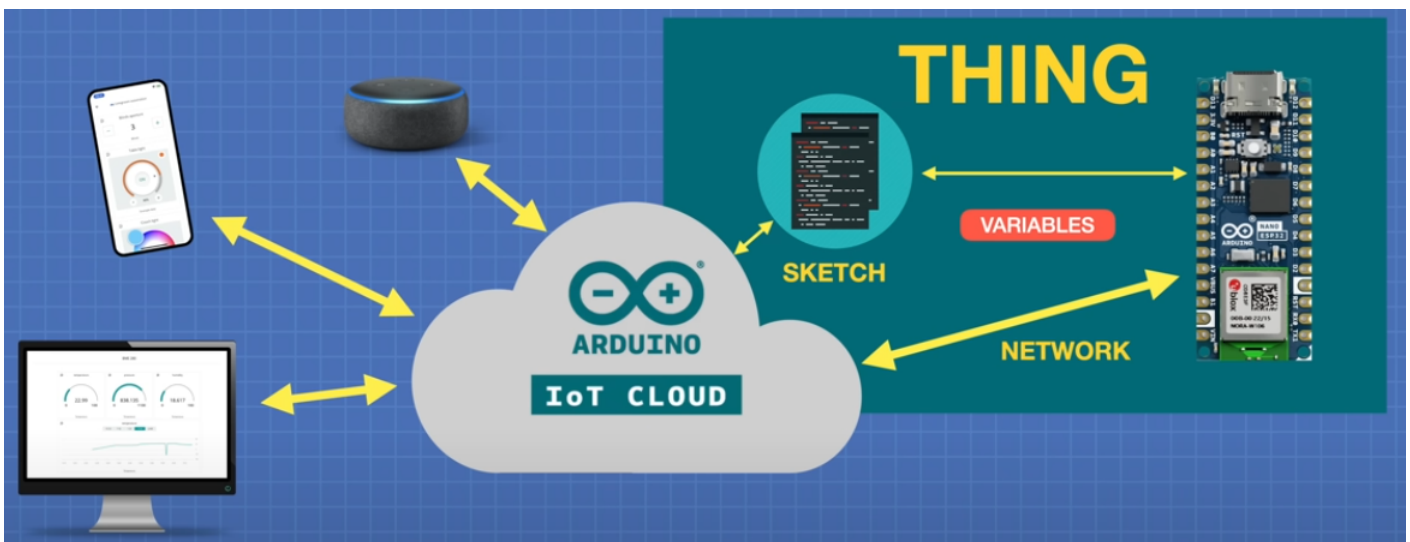


Arduino IDE con IoT: Arduino Cloud

Esta plataforma <https://docs.arduino.cc/arduino-cloud/> nos permite conectar nuestras placas (Arduino v4, ESP32, et...) con un panel de control **Dashboard** y así controlarlos a distancia por Internet.

El mecanismo es sencillo, el ESP32 conectado por internet, pasa variables a un código (Sketch), a este conjunto se le llama **Thing**, y este se lo comunica a **IoT CLOUD** y la plataforma lo comunica a los paneles de control. **Dashboard** que se puede ver desde el PC o desde el móvil. El proceso también funciona al revés.



Extraído de [Youtube Exploring the Arduino Nano ESP32](#)

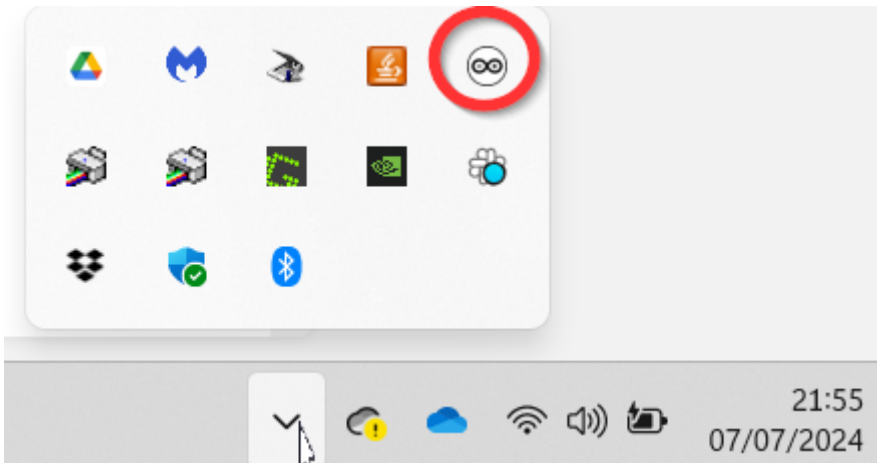
1. Creamos una cuenta en **Arduino Cloud**
2. Instalamos **Arduino Create Agent**
3. **Build the Thing** es decir preparamos nuestra placa ESP32 con el Sketch
 1. Creamos the device
 2. Creamos the thing
 3. Añadimos las variables
 4. Creamos el sketch y lo grabamos en el ESP32
4. Construimos un **Dashboard** o panel de control

PASO 1 LOGUEARSE EN ARDUINO CLOUD

En Plan permite una cuenta gratuita sólo se pueden 2 **things** ver <https://cloud.arduino.cc/plans>

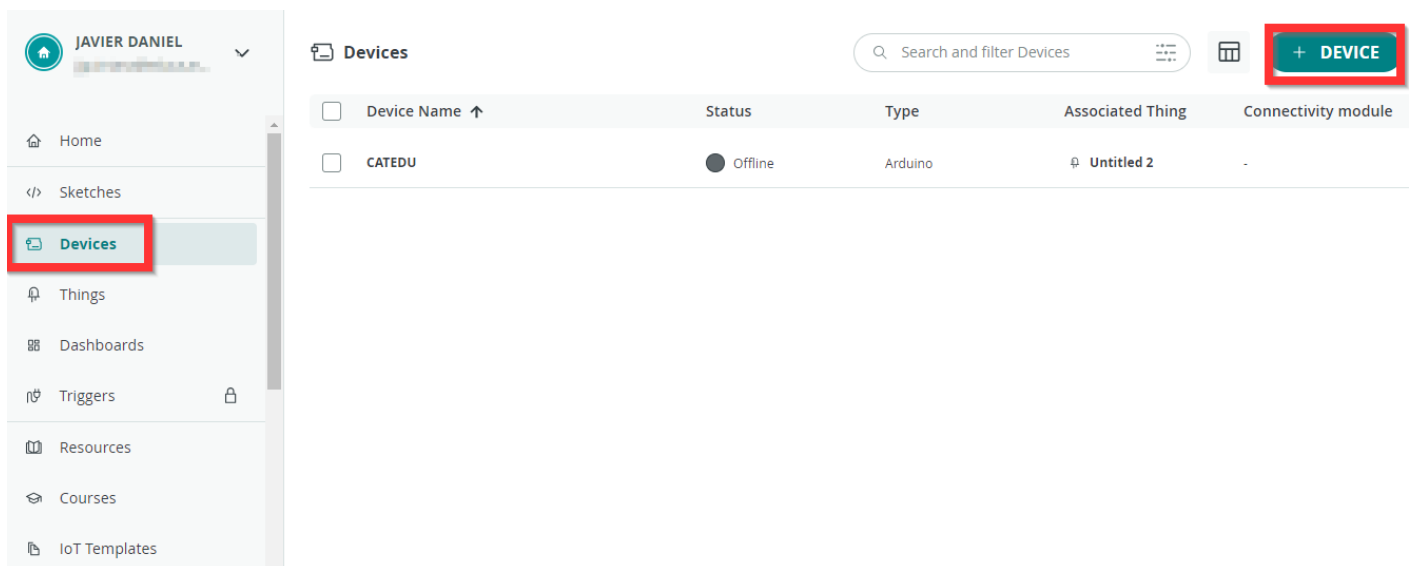
PASO 2 ADRUINO CREATE AGENT

Arduino Create Agent te lo puedes descargar desde <https://cloud.arduino.cc/download-agent>, se descarga, se ejecuta, hay que seguir los pasos, se queda en segundo plano en el PC y no tienes que preocuparte



PASO 3 Build the Thing: CREATE DEVICE

Primero añadimos un Device o placa en <https://app.arduino.cc/devices>



Elegimos placa Arduino

Setup Device



RECOMMENDED

AUTOMATIC

Auto-sketch generation, online editing, easy upload



Arduino board ⓘ

⌘ Arduino language (C++)



Third party device ⓘ

⌘ Arduino language (C++)

MANUAL

Manual programming and upload, offline editing



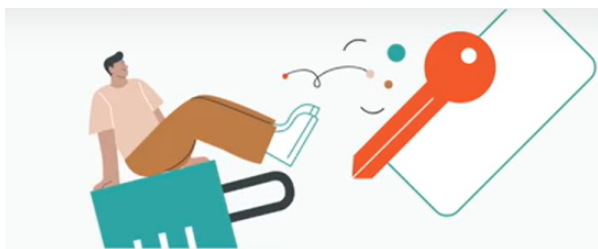
Any Device ⓘ

⌘ Python, MicroPython, JavaScript (NodeJS)

Si falla, ponemos la placa en modo Bootloader (ver qué es eso en

<https://libros.catedu.es/books/arduino-alvik/page/instalar-micropython>) y entonces detectará el puerto

Conectamos nuestro Arduino Alvik y saldrá un diálogo con un TOKEN on **Secret key** que lo guardaremos **ante todo no hacerlo público**



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

XXXXXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX

Secret Key



Secret key cannot be recovered

Please keep it safe, if you lose it you will have to delete and setup your device again.

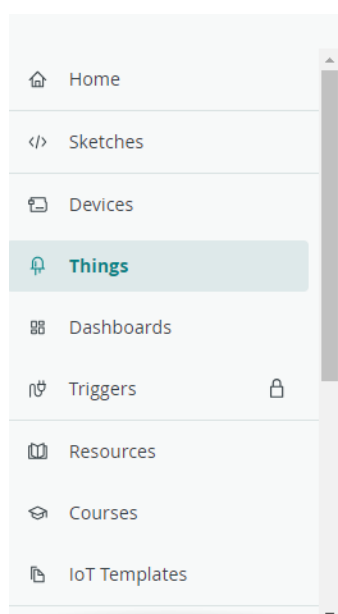


I saved my device ID and Secret Key

CONTINUE

PASO 3 Build the Thing: CREATE THING

Una vez creada la placa, nos vamos a Thing, crear



Follow these simple steps to create your first Thing on Arduino Cloud



Create a Thing

A Thing is a virtual entity that lets you link your physical device to the Cloud: it includes variables, sketch and metadata



Associate Device and Network

Select the Device you want to use and enter your network credentials, so you can send and receive data remotely

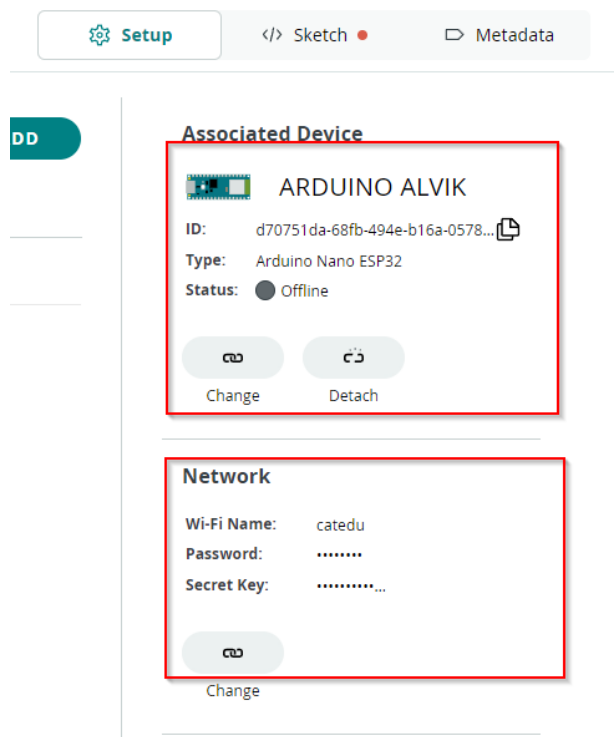


Start creating!

Easily add Cloud Variables that will be automatically included in the sketch, and that are used to exchange data with the Cloud

+ CREATE THING

Asociamos el Thing al Device, y le configuramos una red wifi (te predirá el Secret Key)



PASO 3 Build the Thing: CREATE THING-VARIABLES

Luego añadimos variables, por ejemplo RGBverde que va a encender y apagar la luz verde, va a ser tipo Bool y Read&Write

RGBverde

Declaration

`bool` rGBverde

Type

Boolean

Variable Permission

Read & Write

Update Policy

On change

ID

72371e05-f94d-487a-bb77-f20b97f4dc18



Last Value


false


Last Update


11 Jul 2024 12:12:50


PASO 3 Build the Thing: CREATE THING-SKETCH

Dentro de Thinks nos vamos a SKETCH

 Things > micosa-alkvik ▾

 Setup

 Sketch

 Metadata

Cloud Variables

ADD

Name ▾	Last Val...	Last Update
 RGBverde bool rGBverde;	false	11 Jul 2024 12:12:50

Associated Device

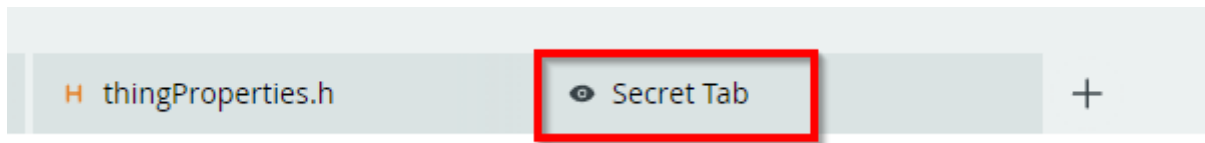
 **ARDUINO ALVIK**

ID: d70751da-68fb-494e-b16a-0578...

Type: Arduino Nano ESP32

Status:  Offline

y vemos que ha creado un código ***thingProperties.h*** que tiene que tener el SSID de la wifi, su contraseña y la palabra clave de nuestro ESP32, podemos ponerlo manualmente o nos fijamos y en **Secret Tab** estan ya puestos :

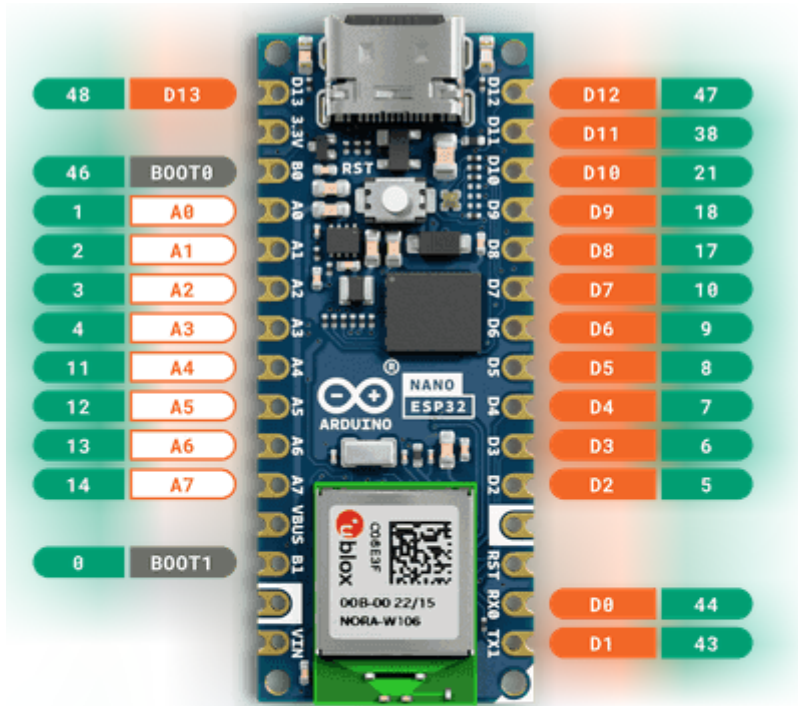


El otro script es el nombre que hemos creado en Thing y vemos que :

- LINEA 9 Esta declarada la variable que hemos añadido
- LINEA 16 Incluye la librería ***thingProperties.h***
- *LINEA 41* Añadimos en setup() la declaración que D13 SERÁ SALIDA **pinMode(D13,OUTPUT);**
- LINEA 60 AL 66 Añadimos en **onRGBverdeChange** una condicional, de tal manera que si la variable es cierta, que encienda el led y si es falsa que lo apague

¿Por qué es D13? ¿NO TENDRÍA QUE SER 48?

Eso ya lo hemos visto en <https://libros.catedu.es/books/arduino-alvik/page/parpadeo-led-esp32>



i ESP32 pin numbers

i Nano pin numbers

Fuente <https://docs.arduino.cc/tutorials/alvik/user-manual/>

```
/*
Sketch generated by the Arduino IoT Cloud Thing "Untitled"
https://create.arduino.cc/cloud/things/34a0aae1-c7b9-42ab-92d4-0e37bd51031f

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when changes are made to the
Thing

bool rGBverde;
```


Variables which are marked as READ/WRITE in the Cloud Thing will also have functions which are called when their values are changed from the Dashboard.

These functions are generated with the Thing and added at the end of this sketch.

```
*/
```

```
#include "thingProperties.h"
```

```
void setup() {
```

```
  // Initialize serial and wait for port to open:
```

```
  Serial.begin(9600);
```

```
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
```

```
  delay(1500);
```

```
  // Defined in thingProperties.h
```

```
  initProperties();
```

```
  // Connect to Arduino IoT Cloud
```

```
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
```

```
  /*
```

```
    The following function allows you to obtain more information
    related to the state of network and IoT Cloud connection and errors
    the higher number the more granular information you'll get.
```

```
    The default is 0 (only errors).
```

```
    Maximum is 4
```

```
  */
```

```
  setDebugMessageLevel(2);
```

```
  ArduinoCloud.printDebugInfo();
```

```
  /// MI CODIGO
```

```
  pinMode(D13,OUTPUT);
```

```
}
```

```
void loop() {
```

```
  ArduinoCloud.update();
```

```
// Your code here

}

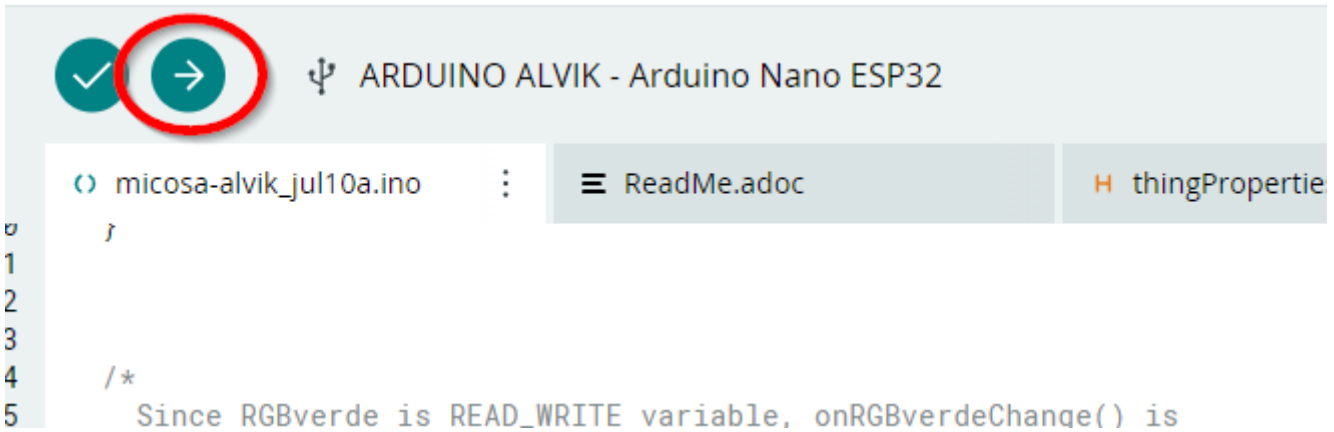
/*
  Since RGBverde is READ_WRITE variable, onRGBverdeChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onRGBverdeChange() {
  // Add your code here to act upon RGBverde change
  if (rGBverde){
    digitalWrite(D13,HIGH);

  }else{
    digitalWrite(D13,LOW);

  }
}

/*
  Since RGBrojo is READ_WRITE variable, onRGBrojoChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onRGBrojoChange() {
  // Add your code here to act upon RGBrojo change
}
```

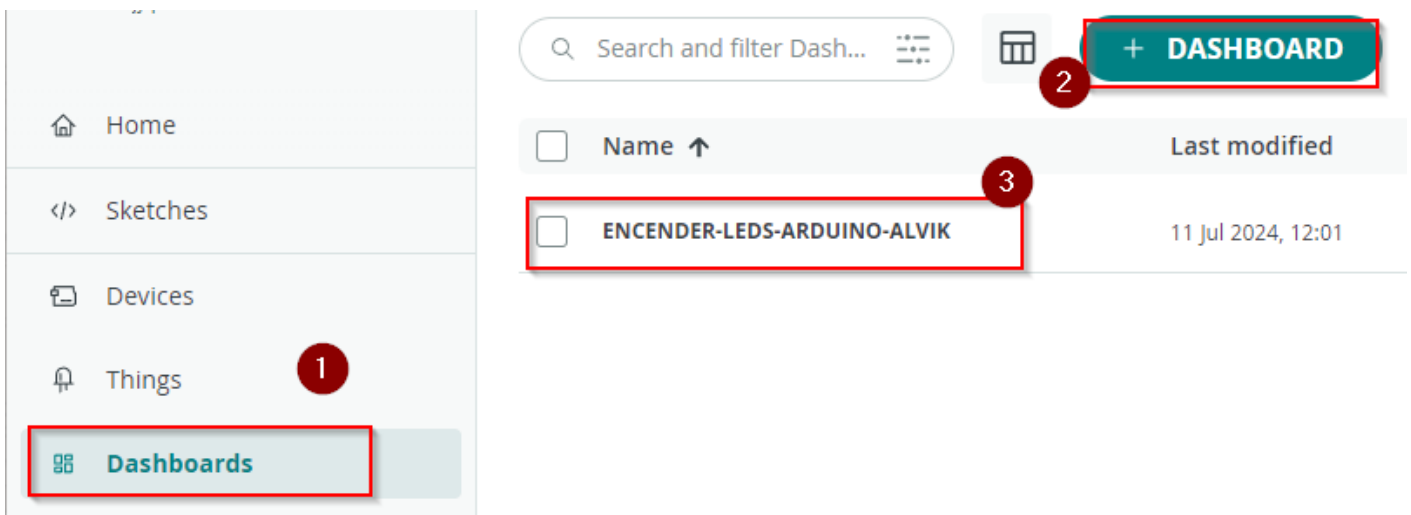
Lo subimos



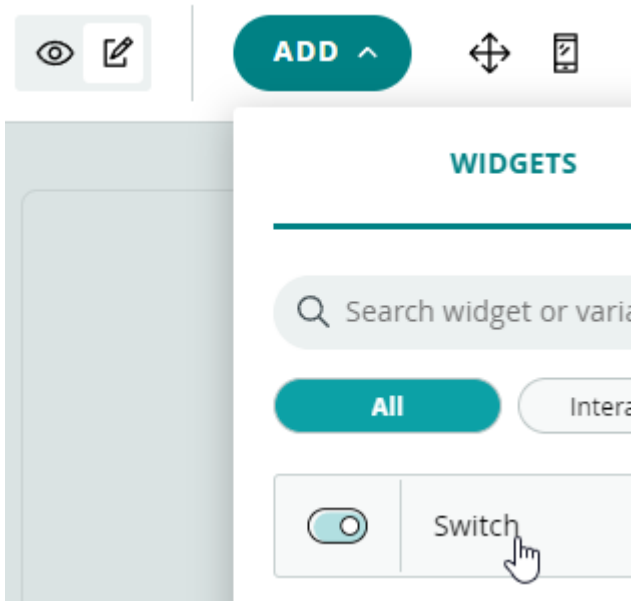
Ojo, tienes que tener el Arduino **Arduino Create Agent paso 2**

PASO 4 Dashboard

Creamos un panel de control



Y le añadimos un Switch asociado a la variable RGBverde



Podemos ver el dashboard en un teléfono móvil instalando la [APP Arduino IoT Cloud Remote](#)

Arduino IoT Cloud Remote

Arduino

La aplicación Arduino para controlar sus proyectos de Internet de las cosas desde cualquier lugar.

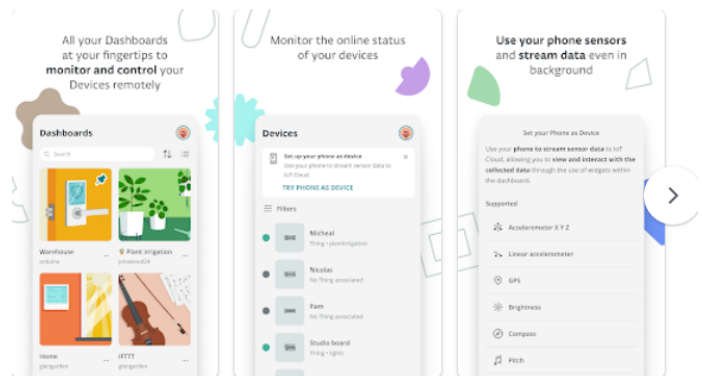


3,8★
992 reseñas

100 mil+
Descargas

PEGI 3

Descargar



Al loguearse con tu cuenta, ya nos aparece el Dashboard

Resultado

<https://www.youtube.com/embed/YDVMYbJtWUU>

Revision #10

Created 7 July 2024 21:19:15 by Javier Quintana

Updated 27 March 2025 23:07:32 by Javier Quintana