

lot aplicado al monitoreo de variables ambientales

Grado sugerido: Once

Deiber Andrés Aldana Pulido

Publicado en el Banco Virtual de Recursos de Colombia Programa en el año 2025.

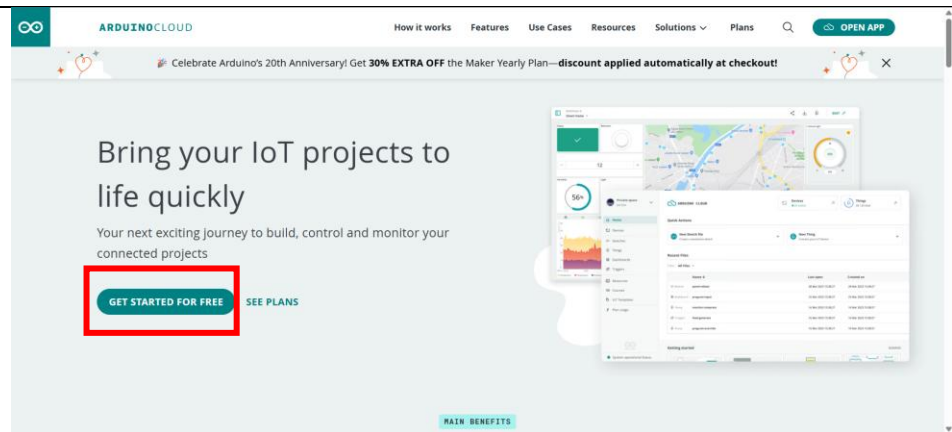
Este material se comparte bajo la licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0). Puede copiar y redistribuir el material en cualquier medio o formato, siempre que dé el crédito adecuado al autor, no lo use con fines comerciales, y no remezcle, transforme o cree a partir del material.

Para más información, consulte la licencia completa en [Deed - Attribution-NonCommercial-NoDerivatives 4.0 International - Creative Commons](#)

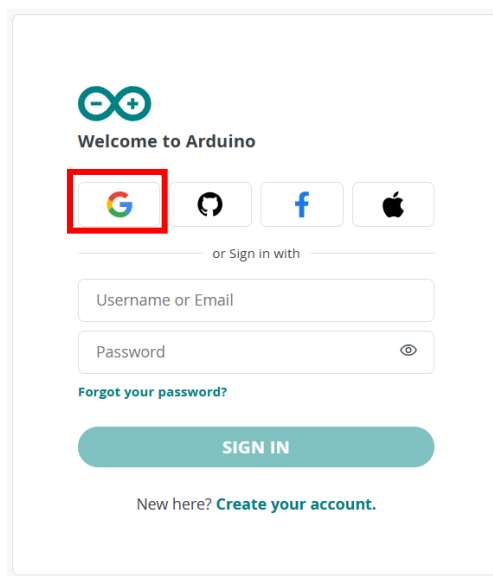
Para contactar al autor/a de este recurso, escriba a: ingaldana569@gmail.com

IoT APLICADO AL MONITOREO DE VARIABLES AMBIENTALES

Aprendizajes esperados	<p><i>Con esta guía podrás alcanzar los siguientes aprendizajes:</i></p> <ul style="list-style-type: none"> • Comprender qué es el Internet de las Cosas (IoT) y cómo se aplica al monitoreo ambiental. • Configurar un dispositivo ESP32 para capturar y enviar datos a la nube usando Arduino IoT Cloud. • Visualizar datos de sensores en tiempo real a través de un panel web o móvil.
Duración	<i>120 minutos</i>
Materiales Requeridos	<ul style="list-style-type: none"> • Computador con conexión a Internet • Placa ESP32 • Sensor DHT11 • 1 capacitor electrolítico de 1 uF • Protoboard • Cables Dupont • Cuenta gratuita en https://cloud.arduino.cc
Actividades para desarrollar	<p><i>El IoT es una red de dispositivos electrónicos (denominados cosas) que contienen sensores, software y otras tecnologías, y están conectados a Internet para intercambiar datos e interactuar con otros dispositivos y con personas (también llamados usuarios). El concepto de cosas incluye innumerables aparatos, como sensores y luces inteligentes para el hogar, pulseras que monitorizan la salud, piezas de sistemas de seguridad y componentes relacionados con la conducción automática; todos ellos, conectados a Internet. La mayor parte de la comunicación entre esos dispositivos consiste en enviar datos y recibir instrucciones. El IoT suele estar muy automatizado o mejorado con software inteligente, por lo que requiere poca interacción entre personas y máquinas, y a veces ninguna (MinnaLearn, 2022).</i></p> <p>Con el desarrollo de la presente guía, los estudiantes desarrollarán habilidades en programación de sensores para el monitoreo de variables y la transmisión y visualización de esta información a través de Internet, con la finalidad de automatizar el proceso de monitoreo de variables y brindar una buena experiencia al usuario.</p> <ol style="list-style-type: none"> 1. CONFIGURACIÓN DE PLATAFORMA ARDUINO IOT CLOUD (Arduino, 2025) <ul style="list-style-type: none"> ✓ Ingresar a https://cloud.arduino.cc/. Luego ingresar en la opción Get Started for free

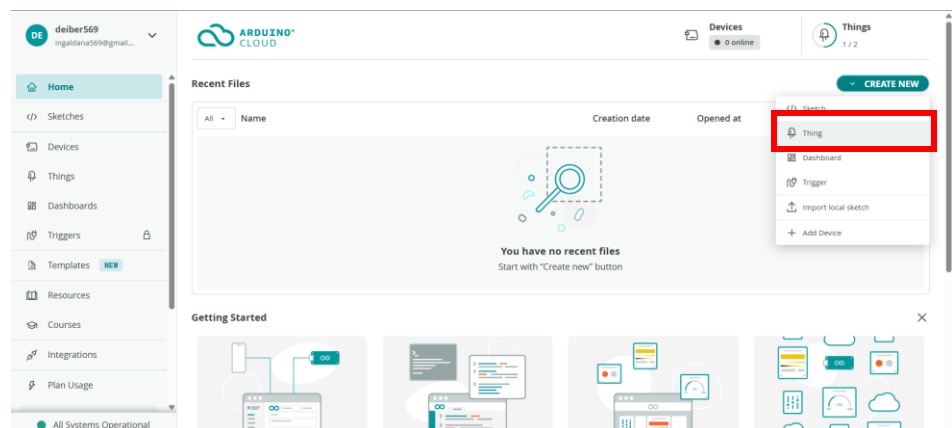


✓ Crear cuenta e ingresar

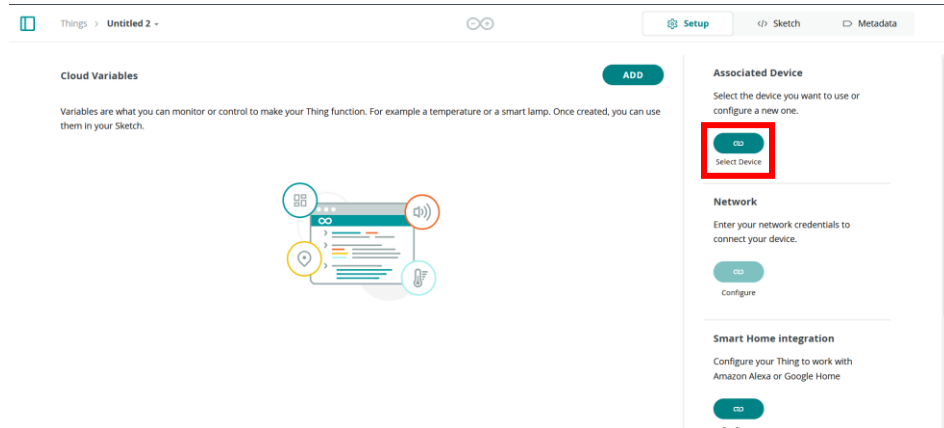


Nota: Se recomienda ingresar usando la cuenta de Gmail.

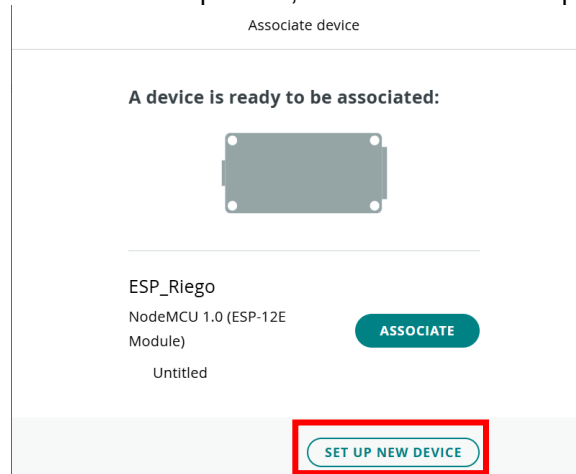
✓ Clic en crear nueva "Thing"



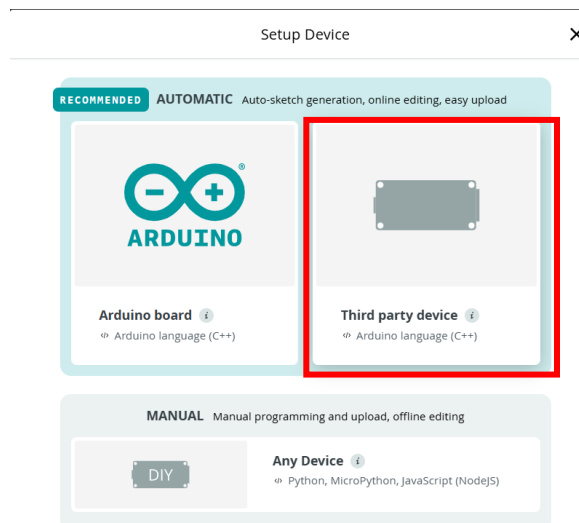
- ✓ Agregar nuevo dispositivo. En este paso se debe agregar la tarjeta que se tenga disponible, para el caso del ejemplo se cuenta con la tarjeta **DOIT ESP32 DEVKIT V1**, seguir los siguientes pasos:
- Clic en seleccionar dispositivo



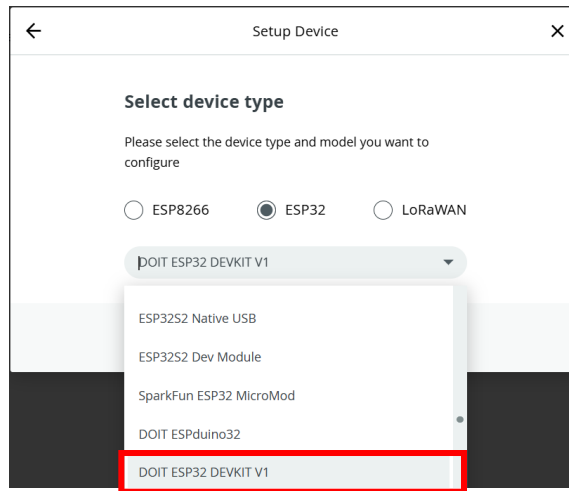
Nota: en caso de tener otro dispositivo, seleccionar nuevo dispositivo



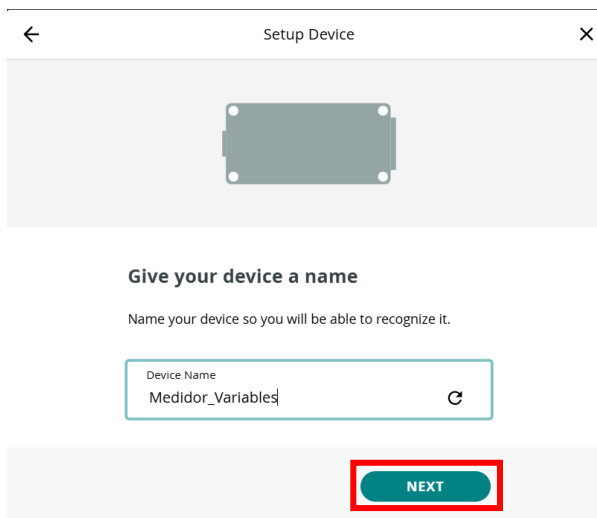
- Clic en dispositivos de terceros (tarjetas no oficiales de Arduino)



- Seleccionar la tarjeta DOIT ESP32 DEVKIT V1




- Asignar un nombre al dispositivo



- Copiar el ID del dispositivo y la llave secreta, ya que serán necesarios más adelante. Clic en continuar

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
[Redacted]


Secret Key
[Redacted]

☐ I saved my Device ID and Secret Key

CONTINUE

- Se finaliza el proceso dando clic en Done.

Setup device



Congratulations! You are all set

Your DOIT ESP32 DEVKIT V1 Medidor_Variables has been successfully set up. You can now connect your device to sensors, actuators and other inputs or outputs!

DONE


- ✓ Agregar red Wifi. En este paso se debe agregar las credenciales de la red wifi (nombre y contraseña) y la clave del dispositivo (copiada previamente)

Things > Untitled 2 -

Setup Sketch Metadata

Cloud Variables **ADD**

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.



Associated Device

Medidor_variables

ID: 2a6c8d74-bbd3-4b64-8ac3-8f2f...

Type: DOIT ESP32 DEVKIT V1

Status: Offline

Change **Detach**

Network

Enter your network credentials to connect your device.

Configure

Smart Home integration

Configure your Thing to work with Amazon Alexa or Google Home

Configure network

Enter your network credentials to allow your device to connect to the Cloud.

Wi-Fi Name *

Password

Secret Key *

IMPORTANT: Remember to go to the "Sketch" tab and upload the sketch to load the credentials on the board.

SAVE

- ✓ Crear Variables. En este paso se deben crear las variables que se van a monitorear, para el ejemplo sólo se va a hacer lectura de temperatura y humedad ambiente con el sensor DHT11.

Things > Untitled 2

Setup

Sketch

Metadata

Cloud Variables

ADD

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.

Associated Device

Medidor_variables

ID: 2aecfd74-bbd3-4b64-8ac1-af2f...

Type: DHT ESP32 DEVKIT V1

Status: Offline

Change

Detach

Network

Wi-Fi Name: Familia_...

Password:

Secret Key:

Change

Smart Home integration

Configure your Thing to work with Amazon Alexa or Google Home

Variable para la temperatura

Add variable

Name

temperatura

Sync with other Things

Floating Point Number eg. 1.55

Declaration

float temperatura;

Variable Permission

Read & Write

Read Only

CANCEL

ADD VARIABLE

Variable para la humedad

Add variable

Name

humedad

Sync with other Things

Floating Point Number eg. 1.55

Declaration

float humedad;

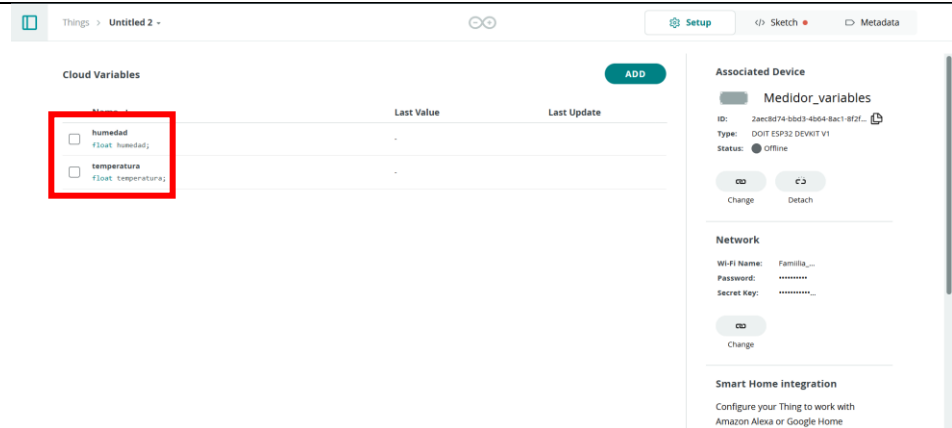
Variable Permission

Read & Write

Read Only

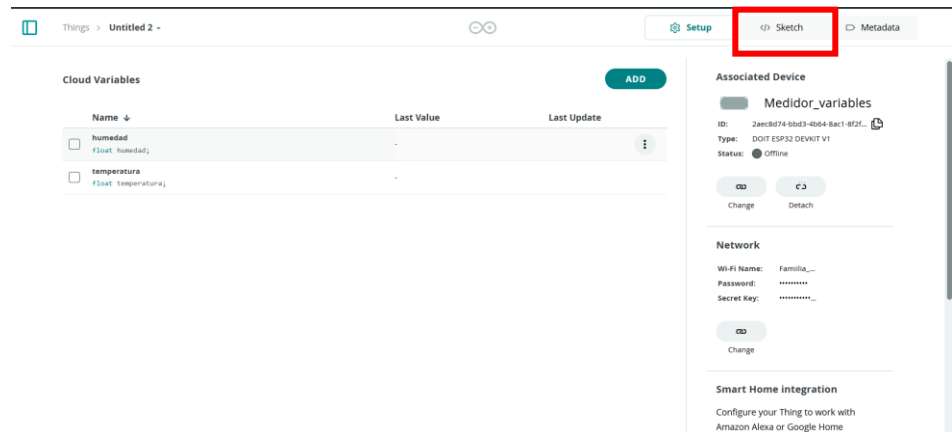
CANCEL

ADD VARIABLE



2. MEDIDOR DE VARIABLES CON IoT

- ✓ Dar clic en “Sketch”. En este paso se debe crear el algoritmo, teniendo en cuenta que se van a usar las variables creadas en el paso anterior.



- ✓ Escribir el siguiente algoritmo

```
#include "thingProperties.h" // Variables y conexión configuradas en Arduino IoT Cloud
#include <DHT.h>             // Librería para el sensor DHT11
#define DHTPIN 13           // Pin digital donde se conecta el sensor
#define DHTTYPE DHT11      // Tipo de sensor (DHT11)
DHT dht(DHTPIN, DHTTYPE);  // Se crea el objeto del sensor

void setup() {
  Serial.begin(9600);        // Inicia comunicación serial
  delay(1500);               // Espera para evitar errores de conexión
  initProperties();          // Inicializa variables de la nube
  ArduinoCloud.begin(ArduinoIoTPreferredConnection); // Conecta a IoT Cloud
  setDebugMessageLevel(2);   // Nivel de mensajes (0 a 4)
  ArduinoCloud.printDebugInfo(); // Imprime información de conexión
  dht.begin();               // Inicia el sensor DHT11
}
```

```

void loop() {
  ArduinoCloud.update();    // Sincroniza variables con la nube

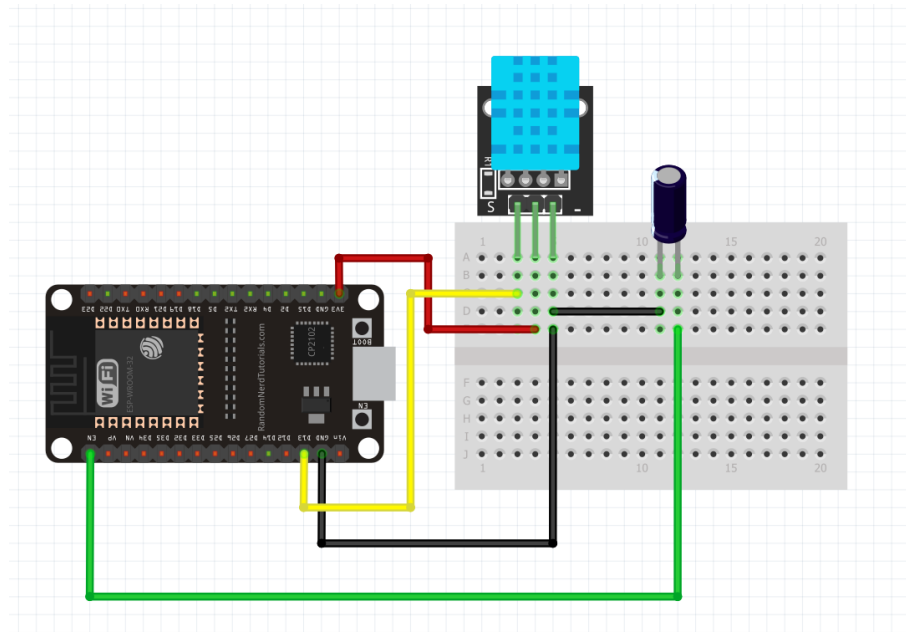
  // Lectura de temperatura y humedad
  temperatura = dht.readTemperature();
  humedad = dht.readHumidity();

  // Mostrar valores por monitor serial
  Serial.print("Temperatura: ");
  Serial.println(temperatura);
  Serial.print("Humedad: ");
  Serial.println(humedad);

  delay(5000); // Espera 5 segundos entre lecturas
}

```

- ✓ Realizar el montaje del circuito electrónico



Nota: Circuito diseñado usando el software Fritzing (2025). Se usa el pin D13 (aunque se puede usar cualquiera) y se debe conectar un capacitor electrolítico de 1 uF entre en pin EN y tierra, esto para poder programar la tarjeta sin tener que presionar el pulsador BOOT. En el Anexo A se muestra la distribución de pines de la tarjeta ESP32 DEVKIT V1.

3. CREACIÓN DE DASHBOARD

- ✓ Dar clic en Dashboards y luego seleccionar la opción para crear uno nuevo

DE deiber569
ingalana569@gmail...

Home

Sketches

Devices

Things

Dashboards

Triggers

Templates NEW

Resources

Courses

Integrations

Plan Usage

All Systems Operational

Dashboards

Search and filter Dashboards

View

+ DASHBOARD

Dashboard Name

Last modified

Sharing with

Kingo Automatizado

27 Jul 2023, 8:34

Deiber Andres Albano

✓ Dar clic en la opción de editar

Dashboards > Dashboard Monitore...

EDIT

✓ Agregar Widget para la variable temperatura

ADD

WIDGETS

THINGS

Search widget or variable type

All

Interaction

Visualisation

Annotation

Switch

Push Button

Slider

Stepper

Messenger

Color

Dimmed light

Colored light

Time Picker

Scheduler

Value

Value Selector

Value Dropdown

Status

Gauge

Percentage

LED

Map

Chart

Advanced Map

Gauge

Supported variable types: Float, Int and all Specialized types based on these 2 types

The readability of a value and the clarity of a slider, all in one classic widget.

24.1

✓ Cambiar parámetros y asociar a la variable “temperatura”

Pág. 9 de 16

Widget Settings

Name

Temperatura

Hide widget frame

Linked variable

This widget is displaying example data.
Select a source variable to display its value.

Link Variable

Value range

Min

0

Max

100

DONE

←

Link Variable to Temperatura

Thing

Untitled

ESP_Riego - NodeMCU 1.0 (ESP-12E Module)

Untitled 2

Medidor_variables - DOIT ESP32 DEVKIT V1

Variables

humedad

Float

temperatura

Float

temperatura

Thing

Untitled 2

Type

Float

Last value

-

Permission

Read-Only

Update policy

On change

Last update

N/A

LINK VARIABLE

✓ Realizar el mismo procedimiento para la variable “humedad”, con la diferencia que el Widget será de tipo porcentual.

Dashboards

Dashboard Monitore...

ADD

WIDGETS

THINGS

Search widget or variable type

ALL

Interaction

Visualisation

Annotation

messenger

Time Picker

Value Dropdown

LED

Advanced Chart

Color

Scheduler

Status

Map

Sticky Note

Value

Gauge

Chart

Image

Percentage

Value Selector

Advanced Map

Image Map

Percentage

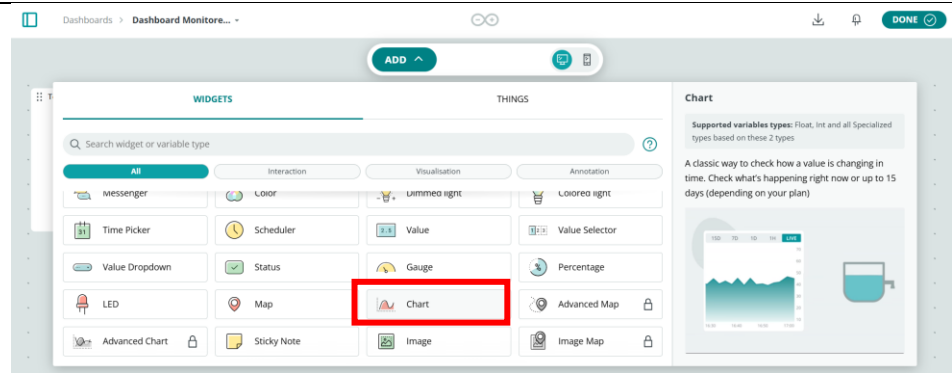
Supported variables types: Int, Float and all Specialized types based on these 2 types

A simple widget to show a percentage. You can also set a threshold value to make it change color dynamically. Easily modified with custom icons.

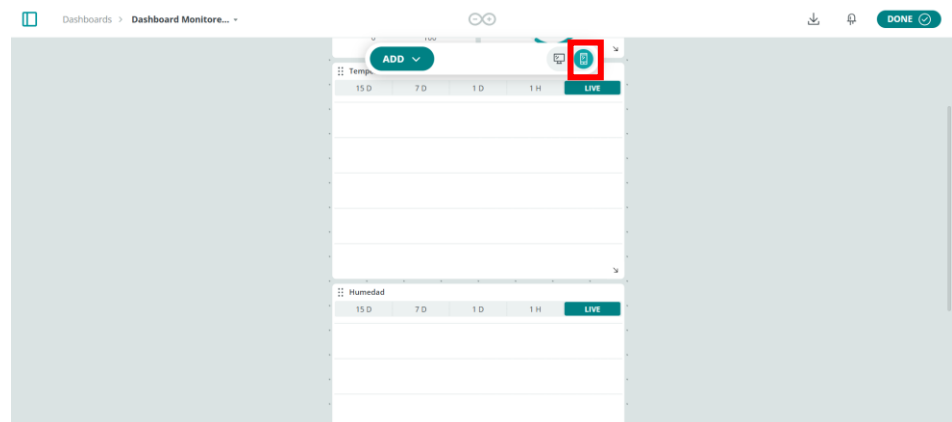
100%

✓ Agregar Widgets de gráfica en el tiempo para las dos variables:

Pág. 10 de 16



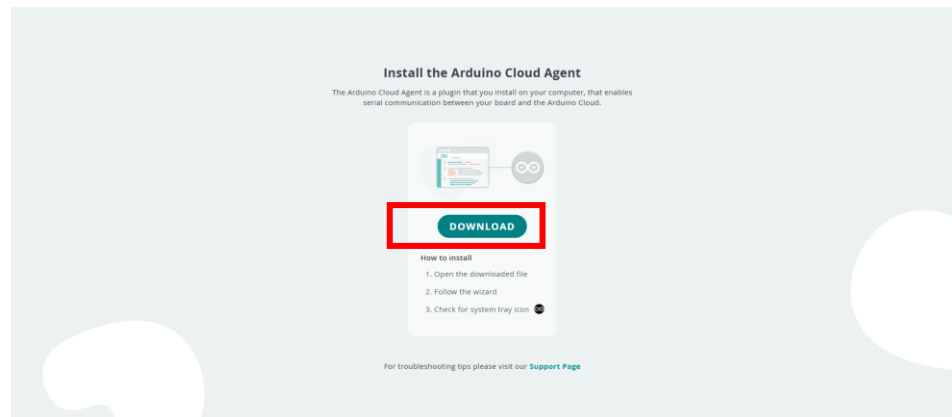
- ✓ Seleccionar la vista para dispositivos móviles y organizar las diferentes Widgets



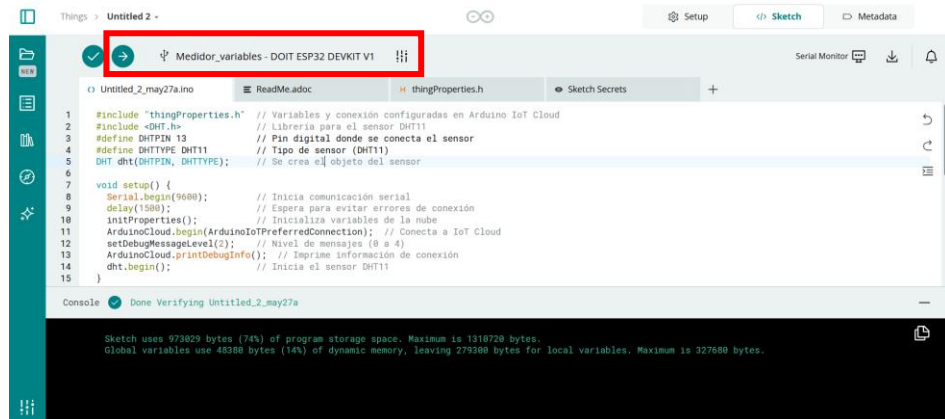
4. CARGA DEL PROGRAMA A LA TARJETA

Para poder cargar el código a la tarjeta de forma alámbrica es necesario tener el **Arduino Create Agent** instalado, si no lo tienen deben seguir los siguientes pasos:

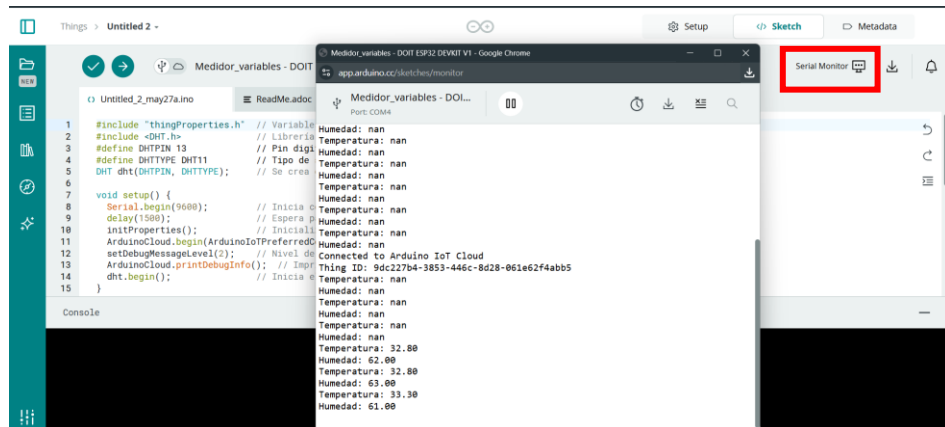
- ✓ Ingresar a la página <https://cloud.arduino.cc/download-agent>
- ✓ Descargar e instalar el agente



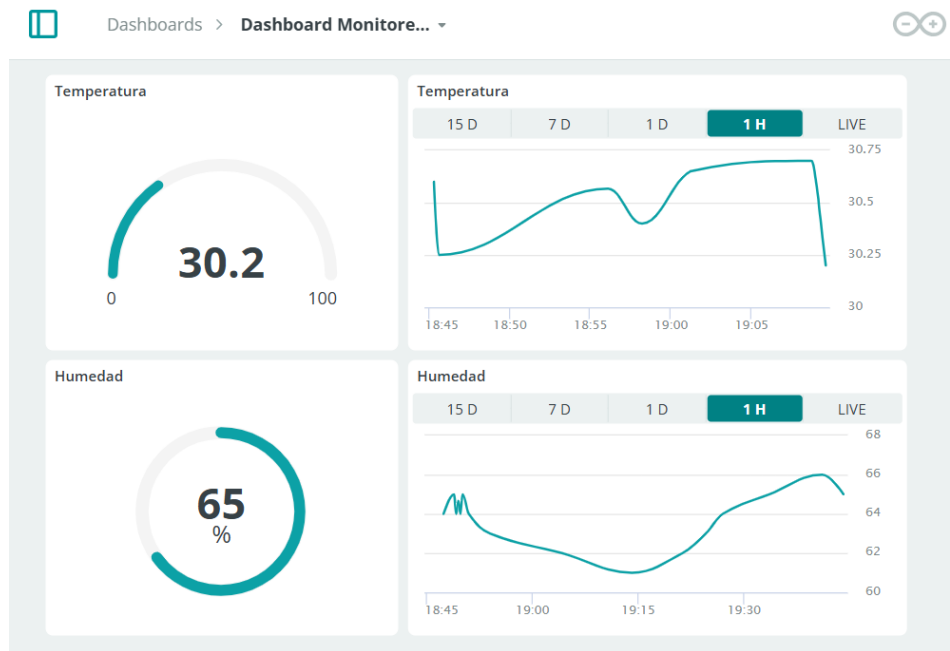
- ✓ Una vez instalado el agente, volver al sketch y verificar que reconozca la tarjeta. A continuación, dar clic en cargar



- ✓ Para verificar la transmisión de datos se puede abrir el monitor serial




- ✓ Una vez se verifique que la tarjeta si está conectada a internet y que está realizando la lectura de las variables, abrir el dashboard y visualizar la información.




- ✓ Para visualizar el dashboard desde un dispositivo móvil deben instalar la aplicación **Arduino IoT Cloud Remote**. Posteriormente ingresar con el mismo correo con el que ingresaron por el navegador web.

4:50 


Welcome to Arduino

Email or Username




Password 

[Forgot your password?](#)

SIGN IN

New here? [Create your account](#)

or sign in with

- ✓ Por último, seleccionar el dashboard diseñado y se podrá visualizar la información en tiempo real.

	
Adaptaciones	<ul style="list-style-type: none"> • contextos sin conectividad <p>En instituciones donde no se cuenta con acceso a Internet, el sistema puede implementarse en modo local (Sin la aplicación de IoT) usando el monitor serial del IDE de Arduino, permitiendo visualizar los datos sin depender de la nube. Los estudiantes pueden registrar las lecturas en tablas escritas, y realizar gráficas de la variación de la temperatura y humedad. Esta estrategia permite desarrollar pensamiento computacional y habilidades técnicas sin requerir conectividad permanente.</p> <ul style="list-style-type: none"> • Trabajo con grupos grandes de estudiantes <p>Durante el desarrollo de esta guía se sugiere conformar equipos de cinco estudiantes, asignando a cada uno un rol con responsabilidades específicas. Esta estrategia promueve el trabajo colaborativo, la equidad en la participación y el desarrollo de habilidades técnicas y comunicativas.</p> <ul style="list-style-type: none"> ✓ Programador: Es el encargado de escribir, cargar y ajustar el código en la plataforma Arduino Cloud. Interpreta la lógica de funcionamiento del sistema, prueba el programa y corrige errores relacionados con la lectura del sensor, el envío de datos o el encendido de alertas. También explica al equipo cómo funciona el algoritmo.

	<ul style="list-style-type: none"> ✓ Ensamblador: Construye el circuito físico en la protoboard conectando correctamente los componentes (sensor, capacitor etc) a la placa ESP32. Sigue el diagrama de conexiones y verifica que el montaje sea seguro y funcional. Trabaja de forma coordinada con el programador para hacer pruebas conjuntas. ✓ Ingeniero de materiales: Organiza, selecciona y entrega al equipo todos los componentes necesarios para el proyecto. Lleva un control de los materiales usados y cuida que se utilicen correctamente. Puede proponer reemplazos cuando algún elemento no esté disponible y sugiere mejoras prácticas en el uso de recursos. ✓ Validador: Se encarga de probar el sistema completo. Revisa que el sensor registre valores, que los datos se visualicen correctamente y que las alertas se activen en las condiciones definidas. Registra observaciones y propone ajustes en el montaje o el código si es necesario. También puede medir la precisión del sistema. ✓ Comunicador: Documenta con fotos, videos o apuntes cada fase del proyecto. Es quien presenta los resultados del grupo ante el docente o en exposiciones. Se asegura de que la evidencia se organice de forma clara y que el equipo exprese con precisión lo aprendido. También puede redactar informes o conclusiones. <p>• Zonas rurales</p> <p>En entornos rurales, la guía se puede aplicar a situaciones reales como el monitoreo de huertas escolares, viveros o galpones. Los estudiantes pueden utilizar sensores económicos, adaptar el prototipo a las necesidades locales y registrar sus observaciones manualmente. Esta adaptación fortalece el vínculo entre tecnología y comunidad, permitiendo aprendizajes contextualizados y útiles para la vida cotidiana de los estudiantes.</p> <p>• Diversidad de género y etnicidad</p> <p>Se sugiere rotar los roles dentro de los equipos para asegurar que todos los estudiantes, sin distinción de género u origen cultural, participen activamente en todas las fases del proyecto. Además, se puede contextualizar el uso del sistema en escenarios culturalmente significativos para las comunidades presentes en el aula, por ejemplo, monitoreando espacios que sean valorados o utilizados por distintos grupos familiares o comunitarios. Esta adaptación promueve la inclusión y el reconocimiento de la diversidad como un valor en el proceso de aprendizaje.</p>
Referencias	<p>Arduino. (2025). <i>Arduino Cloud</i>. https://app.arduino.cc</p> <p>Fritzing. (2025). <i>Welcome to Fritzing</i>. https://fritzing.org/</p> <p>MinnaLearn. (2022). <i>Introducción al Internet de las cosas</i>. https://courses.minnalearn.com/es/courses/emerging-technologies/the-internet-of-things/an-introduction-to-the-internet-of-things/</p> <p>Embedded Systems Design. (2025). <i>Overview of the ESP32 DevKit DOIT V1</i>. https://embedded-systems-design.github.io/overview-of-the-esp32-devkit-doit-v1/</p>

ANEXOS

ANEXO A. Distribución de pines de la tarjeta ESP32 DEVKIT V1

ESP32 DEVKIT V1 – DOIT version with 30 GPIOs

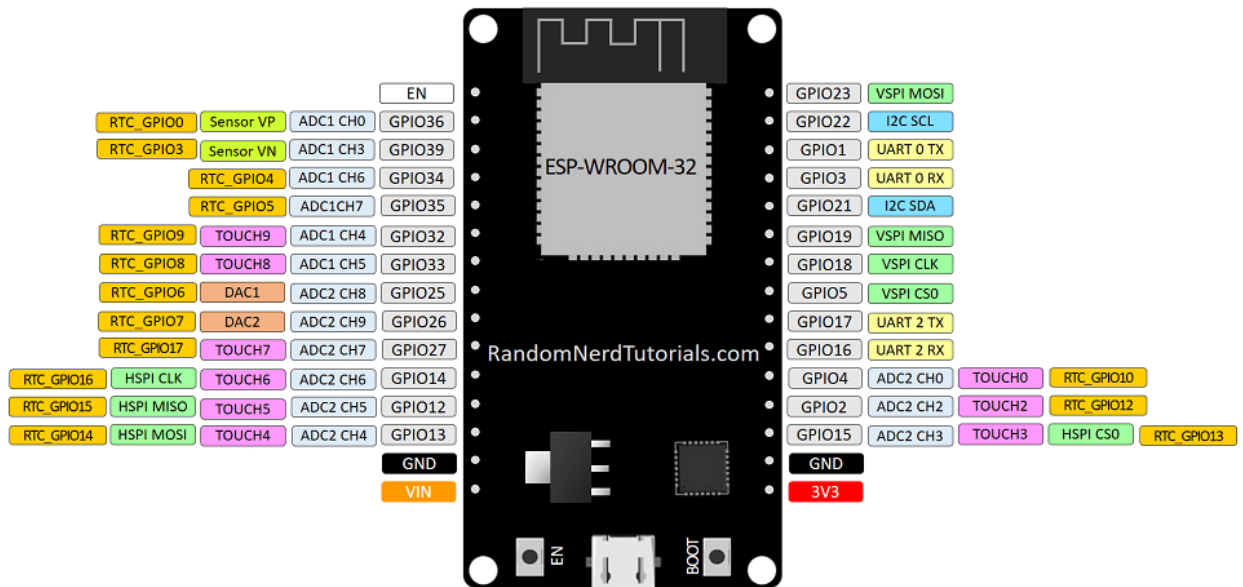


Imagen tomada de (Embedded Systems Design, 2025)