

iot 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 APPLICADO AL MONITOREO DE VARIABLES AMBIENTALES

Aprendizajes esperados	<p>Con esta guía podrás alcanzar los siguientes aprendizajes:</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	120 minutos
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

Bring your IoT projects to life quickly

Your next exciting journey to build, control and monitor your connected projects

GET STARTED FOR FREE SEE PLANS

✓ Crear cuenta e ingresar

Welcome to Arduino

or Sign in with

Username or Email

Password

Forgot your password?

SIGN IN

New here? [Create your account.](#)

Nota: Se recomienda ingresar usando la cuenta de Gmail.

✓ Clic en crear nueva “Thing”

Recent Files

Devices 0 online

Things 1 / 2

CREATE NEW

Thing

Dashboard

Trigger

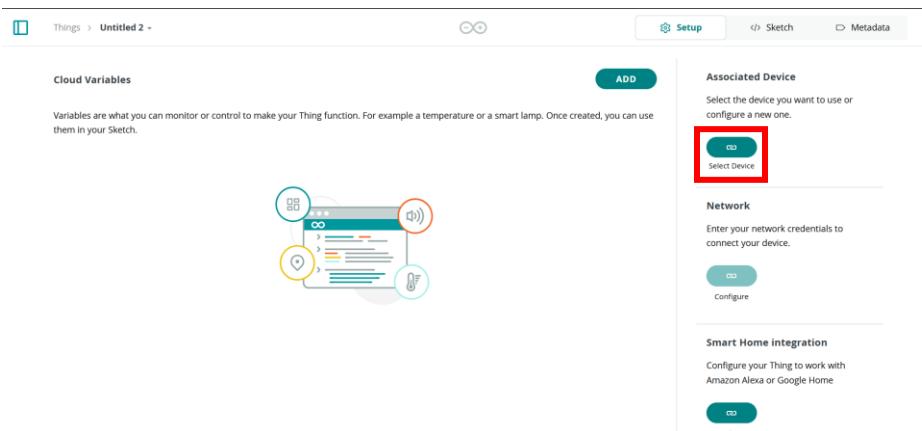
Import local sketch

Add Device

Home Sketches Devices Things Dashboards Triggers Templates NEW Resources Courses Integrations Plan Usage All Systems Operational

Getting Started

- ✓ 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

Associate device

A device is ready to be associated:

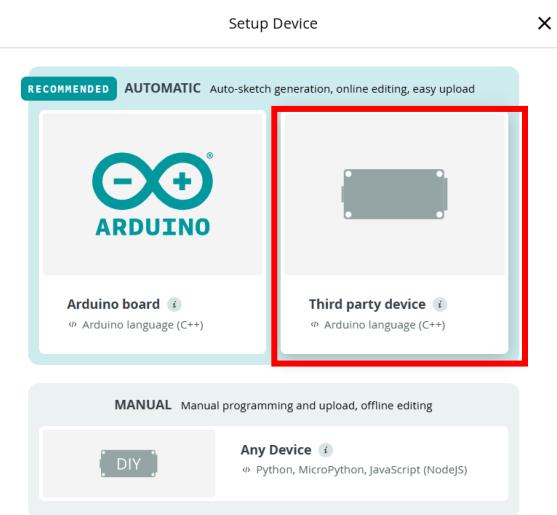
ESP_Riego
NodeMCU 1.0 (ESP-12E Module)

Untitled

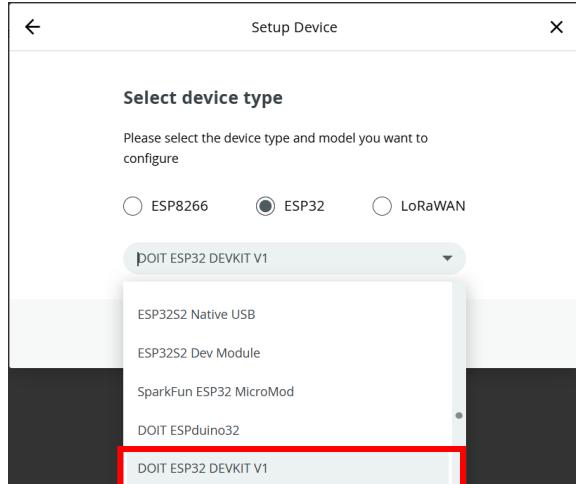
ASSOCIATE

SET UP NEW DEVICE

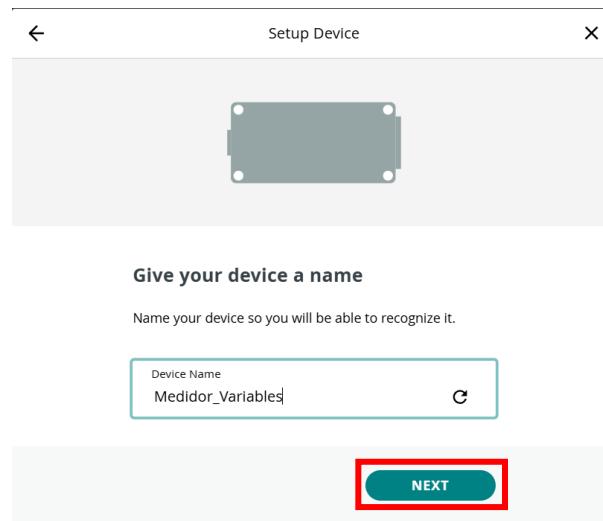
- Clic en dispositivos de terceros (tarjetas no oficiales de Arduino)



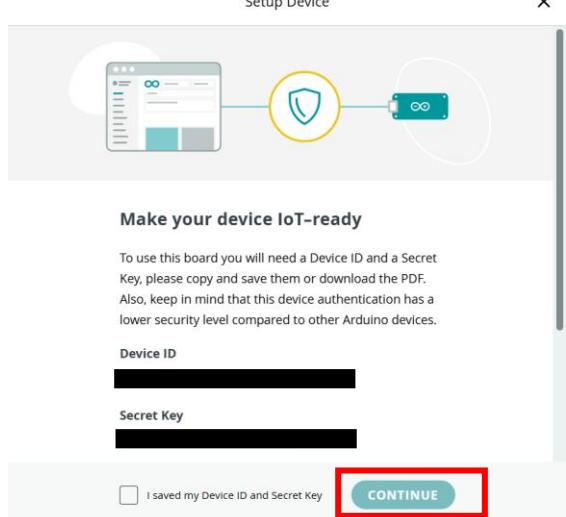
- Seleccionar la tarjeta DOIT ESP32 DEVKIT V1



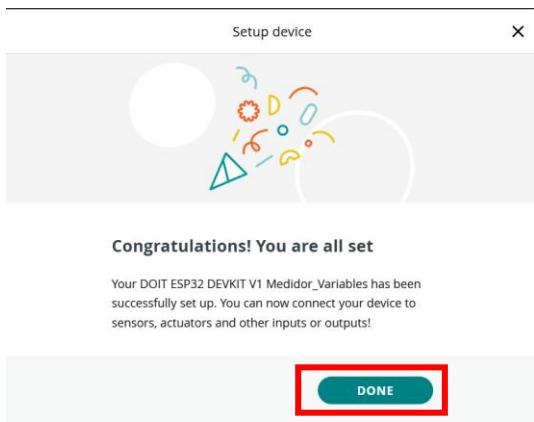
- Asignar un nombre al dispositivo



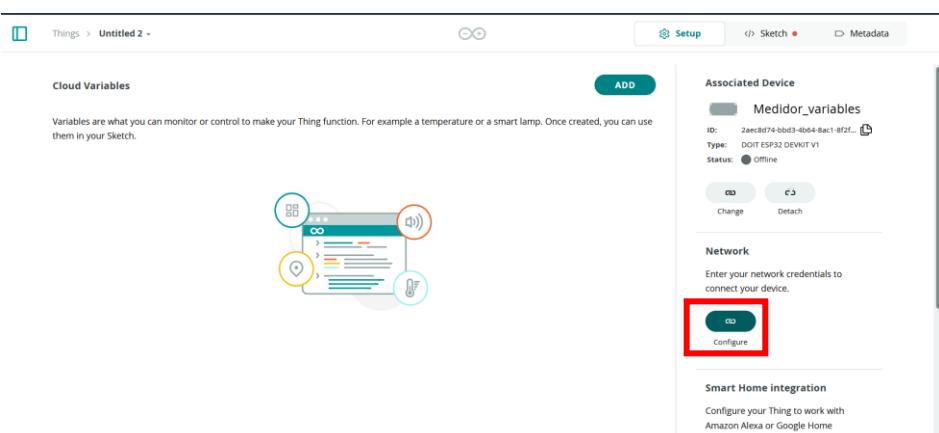
- Copiar el ID del dispositivo y la llave secreta, ya que serán necesarios más adelante. Clic en continuar



- Se finaliza el proceso dando clic en 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)



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.

Cloud 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.

Associated Device

Medidor_variables

ID: 2ae03f74-bb03-40d4-8ac1-8f2f... [Edit](#)

Type: DOTT 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

The screenshot shows the Arduino IoT Cloud interface for a device named "Medidor_variables". In the "Cloud Variables" section, there are two variables: "humedad" (float) and "temperatura" (float). The "Associated Device" section shows the device's ID, type (DOIT ESP32 DEVKIT V1), and status (Offline). The "Network" section displays Wi-Fi settings. The "Smart Home integration" section is present.

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.

The screenshot shows the same interface as above, but the "Sketch" tab is highlighted in red. This indicates the user should click on it to begin writing the algorithm.

- ✓ Escribir el siguiente algoritmo

```

#include "thingProperties.h" // Variables y conexión configuradas en Arduino IoT Cloud
#include <DHT.h> // Librería para el sensor DHT
#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

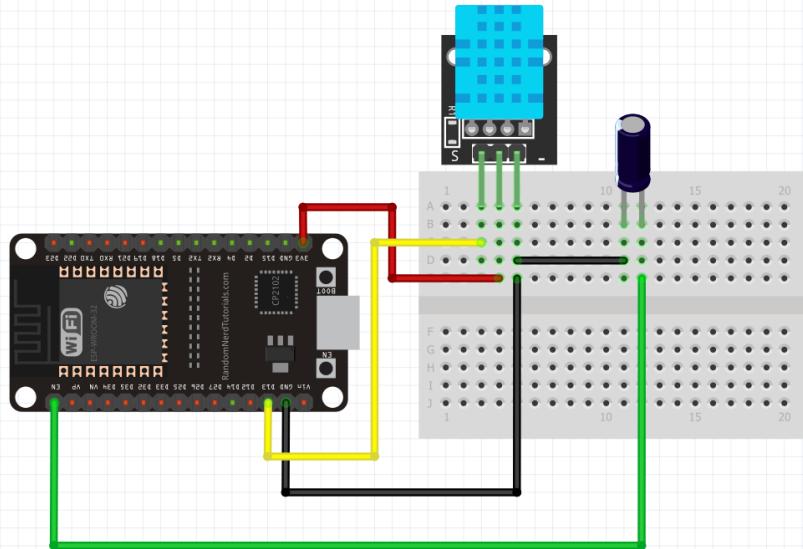
    // Lectura de temperatura y humedad
    temperatura = dht.readTemperature();
    humedad = dht.readHumidity();

    // 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

The screenshot shows the ThingSpeak web interface. On the left, a sidebar menu includes Home, Sketches, Devices, Things, Dashboards (selected), Triggers, Templates (NEW), Resources, Courses, Integrations, and Plan Usage. A status bar at the bottom indicates "All Systems Operational". The main area is titled "Dashboards" and lists a single dashboard named "Riego Automatizado". To the right of the dashboard list are buttons for "Search and filter Dashboards", "View", "Last modified" (27 jul 2023, 8:34), "Sharing with" (Deiber Andres Aldana), and a prominent green "DASHBOARD" button with a red border.

✓ Dar clic en la opción de editar

The screenshot shows the "Dashboard Monitor..." page in edit mode. The top navigation bar includes "Dashboards > Dashboard Monitor..." and a blue "EDIT" button with a red border. The main content area is currently empty, showing a large light gray space.

✓ Agregar Widget para la variable temperatura

The screenshot shows the "ADD" dialog for selecting a widget. The "WIDGETS" tab is selected, displaying various options like Switch, Push Button, Slider, Stepper, Messenger, Color, Dimmed light, Colored light, Time Picker, Scheduler, Value, Value Selector, Percentage, Value Dropdown, Status, LED, Map, Chart, and Gauge. The "Gauge" option is highlighted with a red border. To the right, a detailed description of the Gauge widget is shown, stating it supports float, int, and specialized types, and provides readability and clarity for values. It includes a preview image of a gauge with the value "24.1".

✓ Cambiar parámetros y asociar a la variable “temperatura”

Widget Settings

Name
Temperatura

Hide widget frame

Linked variable i

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	Variables	temperatura
Untitled ESP_Riego - NodeMCU 1.0 (ESP-12E Module)	humedad Float temperatura Float	Thing: Untitled 2 Type: Float Last value: - Permission: Read-Only Update policy: On change Last update: N/A

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

Dashboards > Dashboard Monitor... >

ADD
...

WIDGETS

Search widget or variable type
All
Interaction
Visualisation
Annotation

Messenger

LoRa

Uptime light

LoRa light

Time Picker

Scheduler

Value

Value Selector

Value Dropdown

Status

Gauge

Percentage

LED

Map

Chart

Advanced Map

Advanced Chart

Sticky Note

Image

Image Map

THINGS

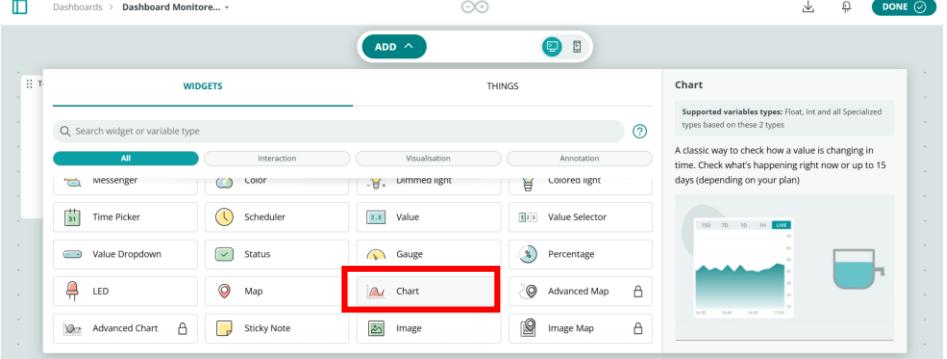
Supported variables types: Int, Float and all Specialized types based on these 2 types.

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.

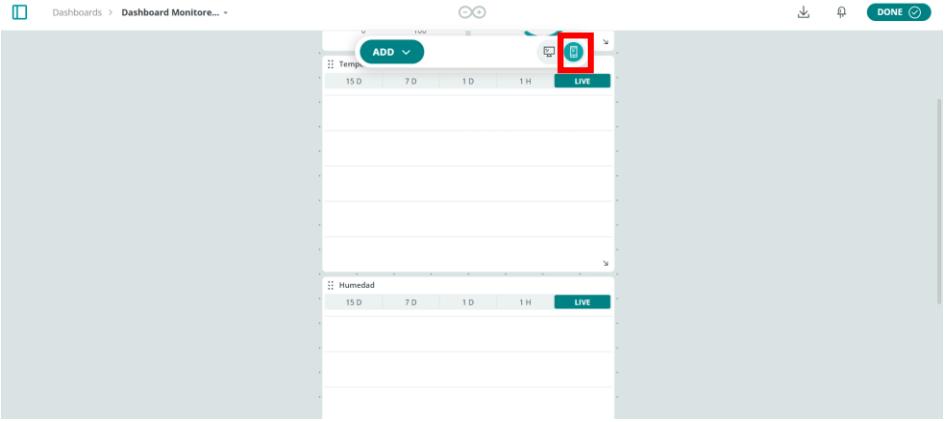
100%

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

Pág. 10 de 16



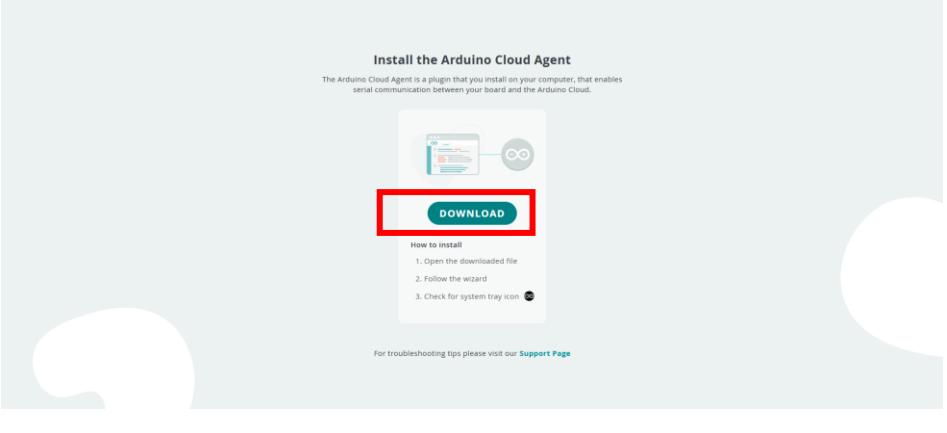
✓ Seleccionar la vista para dispositivos móviles y organizar las diferentes Widgets



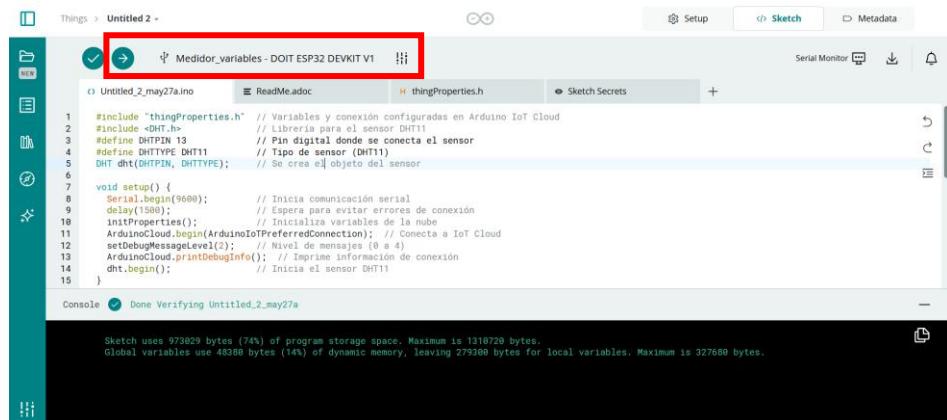
4. CARGA DEL PROGRAMA A LA TARJETA

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



The screenshot shows the Arduino IDE interface. The top bar has tabs for 'Setup', 'Sketch', and 'Metadata'. Below the tabs, there's a 'Serial Monitor' button. The main area shows a sketch named 'Untitled_2' with code for a DHT11 sensor connected to pin 13. The code includes includes for 'thingProperties.h', 'DHT.h', and 'DHT.h' for the DHT11 sensor. It sets up the serial port at 9600 bps, initializes properties, connects to the IoT Cloud, and prints debug information. The 'Sketch' tab is selected. The bottom console window shows the verification process completed.

```

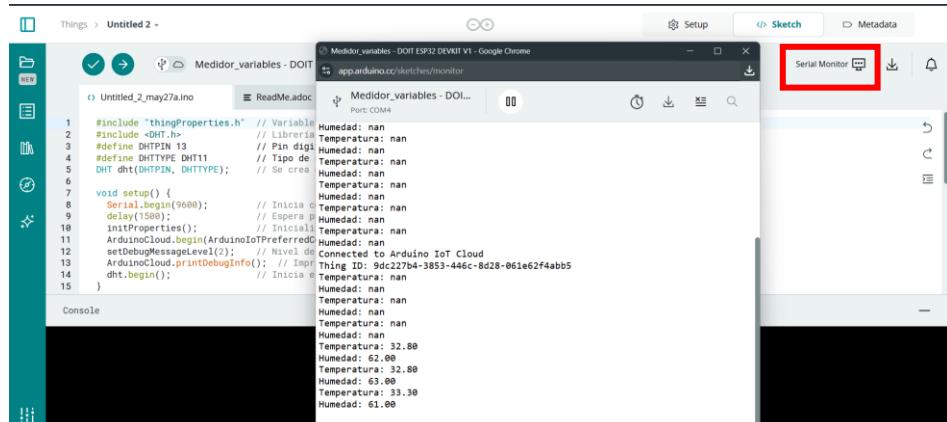
1 #include "thingProperties.h" // Variables y conexión configuradas en Arduino IoT Cloud
2 #include <DHT.h> // Librería para el sensor DHT11
3 #define DHTPIN 13 // Pin digital donde se conecta el sensor
4 #define DHTTYPE DHT11 // Tipo de sensor (DHT11)
5 DHT dht(DHTPIN, DHTTYPE); // Se crea el objeto del sensor
6
7 void setup() {
8   Serial.begin(9600); // Inicia comunicación serial
9   delay(1500); // Espera para evitar errores de conexión
10  initProperties(); // Inicializa variables de la nube
11  ArduinoCloud.begin(ArduinoIoTPreferredRegion()); // Conecta a IoT Cloud
12  setDebugMessageLevel(2); // Nivel de mensajes (0 a 4)
13  ArduinoCloud.printDebugInfo(); // Imprime información de conexión
14  dht.begin(); // Inicia el sensor DHT11
15 }

```

Console Done Verifying Untitled_2.may27a

Sketch uses 973029 bytes (74%) of program storage space. Maximum is 1310729 bytes.
Global variables use 48388 bytes (14%) of dynamic memory, leaving 279300 bytes for local variables. Maximum is 327680 bytes.

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



The screenshot shows the Arduino IDE with the 'Serial Monitor' tab selected. The main sketch window is identical to the previous one. The serial monitor window on the right shows the output of the code execution, displaying temperature and humidity readings from the DHT11 sensor over time.

```

Humedad: nan
Temperatura: nan
Humedad: nan
Temperatura: 32.88
Humedad: 62.00
Temperatura: 32.88
Humedad: 63.00
Temperatura: 33.30
Humedad: 61.00

```

- ✓ 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.

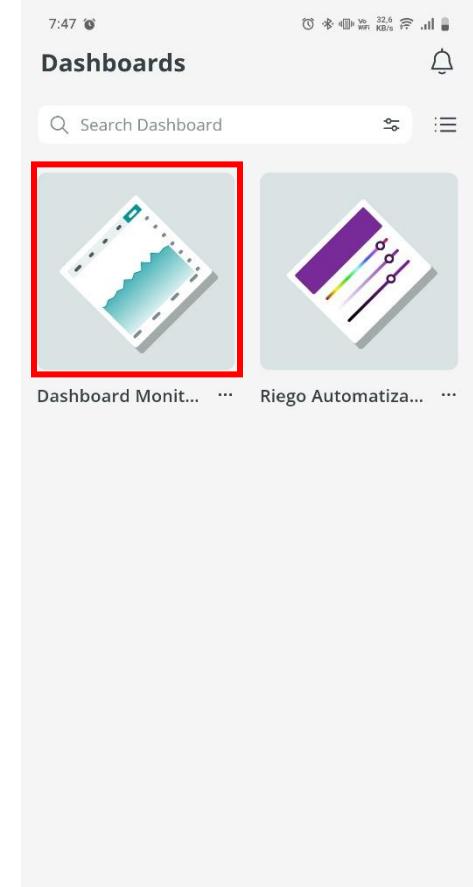
The dashboard displays four main sections:

- Temperatura:** A circular gauge showing 30.2 degrees Celsius.
- Humedad:** A circular gauge showing 65% relative humidity.
- Temperatura (Graph):** A line graph for the last hour showing temperature fluctuations between 30.25 and 30.75 degrees Celsius.
- Humedad (Graph):** A line graph for the last hour showing humidity fluctuations between 60 and 68%.

- ✓ 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.

The mobile app interface shows the "Welcome to Arduino" logo and a sign-in form with fields for "Email or Username" and "Password". It also includes links for "Forgot your password?", "SIGN IN", and "Create your account". Below the sign-in form, there are social media login options for Google, GitHub, and Facebook.

- ✓ 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

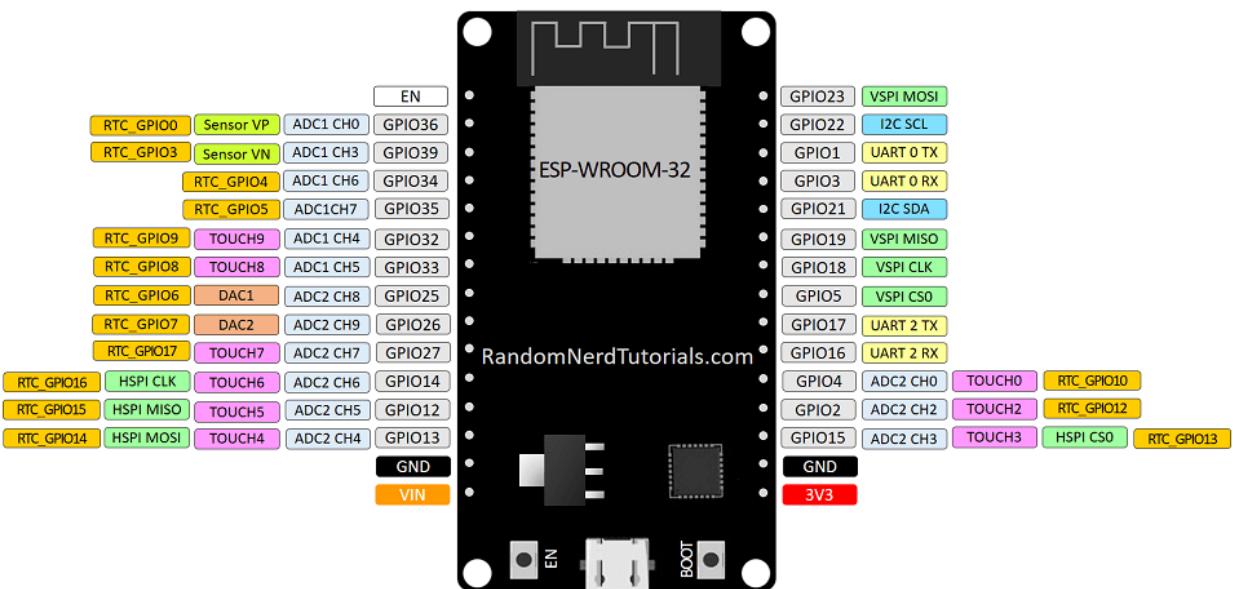


Imagen tomada de (Embedded Systems Design, 2025)