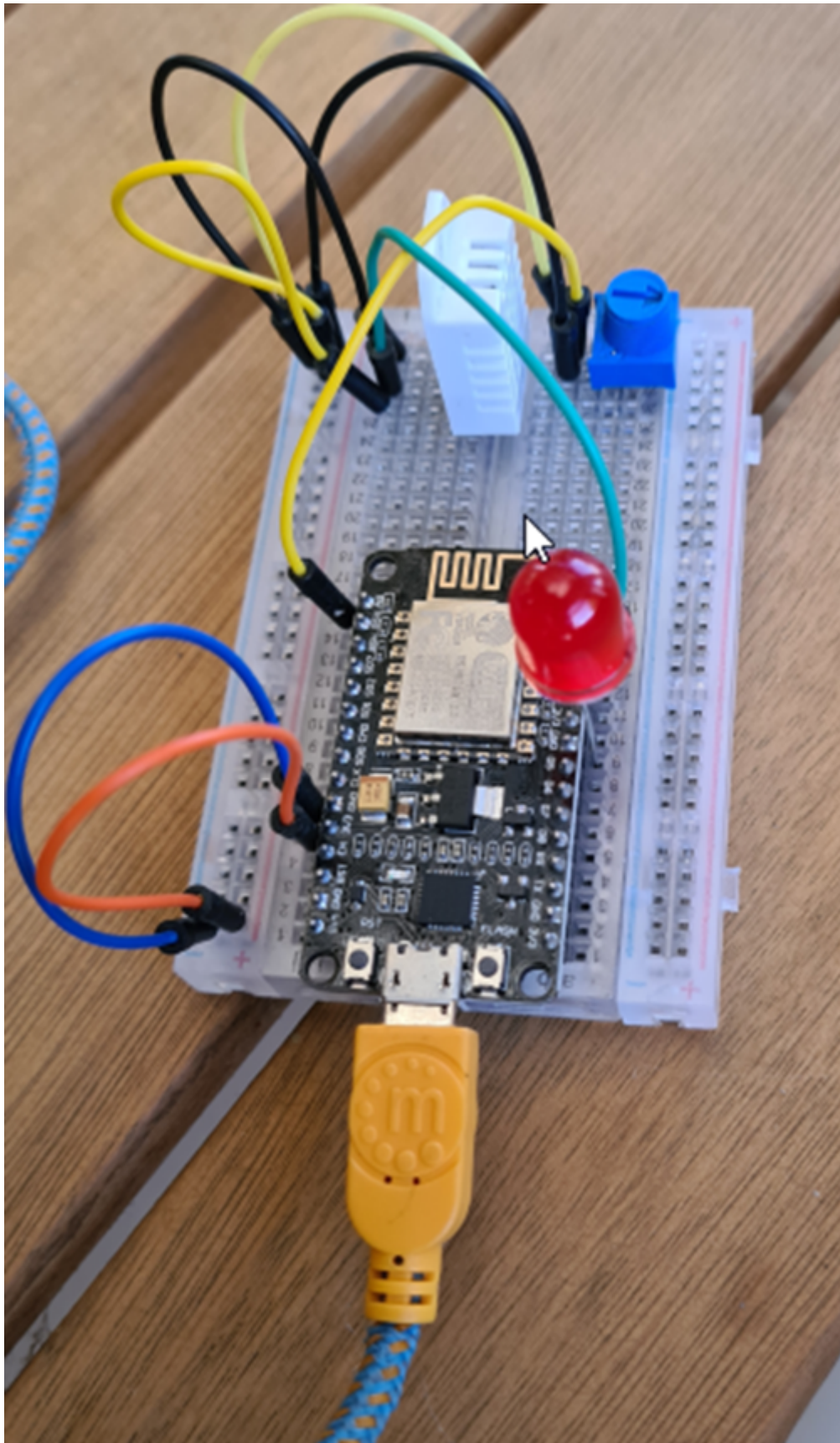
paguayo@edu.uai.cl

La siguiente guía nos muestra cómo utilizar la NodeMCU para leer datos análogos o digitales y enviarlos a la plataforma cloud de Arduino, la cual nos permitirá crear dashboards para monitorear las variables que estamos sensando permitiéndonos acceder desde cualquier parte del mundo a la información que recolectemos. Adicionalmente desde la misma plataforma podremos generar en forma automática el código para nuestra tarjeta y programarla.

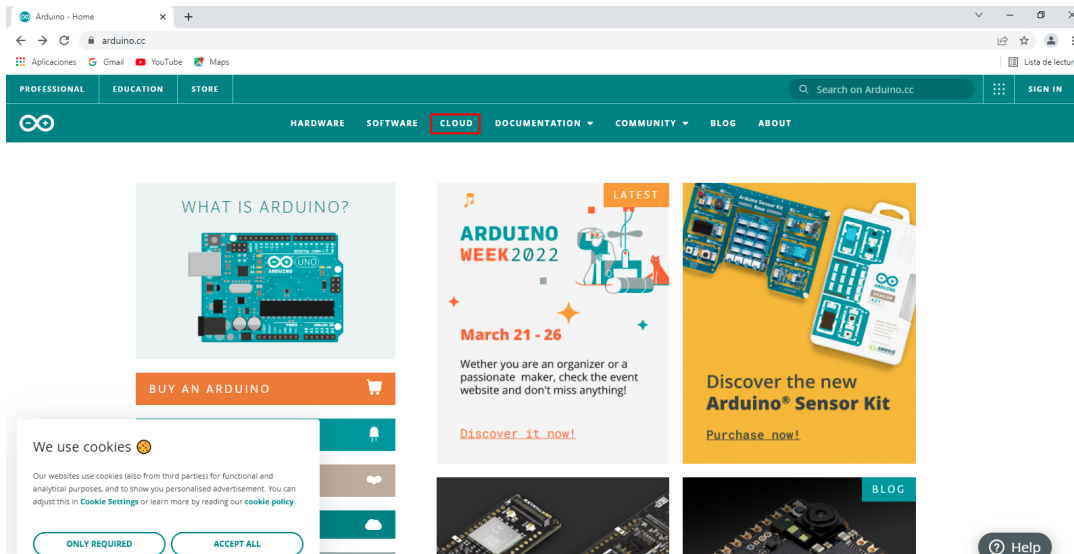
NodeMCU es una tarjeta de desarrollo basada en el módulo ESP8266 la cual cuenta con conectividad WiFi y además es programable con el IDE de Arduino.

Para poder demostrar el uso de la plataforma, crearemos un circuito simple que nos permita leer una entrada analógica (potenciómetro A0) y también controlar una salida digital (LED pin D2). Adicionalmente conectaremos un sensor de Temperatura y Humedad DHT22 (D1).

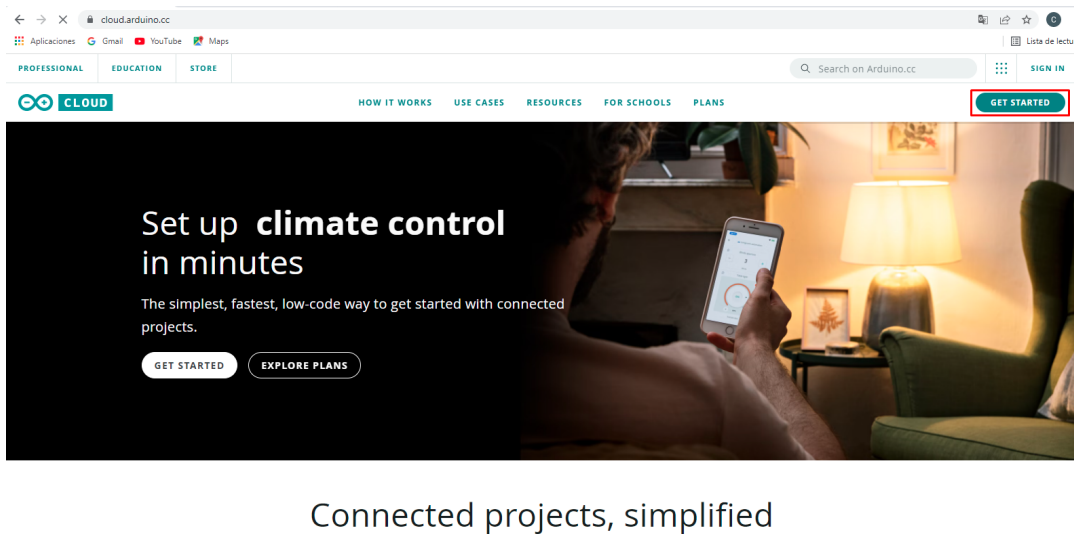




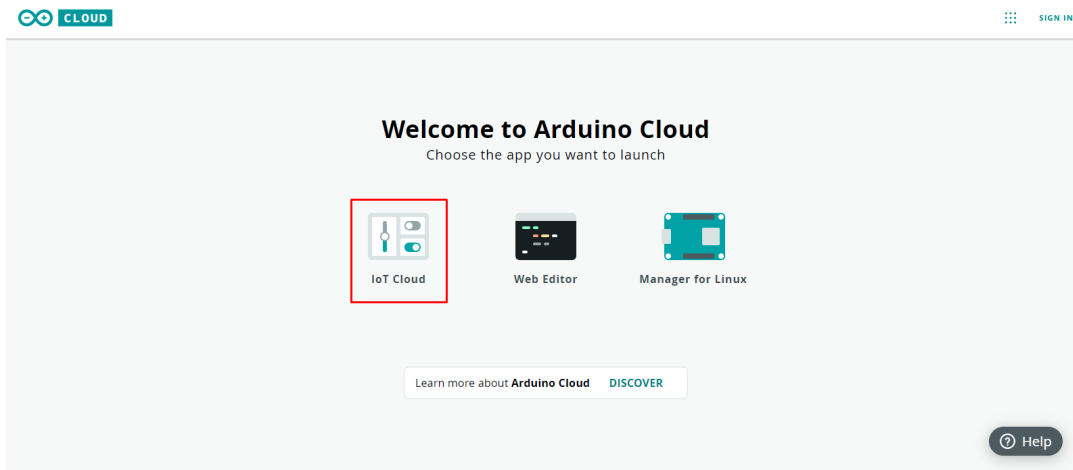
Una vez tengamos armado nuestro circuito ingresamos al sitio web arduino.cc y seleccionamos **CLOUD**



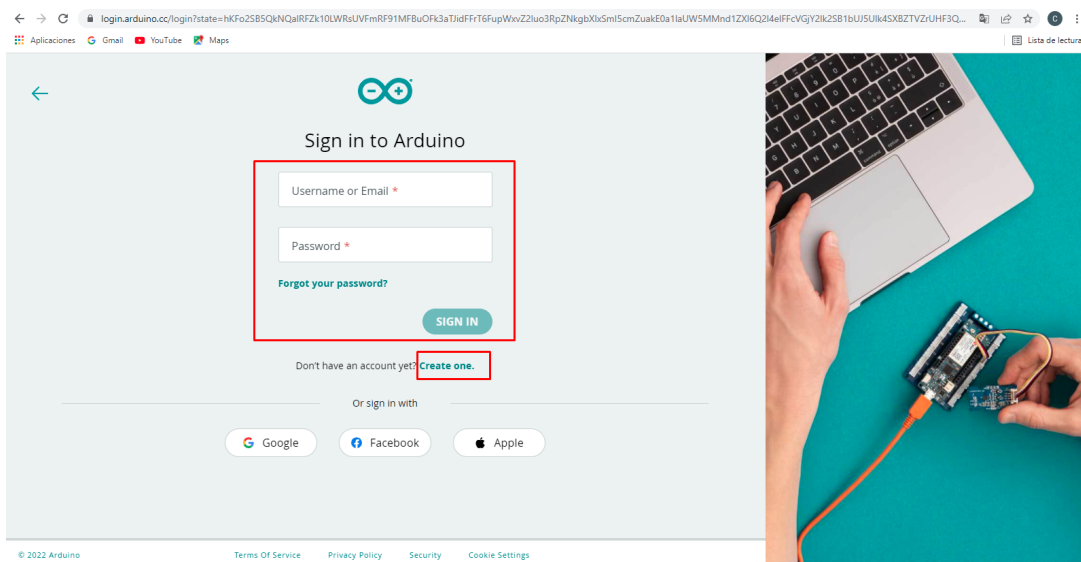
en la página del cloud debemos seleccionar **“GET STARTED”** en la esquina superior derecha



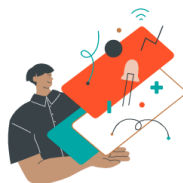
Veremos 3 opciones. En esta guía utilizaremos dos de ellas, el IoT Cloud y el Web Editor. Comenzaremos escogiendo la opción **“IoT cloud”**



Si ya tienes cuenta en arduino.cc puedes usarla para ingresar, de lo contrario debes crear una; es gratuita. Al crear una cuenta se solicita la fecha de nacimiento, esto es porque las cuentas de menores de edad no registran datos manteniendo su privacidad.



Una vez creada la cuenta podemos crear y configurar nuestros dispositivos en la plataforma. La versión gratuita permite crear 2 “things”, que corresponden a los dispositivos IoT o tarjetas de control. Cada una puede tener conectada uno o más sensores o actuadores. Para la versión gratuita el máximo de sensores es de 5 por dispositivo. Para obtener más información sobre los planes disponibles puedes visitar <https://cloud.arduino.cc/plans>



Create your first Thing

A Thing is a connected device that can communicate with the cloud. You can make your Things interact with other Things or anything else in the physical world.

CREATE THING

A continuación, crearemos y asignaremos la información de nuestro dispositivo, para ello haz click en **“CREATE THING”**

Asignamos un nombre cambiando **“Untitled”** por el que nosotros queramos. Cada lectura ya sea digital o análoga que queramos enviar a la nube debemos asociarla con una variable para que la plataforma pueda identificarla, para hacer esto debemos presionar **“ADD VARIABLE”**

Untitled Thing ID: 9c072d6d-00b3-4dad-855a-8c3e76b1370c

Setup Sketch

Variables

Variables are what you can monitor or control to make your Thing function. For example a temperature or a smart lamp. Once created, you can use them in your Sketch.

ADD VARIABLE

Device

Select the device you want to use or configure a new one.

SELECT DEVICE

Select Device

Network


Enter your network credentials to

Set webhook Timezone: America/New York

Para este ejemplo comenzaremos con el sensor de temperatura y humedad relativa, DHT22. El nombre de la variable lo escogemos nosotros, en este caso será **dht_temperatura** y la definiremos de tipo flotante (float) ya que de esta forma nos permitirá el uso decimales en la medición. Configuraremos el sensor como solo lectura y le pediremos que nos envíe un mensaje cuando se produzca un cambio en la información del mismo. La magnitud del cambio que queramos detectar es configurable por medio del parámetro **“Threshold”**. La plataforma permite también el envío de datos en una periodicidad establecida.

Add variable ×

Name
dht_temperatura

 Sync with other Things i

Floating Point Number eg. 1.55

Declaration
`float dht_temperatura ;` i

Variable Permission i

Read & Write Read Only

Variable Update Policy i

On change Periodically

Threshold

Presiona **“ADD VARIABLE”** para crear la variable

Variable Permission i

Read & Write Read Only

Variable Update Policy i

On change Periodically


Threshold
0

CANCEL

Realizamos el mismo procedimiento para crear variable **dht_humedad**

Add variable ×

Name
dht_humedad

 Sync with other Things ⓘ

Floating Point Number eg. 1.55 ▼

Declaration
`float dht_humedad ;` ⓘ

Variable Permission ⓘ

Read & Write Read Only

Variable Update Policy ⓘ

On change Periodically

Variable Permission ⓘ

Read & Write Read Only

Variable Update Policy ⓘ

On change Periodically

Threshold
0

ADD VARIABLE CANCEL

Creamos la variable con **“ADD VARIABLE”**

Ahora creamos una nueva variable para una lectura analógica, en este caso será de tipo entero (int) y solo lectura

Add variable ✕

Name
lectura_analogica

↻ Sync with other Things i

Integer Number eg. 1

Declaration
`int lectura_analogica;`

Variable Permission i

Read & Write Read Only

Variable Update Policy i

On change Periodically

Variable Permission i

Read & Write Read Only

Variable Update Policy i

On change Periodically

Threshold
0

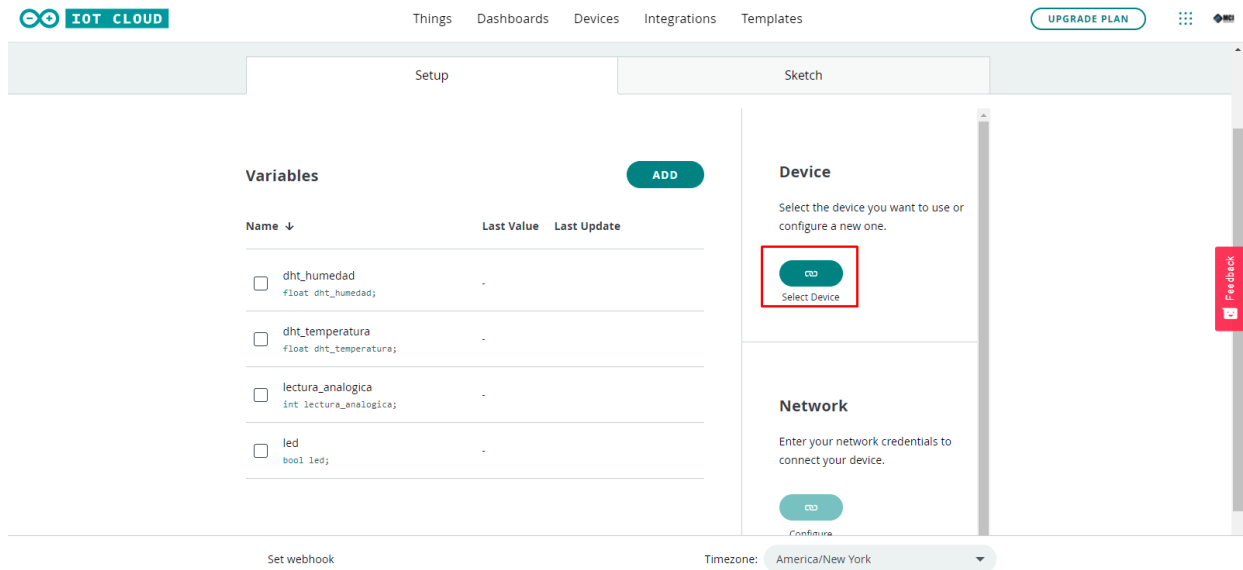
ADD VARIABLE

CANCEL

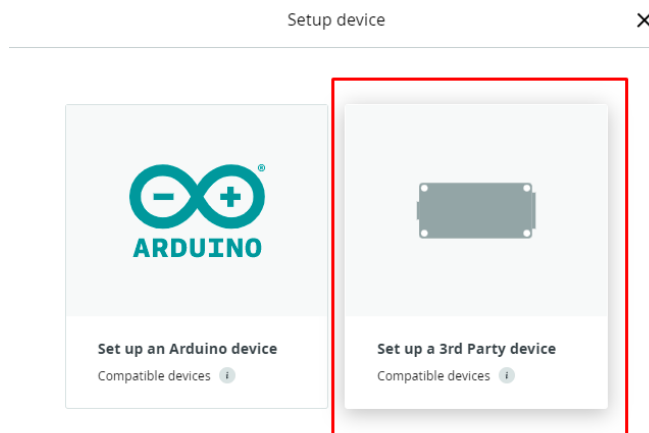
Creamos la variable con **“ADD VARIABLE”**

La variable led se crea de la misma manera, pero en este caso será de tipo **“bool”** y de lectura y escritura puesto que queremos poder consultar el estado del LED además de encenderlo y apagarlo en forma remota.

Con las variables creadas seleccionamos el dispositivo (NodeMCU) que vamos a utilizar.



En este caso seleccionaremos las tarjetas compatibles. Si tuviéramos alguna tarjeta arduino original como con WiFi como la MKR1000, MKR1010, Nano33 IoT, etc deberíamos seleccionar la opción de la izquierda “Set up an Arduino device”



Seleccionamos el tipo de tarjeta y el modelo. Para el NodeMCU escogemos ES8266

← Setup device ×

Select device type

Please select the device type and model you want to configure


ESP8266 ESP32 LoRaWAN

NodeMCU-32S

CONTINUE

Elegimos un nombre para nuestro dispositivo

← Setup device ×



Give your device a name

Name your device so you will be able to recognize it.

Device Name
Esp32

NEXT

Tras presionar “NEXT” obtendremos el mensaje de dispositivo listo. **Es importante descargar el pdf que allí aparece porque tendremos una clave para utilizar el dispositivo de forma segura.** Esta clave no es recuperable por lo que debes mantenerla en un lugar seguro. Si la pierdes la única opción es volver a crear el dispositivo.

Setup device ×

Make your device IoT-ready

To use this board you will need a Device ID and a Secret Key, please copy and save them or [download the PDF.](#)

Also, keep in mind that this device authentication has a lower security level compared to other Arduino devices.

Device ID
08d599e1-b4e4-47aa-a542-c959c05a0c1e

Secret Key
4EWSTYWUYR0QLWEB8Y5X

⚠ Secret key cannot be recovered
Please keep it safe, if you lose it you will have to delete and setup your device again.

I saved my device ID and Secret Key CONTINUE

El pdf que descargues va a contener la información que se muestra a continuación. La “Secret Key” es la clave de la que se hizo mención anteriormente. Estos datos nos permitirán identificar en forma única nuestro dispositivo.



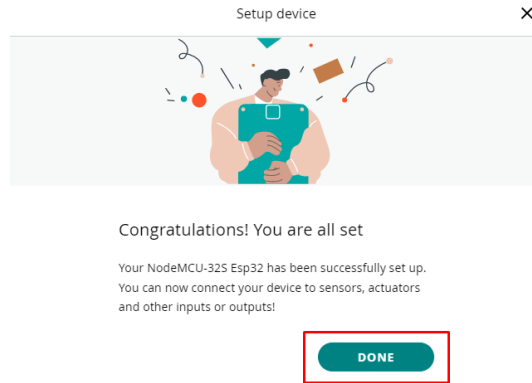
Esp32

ESP32
NodeMCU-32S Configured on March 9, 2022

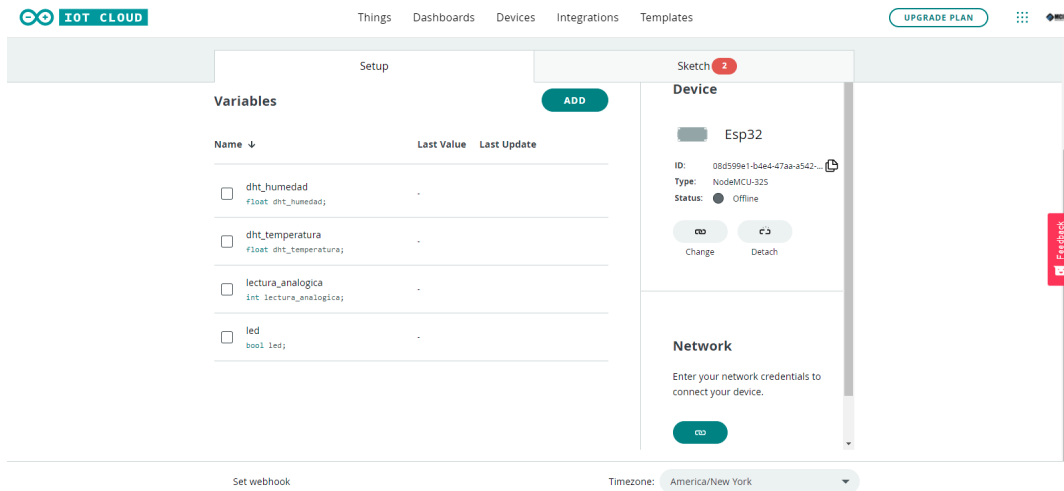
Device ID
08d599e1-b4e4-47aa-a542-c959c05a0c1e

Secret Key
4EWSTYWUYR0QLWEB8Y5X

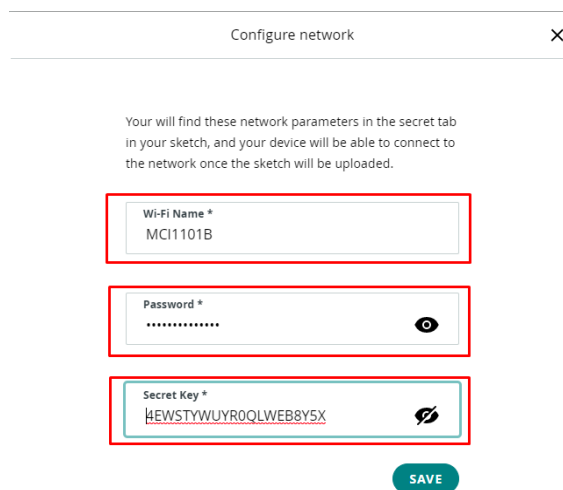
Terminado este paso presionamos "DONE"



Lo siguiente es configurar la red WiFi en nuestro dispositivo para eso vamos a **Network**, presionamos "configure"



Agregamos las credenciales de red, nombre de red (SSID), password y serial key del pdf que descargamos



Así se vería la pantalla con todo lo que se ha realizado hasta ahora

The screenshot shows the Arduino IoT Cloud interface. At the top, there are navigation tabs: Things, Dashboards, Devices, Integrations, and Templates. A 'UPGRADE PLAN' button is visible in the top right. The main content area is divided into two tabs: 'Setup' and 'Sketch'. The 'Setup' tab is active, showing a table of variables:

Name ↓	Last Value	Last Update
<input type="checkbox"/> dht_humedad float dht_humedad;	-	-
<input type="checkbox"/> dht_temperatura float dht_temperatura;	-	-
<input type="checkbox"/> lectura_analogica int lectura_analogica;	-	-
<input type="checkbox"/> led bool led;	-	-

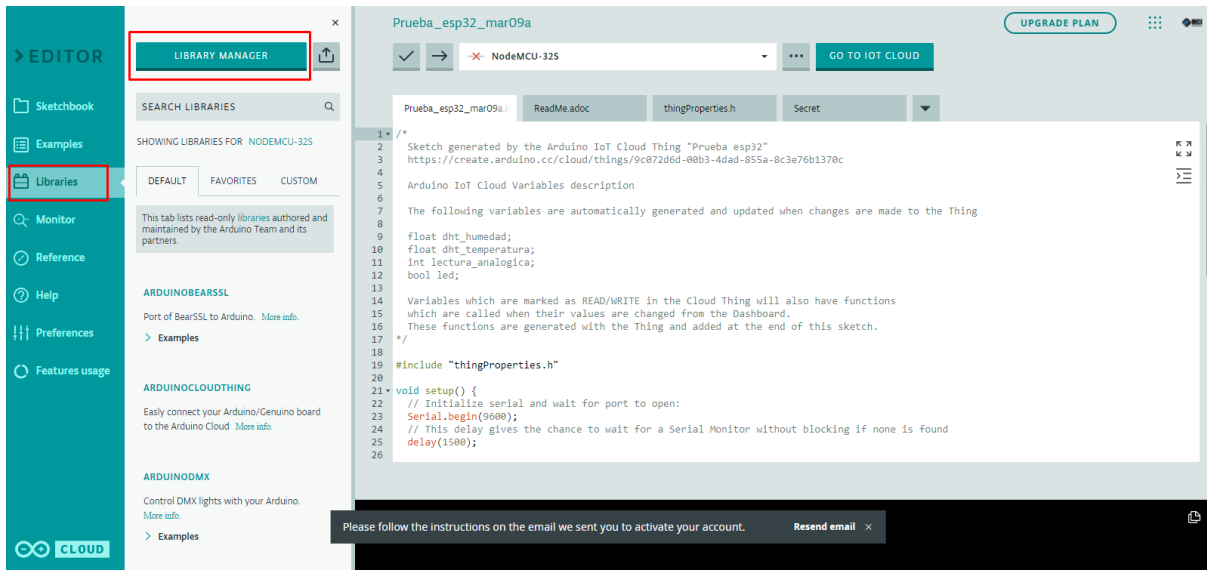
An 'ADD' button is located above the table. To the right, the 'Sketch' tab is partially visible, showing device information for 'Esp32': ID: 08d599e1-b4e4-47aa-a542-..., Type: NodeMCU-32S, Status: Offline. There are 'Change' and 'Detach' buttons. Below that, network settings are shown: Wi-Fi Name: MC110..., Password:, Secret Key:

Para continuar, seleccionamos Sketch y Open full editor

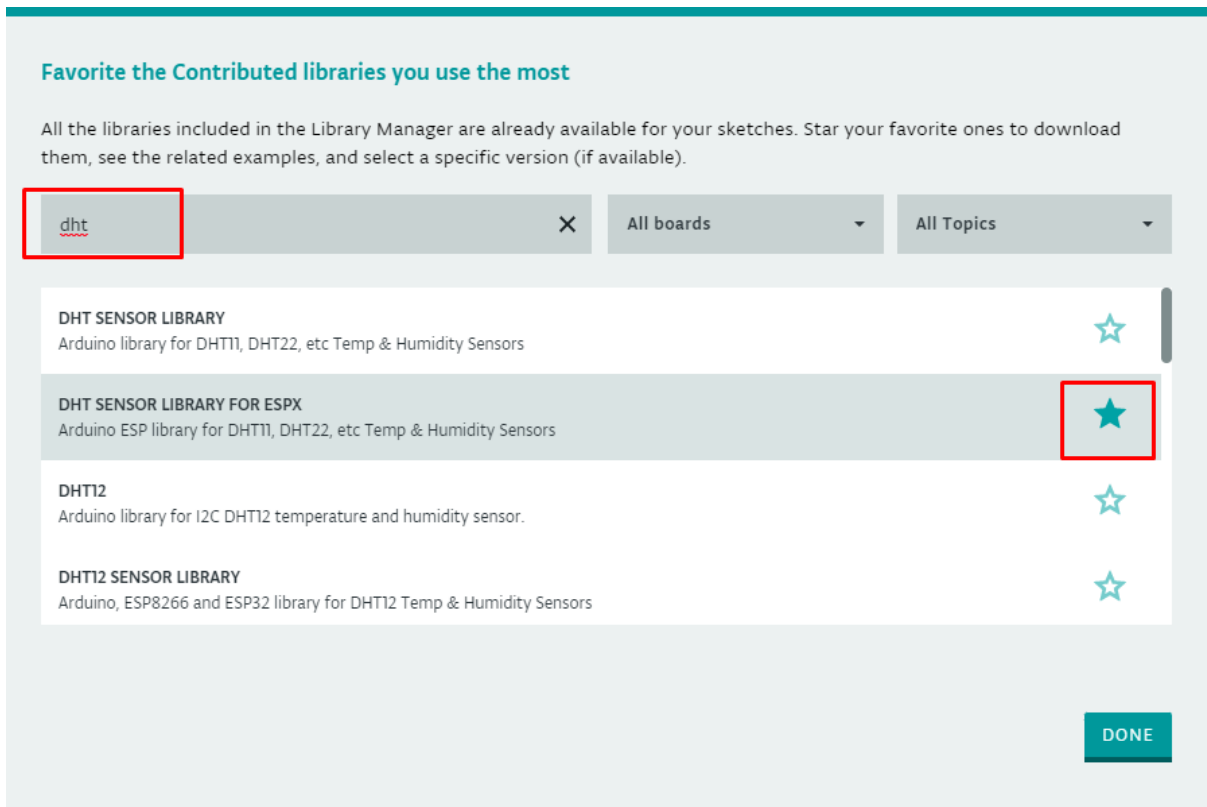
The screenshot shows the Arduino IoT Cloud interface with the 'Sketch' tab selected. The 'Open full editor' button is highlighted with a red box. Below the editor, a code snippet is visible:

```
1 /*
2 Sketch generated by the Arduino IoT Cloud Thing "Prueba esp32"
3 https://create.arduino.cc/cloud/things/9c072d6d-00b3-4dad-855a-8c3e76b1370c
4
5 Arduino IoT Cloud Variables description
6
7 The following variables are automatically generated and updated when changes are made to the Thing
8
9 float dht_humedad;
10 float dht_temperatura;
11 int lectura_analogica;
12 bool led;
13
```

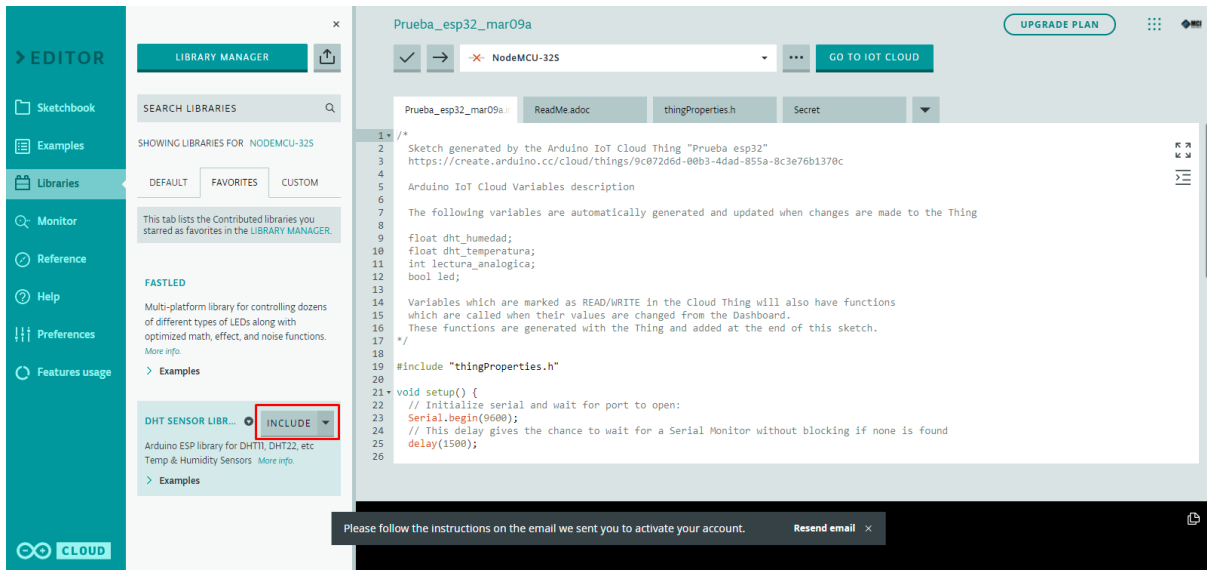
El siguiente paso es necesario para poder utilizar el sensor DHT22 puesto que necesita una librería especial. Si no lo vas a utilizar, puedes omitir este paso. En el editor online, seleccionar “Libraries”, luego “LIBRARY MANAGER”



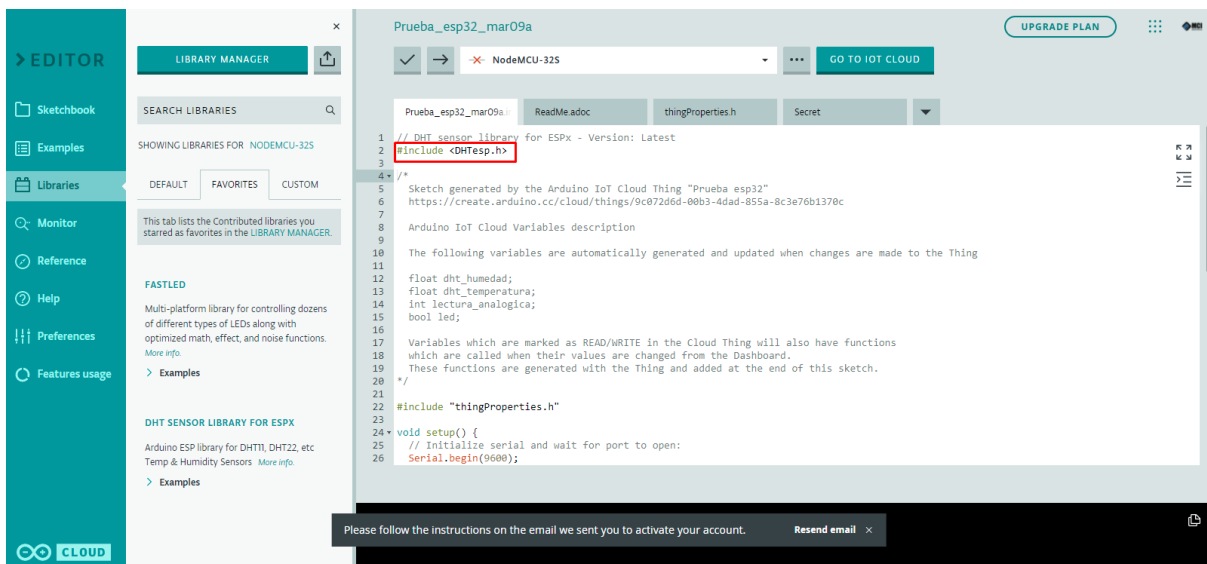
Buscar dht y seleccionar estrella en “DHT SENSOR LIBRARY FOR ESPX”, luego done



De vuelta en la ventana del editor online, presionar “INCLUDE” en el recuadro de la librería



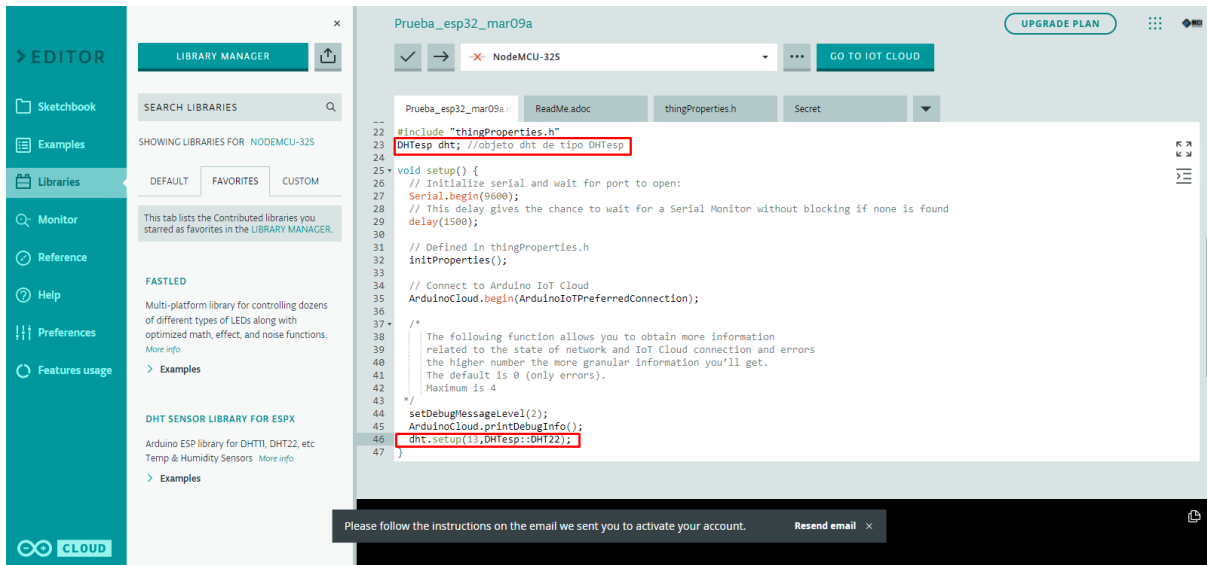
De esta forma, la librería será incluida en el código que vamos a cargar a la tarjeta.



Creamos el objeto dht de tipo **DHTesp** antes del void setup () como se muestra en la siguiente imagen y agregamos la siguiente línea dentro del setup para configurar la librería.

`dht.setup(pin, DHTesp::DHT22);`

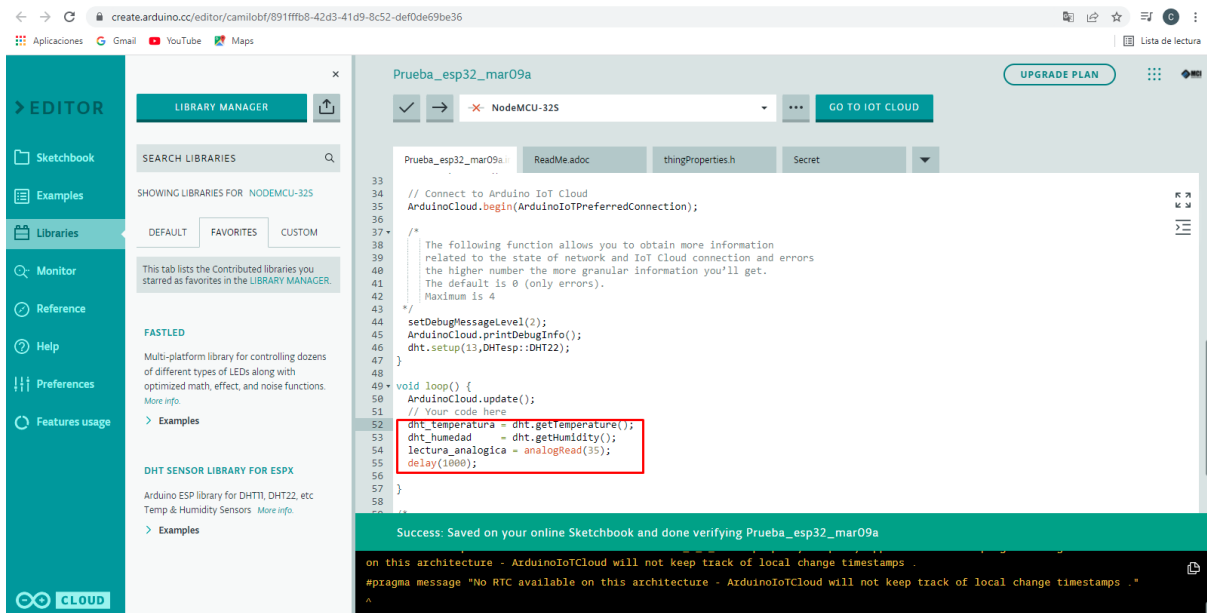
donde **pin** corresponde a pin donde está físicamente conectado el sensor DHT22.



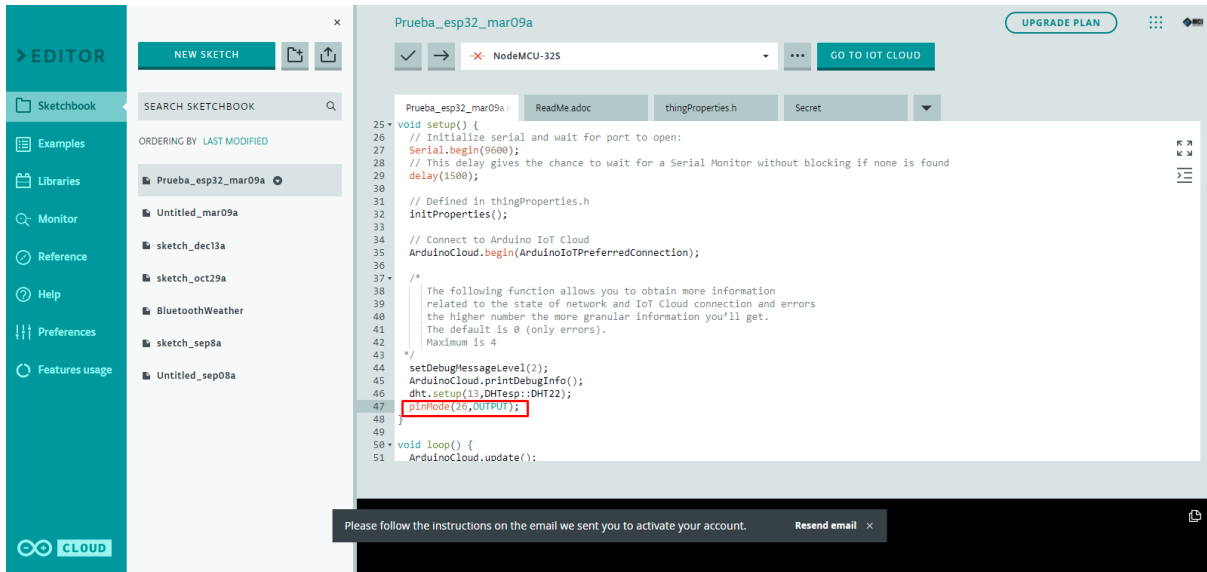
En el loop asignamos la lectura de datos a las variables que creamos previamente

```
dht_temperatura = dht.getTemperature();
dht_humedad = dht.getHumidity();
lectura_analógica = analogRead(35);
delay(1000);
```

Importante: debes verificar que los pines que estás utilizando en las funciones de lectura analógica y digital correspondan a la conexión física que tengas en la tarjeta NodeMCU



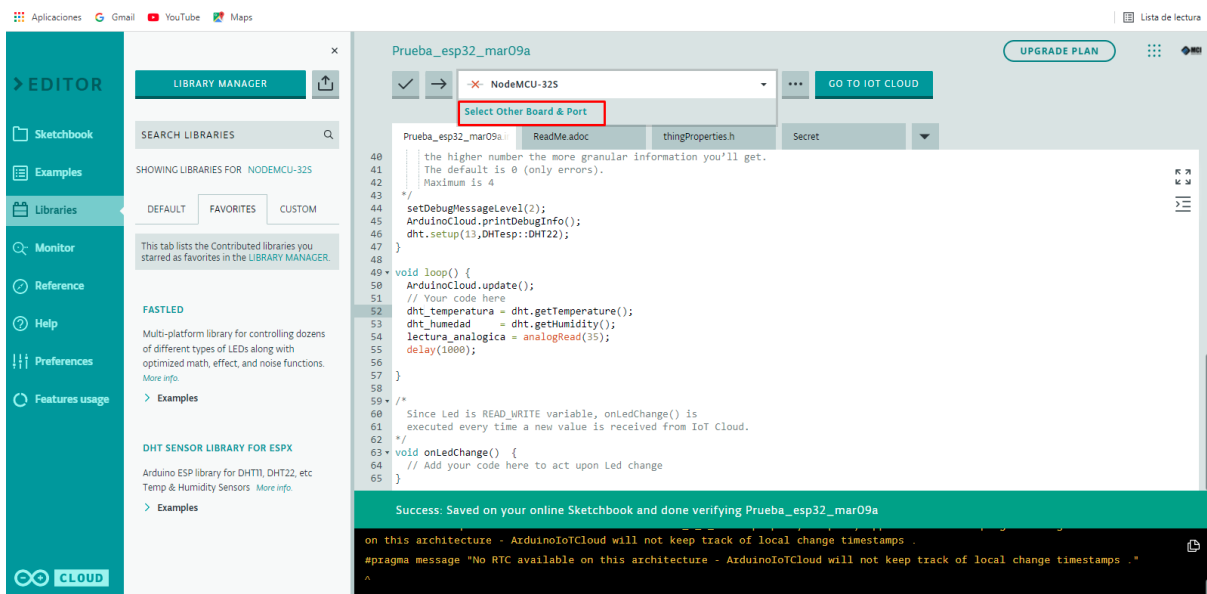
En setup, se debe definir pin donde tenemos conectado el LED como salida



Luego vamos a la función **void onLedChange()** y agregamos el código que queremos que se ejecute cuando cambie la variable Led. En nuestro caso

```
62   Since Led is READ_WRITE variable, onLedChange() is
63   executed every time a new value is received from IoT Cloud.
64   */
65   void onLedChange() {
66     // Add your code here to act upon Led change
67     if(led){
68       digitalWrite(4, HIGH);
69     }
70     else{
71       digitalWrite(4, LOW);
72     }
73   }
```

Finalmente seleccionamos la tarjeta y el puerto



Buscamos el modelo de la tarjeta que vamos a utilizar y el puerto al que está conectada. Para la NodeMCU se debe seleccionar como tarjeta “Generic ESP8266 Module” para cargar el programa

Select Other Board & Port

Select both a BOARD and a PORT if you want to upload a sketch.
If you only select a BOARD you will be able just to compile,
but not to upload your sketch.

BOARDS

nodem

- NodeMCU 0.9 (ESP-12 Module)
- NodeMCU 1.0 (ESP-12E Module)
- NodeMCU-32S ✓

PORTS

COM10

FLAVOURS

80MHz

921600

CANCEL OK

Con ello ya es posible cargar el programa a la tarjeta.

Finalmente nos queda configurar cómo se van a ver los datos que vamos a enviar desde nuestra tarjeta. Para eso vamos de vuelta a la página cloud, seleccionamos dashboard y creamos uno nuevo haciendo click en “**Build Dashboard**”

IoT CLOUD

Things **Dashboards** Devices Integrations Templates

UPGRADE PLAN

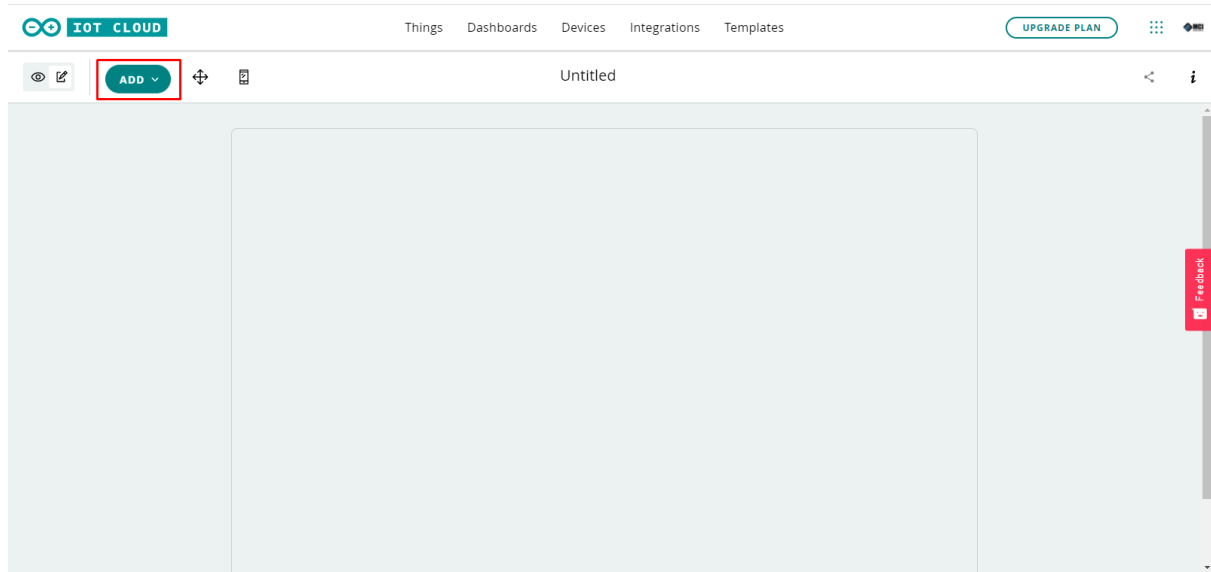
Monitor your Things

Build a Dashboard to easily monitor the status of your Things and control them.

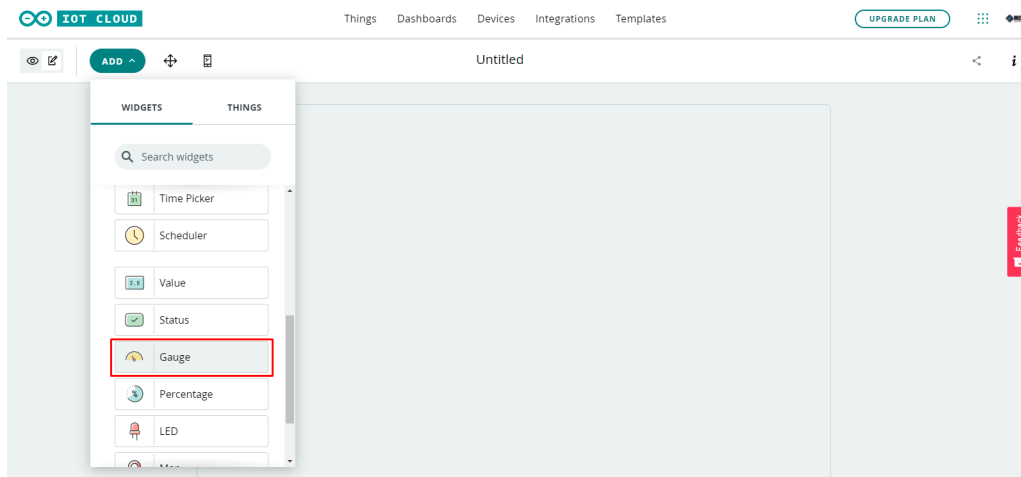
BUILD DASHBOARD

Feedback

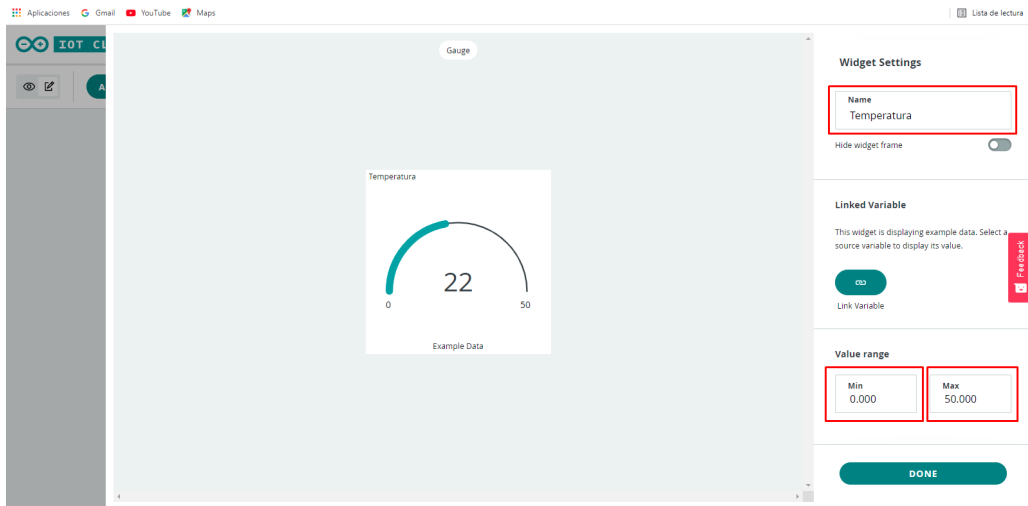
En el cloud agregaremos los widgets presionando “ADD”



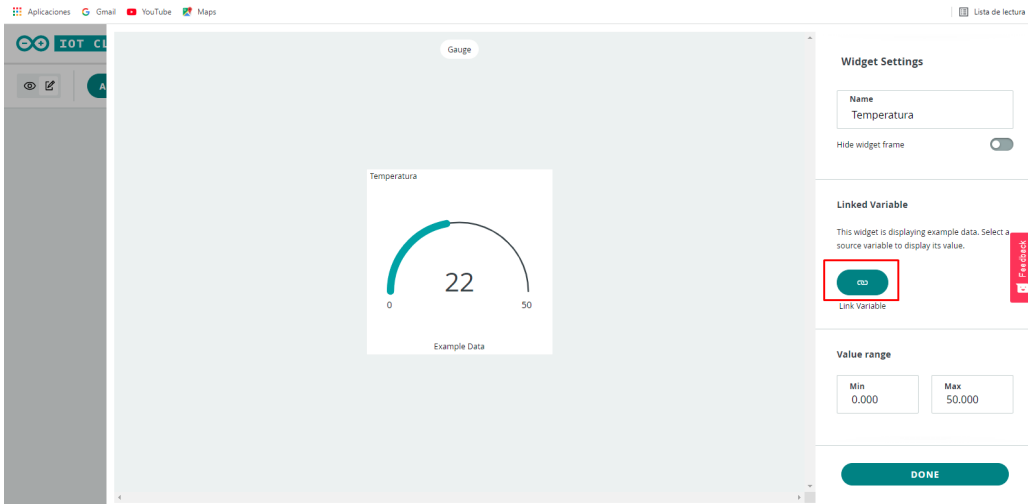
Para mostrar la temperatura seleccionaremos “Gauge”



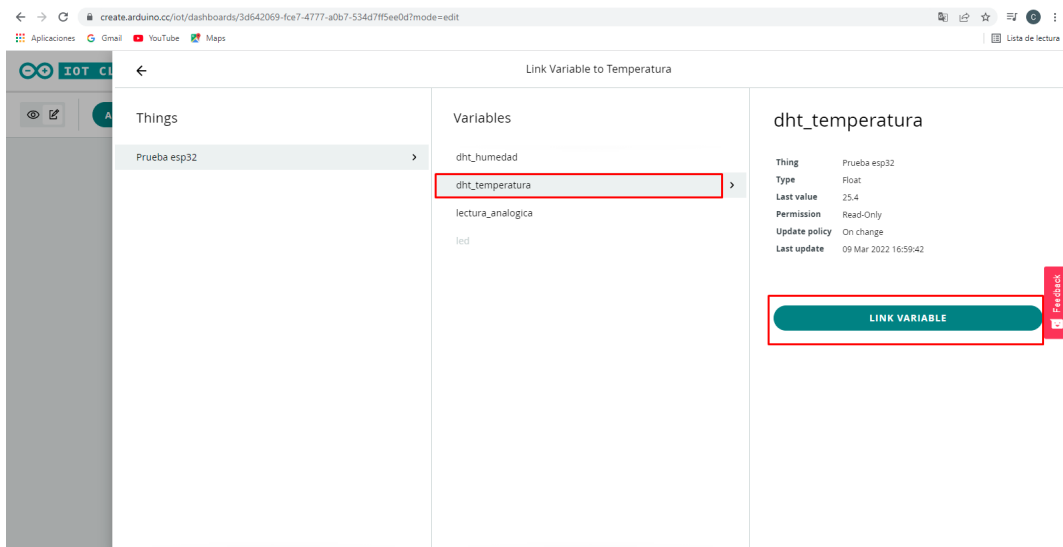
En los parámetros del widget le damos un nombre, definimos los valores mínimo y máximo



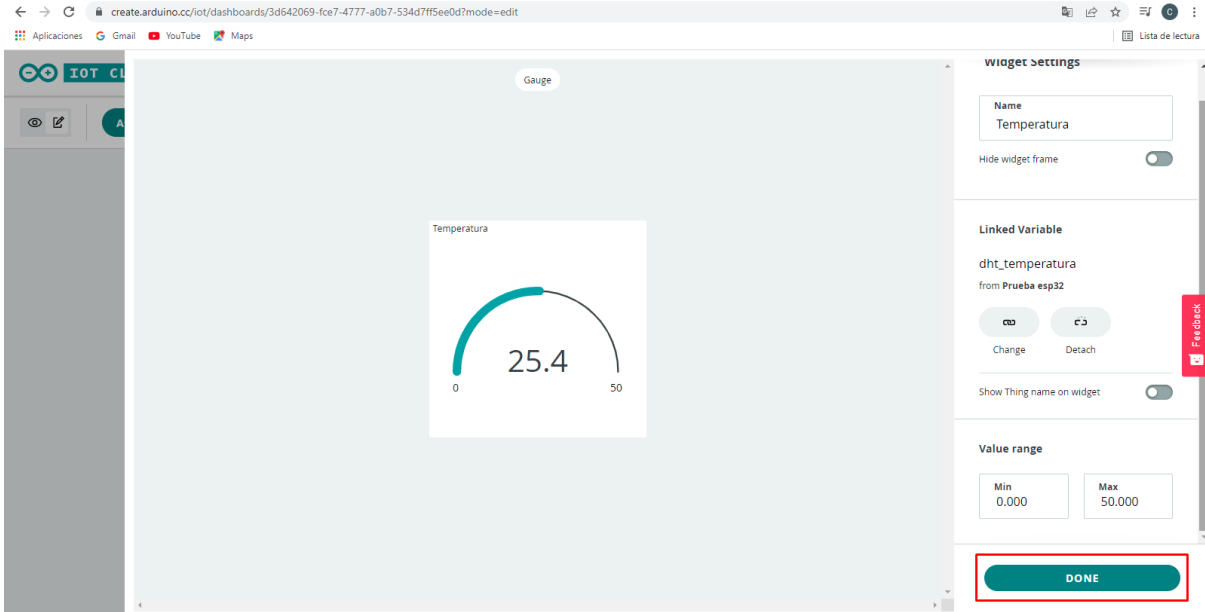
Tenemos que asociar el gauge a la variable que habíamos creado anteriormente



Seleccionamos la variable dht_temperature, luego damos click a “LINK VARIABLE”



Damos click en "DONE"

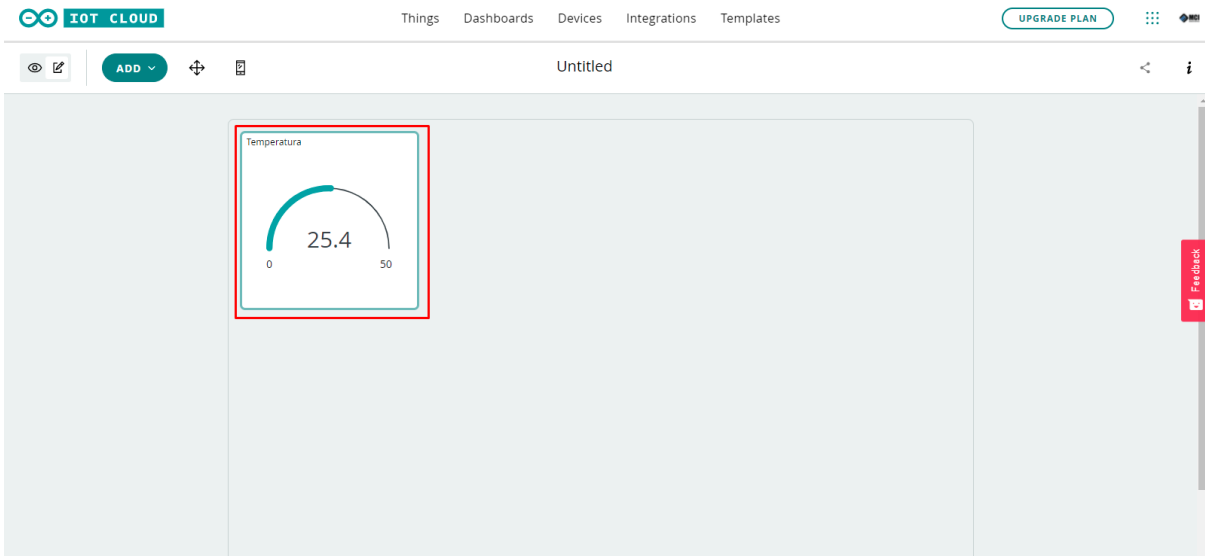


The screenshot shows the Arduino IoT Cloud dashboard editor. The main area displays a gauge widget titled "Temperatura" with a value of 25.4. The gauge has a scale from 0 to 50. To the right, the "Widget settings" panel is open, showing the following configuration:

- Name: Temperatura
- Hide widget frame:
- Linked Variable: dht_temperatura from Prueba esp32
- Change:
- Detach:
- Show Thing name on widget:
- Value range: Min 0.000, Max 50.000

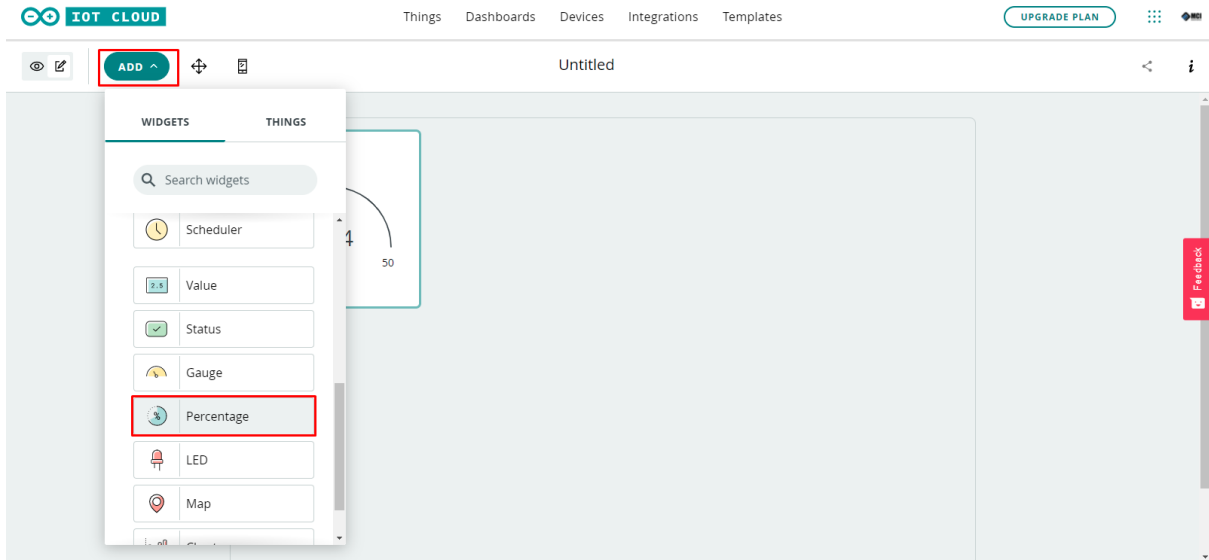
A red box highlights the "DONE" button at the bottom right of the settings panel.

El dashboard ya tiene nuestra variable

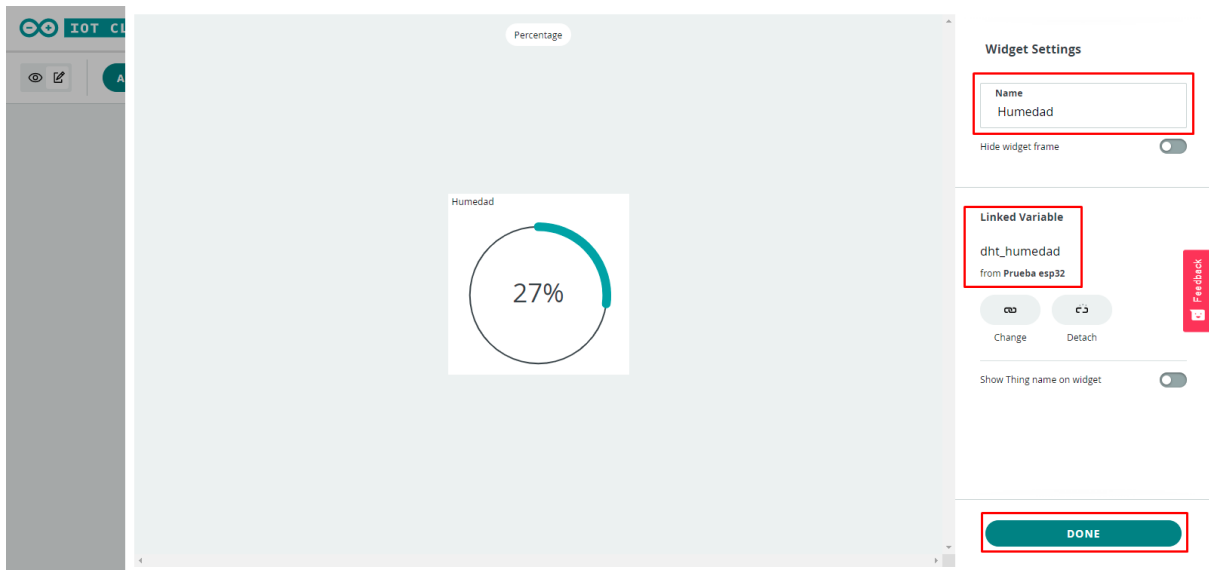


The screenshot shows the final dashboard view. The gauge widget titled "Temperatura" with a value of 25.4 is now displayed on the dashboard. The widget is highlighted with a red box. The dashboard title is "Untitled".

Para la humedad hacemos el mismo procedimiento, pero seleccionaremos el widget de porcentaje “Percentage”

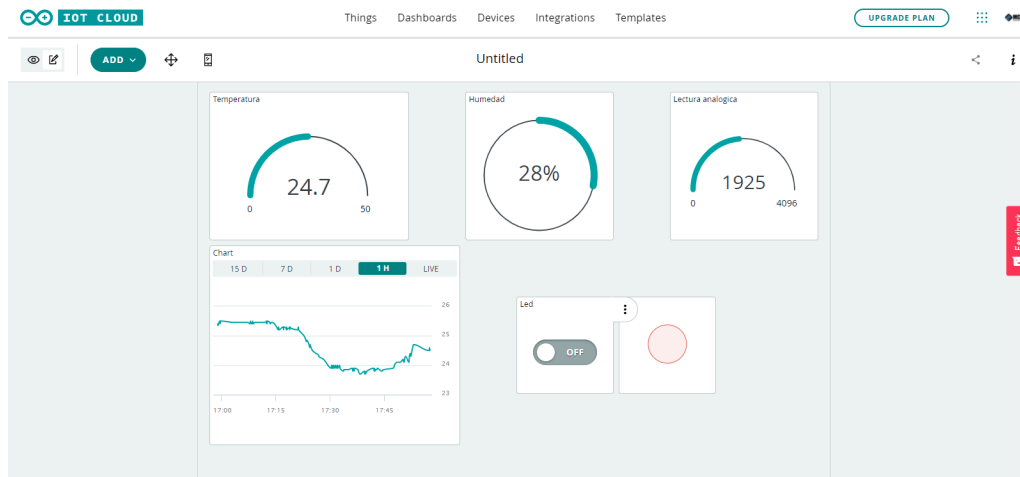


Seguimos el mismo procedimiento anterior

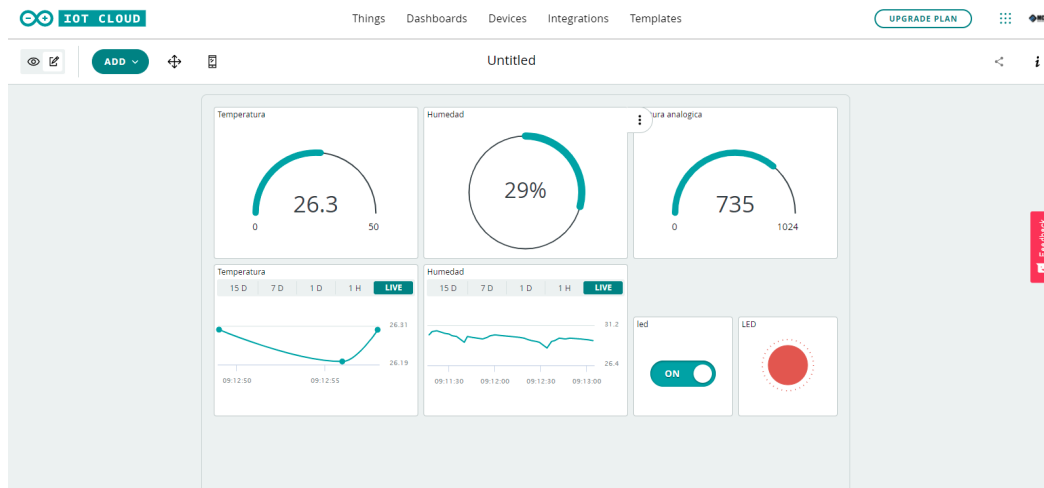


Podemos agregar otros widget asociados a la misma variable para tener más información o comportamiento, por ejemplo, en gráficas

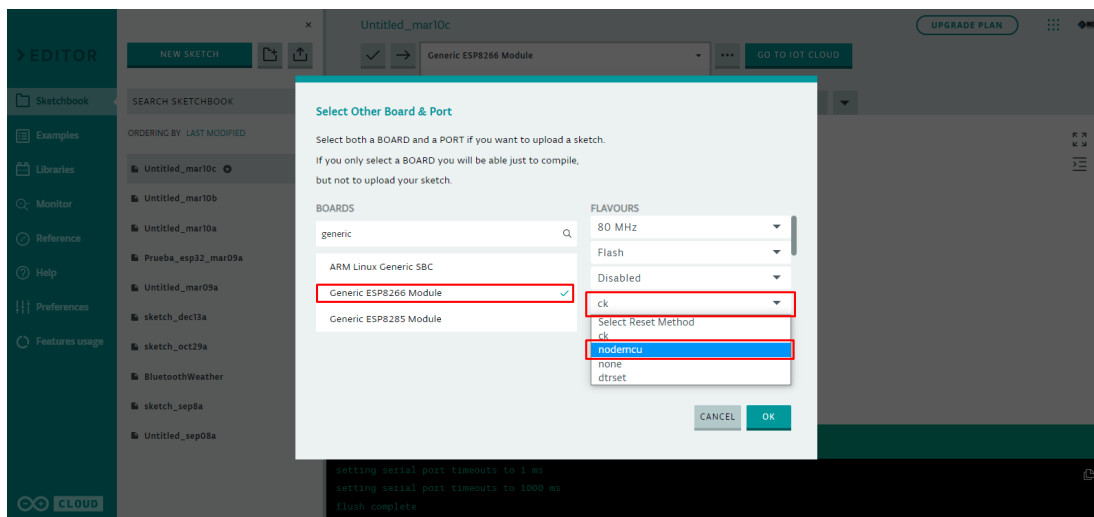
El dashboard completo con las variables quedaría así



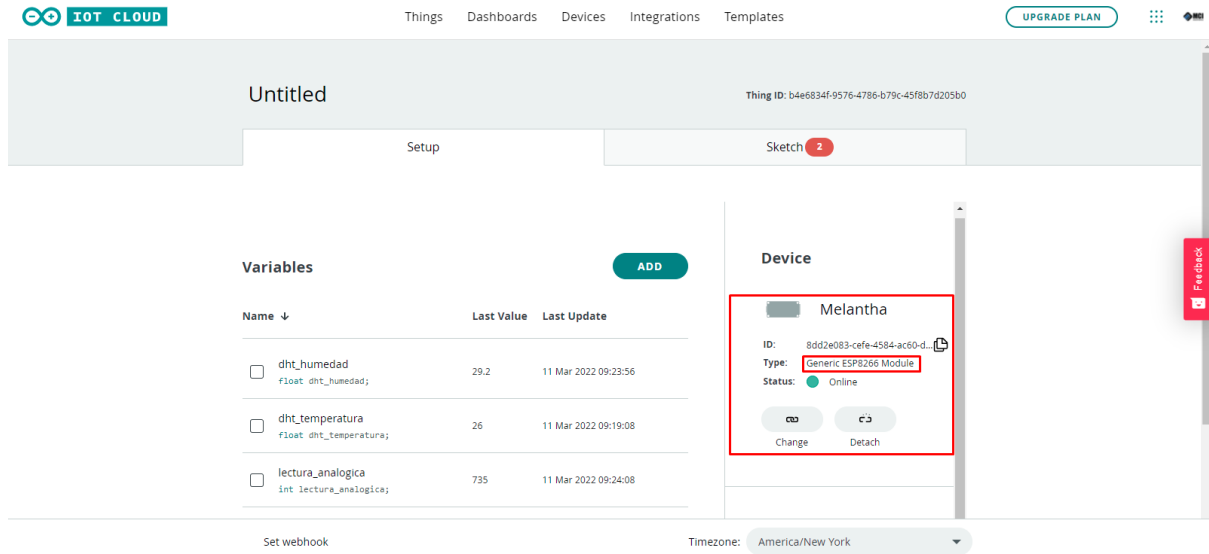
El procedimiento para el **esp8266** en vez del **esp32** que se muestra en las imágenes anteriores es el mismo obteniendo los mismo resultados,



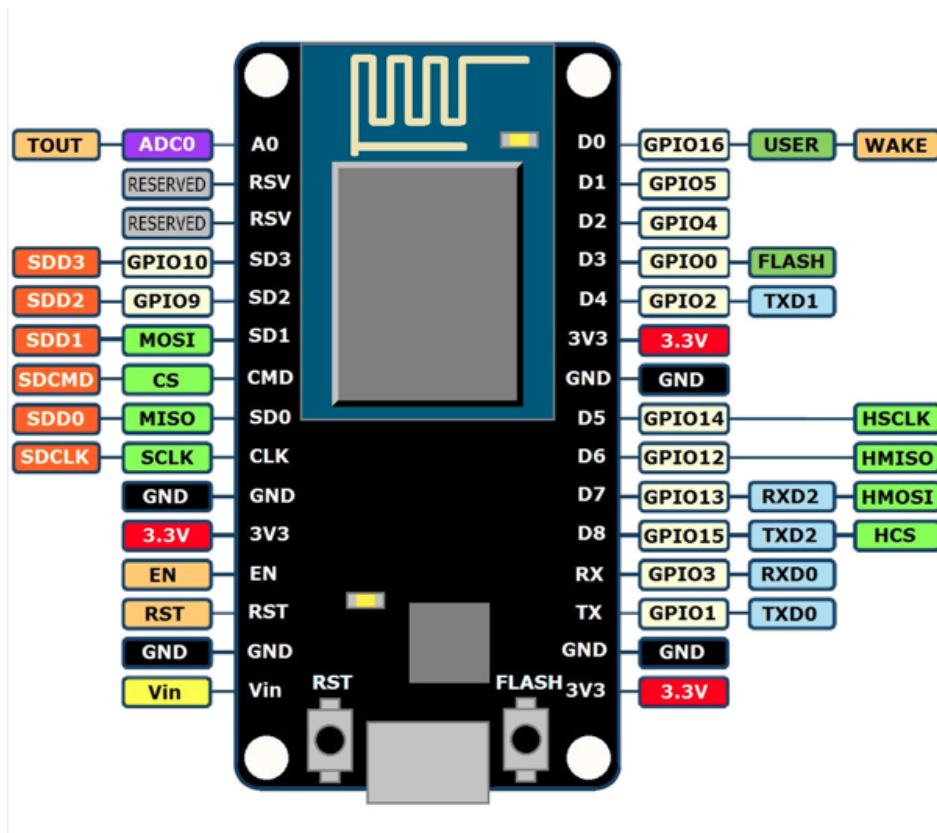
Sin embargo existen algunos pequeños cambios. Se debe seleccionar como tarjeta “Generic ESP8266 Module” para cargar el programa



- Lo mismo para la selección del dispositivo



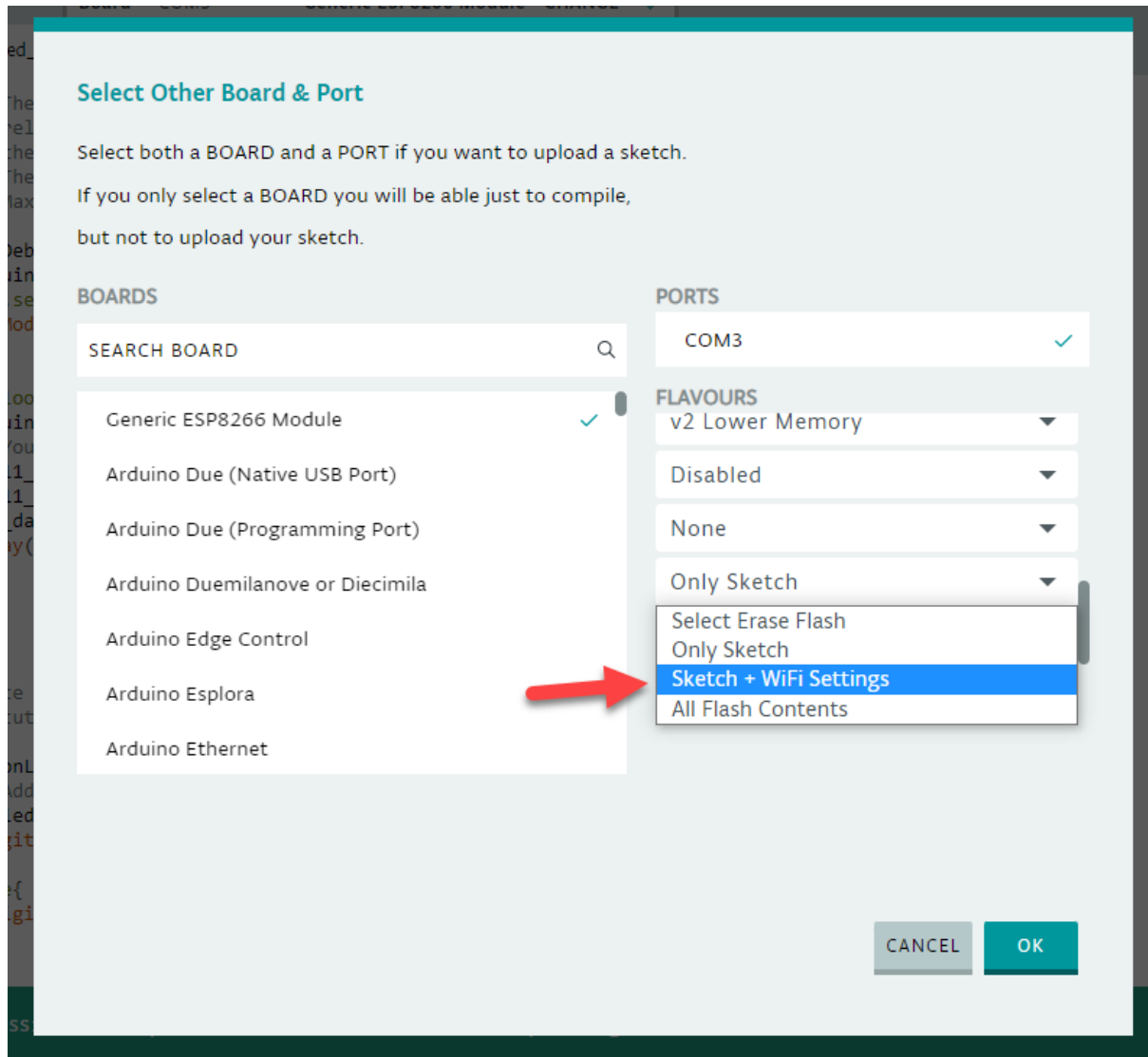
- Los pines usados en el programa se basan en el GPIO no en Dx impresos en la tarjeta



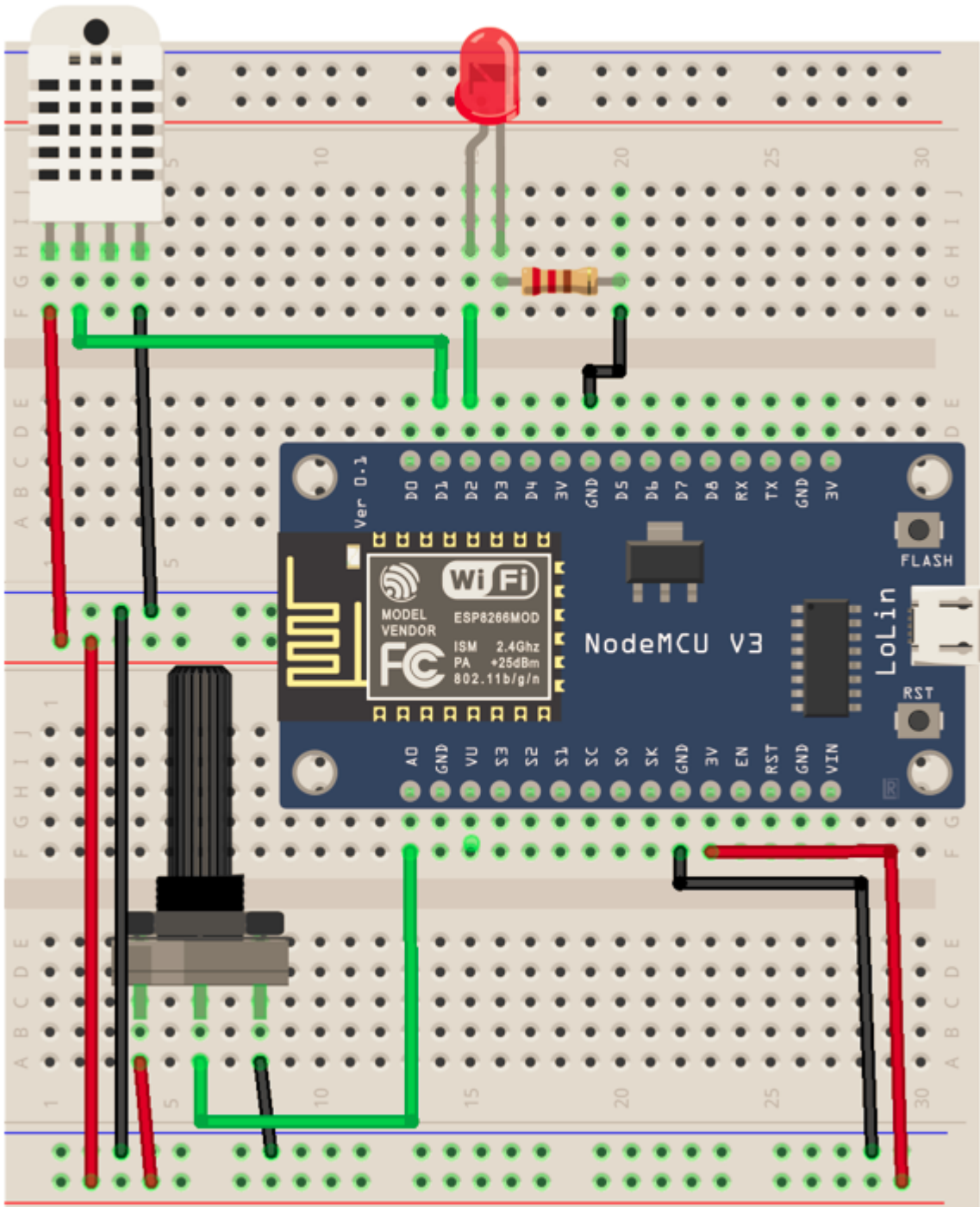
- Se deben usar los pines entre el D0 y D4.
- Programa usado para el Esp8266,

<https://create.arduino.cc/editor/camilobf/8c00add9-4193-4a8a-9034-f11cf5bb08f3/preview>

Si no es la primera vez que utilizas la placa recuerda no solo cargar en scketch en ella si no también la configuración del Wifi



El circuito construido es el siguiente:



El circuito armado se ve de la siguiente manera:

