



Elvi Mihai Sabau Sabau<sup>[51254875L]</sup>

<sup>1</sup> Universidad de Alicante, Alicante, España.  
[emss5@alu.ua.es](mailto:emss5@alu.ua.es)

**Abstract.** En esta práctica conectaremos nuestro dispositivo a un servicio Cloud, en este caso Arduino IoT Cloud, y rastreamos una variable del dispositivo de tal manera que obtendremos dichos cambios en vivo en la capa Cloud.

**Keywords:** IoT, Arduino, Cloud Computing, IoT Cloud.

<b>Practica 3.</b>	<b>3</b>
Descripción.	3
Pasos.	3
Código.	3
Ejecución.	3
<b>Conclusiones.</b>	<b>3</b>

## 1 Práctica 3.

### 1.1 Descripción.

En esta práctica deberemos conectar nuestro dispositivo a una capa cloud, en concreto Arduino Cloud, para ello nos dirigiremos a <https://docs.arduino.cc/cloud/iot-cloud> y procederemos a crearnos una cuenta.

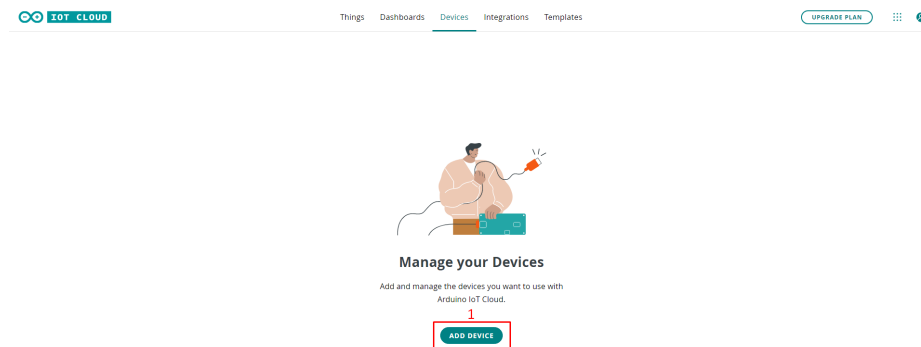
Una vez hecho esto, enlazaremos nuestro dispositivo desde <https://create.arduino.cc/iot/devices>, crearemos una variable rastreada llamada “temperatura” de tipo entero, y desde el editor de código web, editaremos el sketch que viene por defecto para crear una función que cambiará el valor de dicha variable a un número aleatorio entre 0 y 40 cada 3 segundos, y desde el Dashboard observaremos dicho cambio.

Además realizaremos un gráfico desde el panel de control de Arduino Cloud para mostrar una gráfica de dichos datos, y usando la app móvil de Arduino IoT Cloud observaremos dichos cambios desde nuestro smartphone.

### 1.2 Pasos.

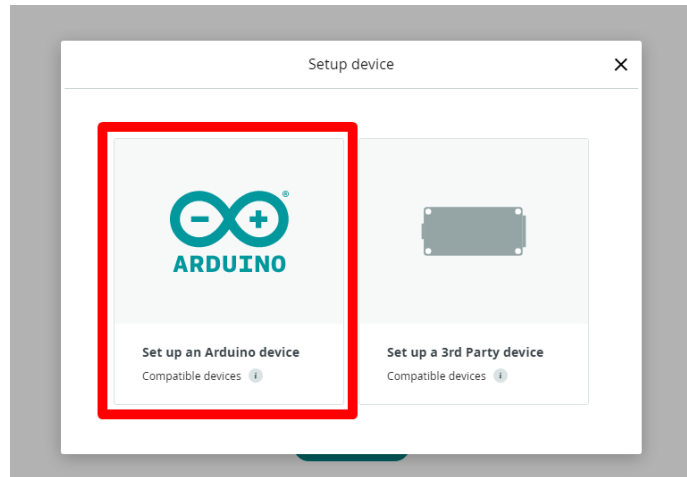
Primero accederemos a <https://docs.arduino.cc/cloud/iot-cloud> y crearemos nuestra cuenta cloud.

Después accederemos a <https://create.arduino.cc/iot/devices> y añadimos un nuevo dispositivo.



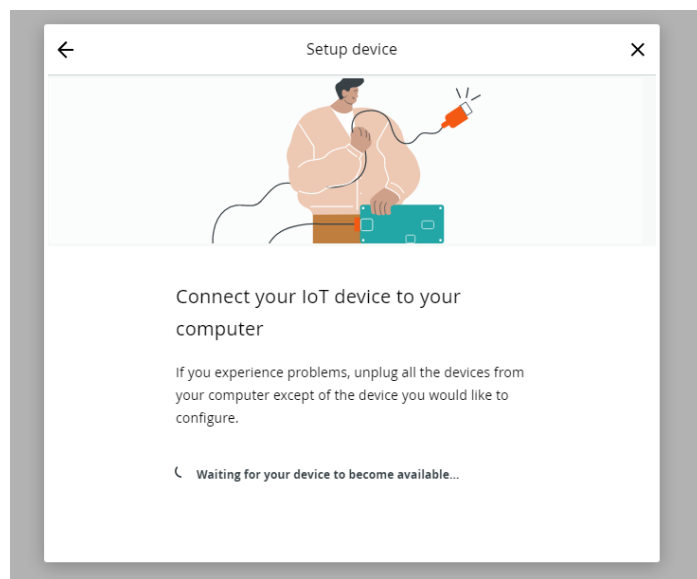
**Fig. 1** Captura de pantalla de la página web de [create.arduino.cc/iot/devices](https://create.arduino.cc/iot/devices), en rojo está enmarcado el botón para añadir un nuevo dispositivo.

Y seleccionamos que queremos añadir un dispositivo Arduino.



**Fig. 2** Captura de pantalla de la página web de [create.arduino.cc/iot/devices](https://create.arduino.cc/iot/devices), en rojo se muestra el botón a pulsar.

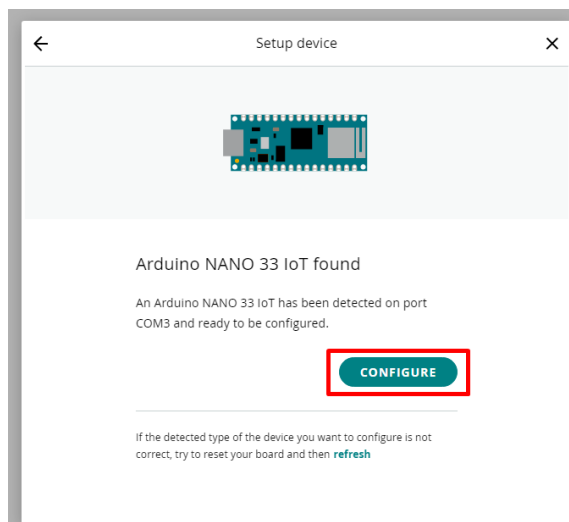
El navegador intentará conectar con nuestro Agente de Arduino, si no lo tenemos instalado, nos pedirá que nos lo instalemos.



**Fig. 3** Captura de pantalla de la página web de [create.arduino.cc/iot/devices](https://create.arduino.cc/iot/devices), se muestra la página intentando conectar con Arduino Agent.

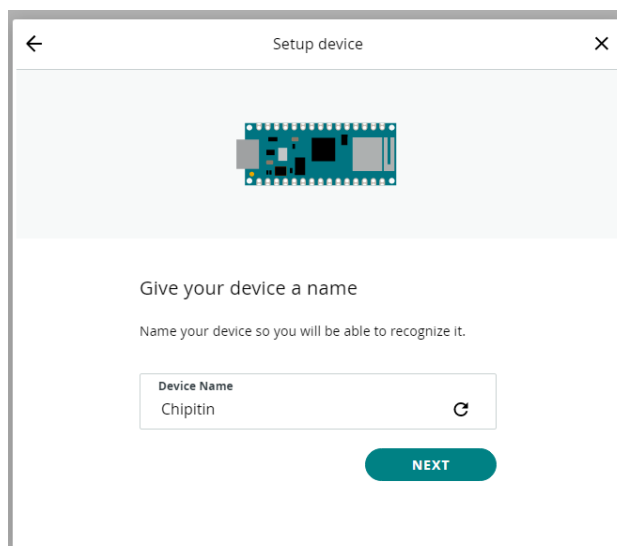
Una vez que el navegador establezca conexión con el agente, se nos mostrarán los dispositivos detectados.

Seleccionamos el dispositivo a configurar, y pulsamos en “Configurar”.



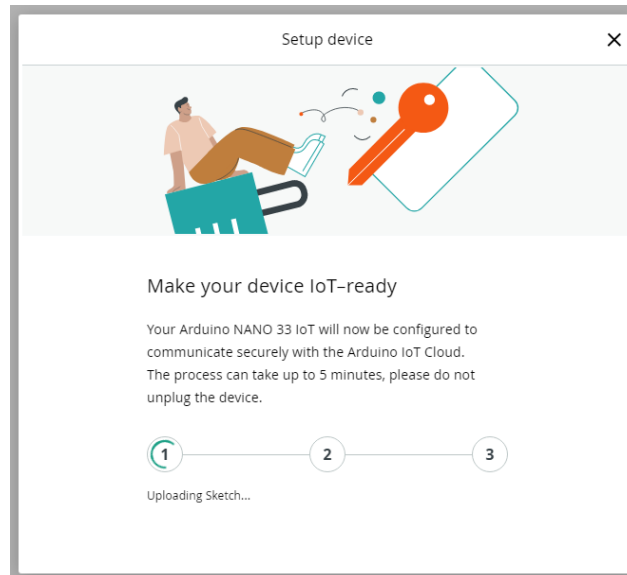
**Fig. 4** Captura de pantalla de la página web de [create.arduino.cc/iot/devices](https://create.arduino.cc/iot/devices), se muestra la lista de dispositivos detectados, listos para configurar.

Una vez iniciada la configuración nos pedirá nombrar el dispositivo, en mi caso lo he llamado “Chipitin”, diminutivo de “Chip”.



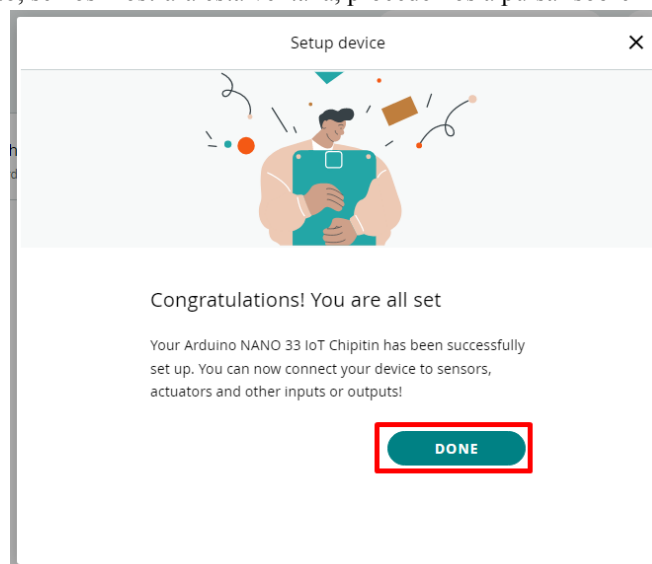
**Fig. 5** Captura de pantalla de la página web de [create.arduino.cc/iot/devices](https://create.arduino.cc/iot/devices), se muestra el formulario para asignar un nombre al dispositivo.

Una vez hecho esto, el agente procederá a cargar el sketch por defecto en el dispositivo, y prepararlo para ser usado desde el panel Cloud.



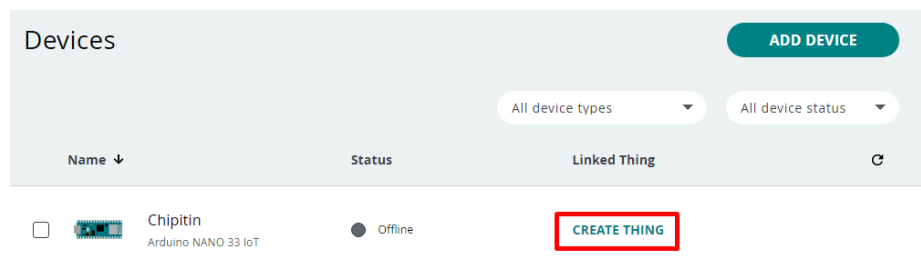
**Fig. 6** Captura de pantalla de la página web de [create.arduino.cc/iot/devices](https://create.arduino.cc/iot/devices), se muestra los pasos que el agente está ejecutando para configurar nuestro dispositivo.

Una vez listo, se nos mostrará esta ventana, procedemos a pulsar sobre “done”:



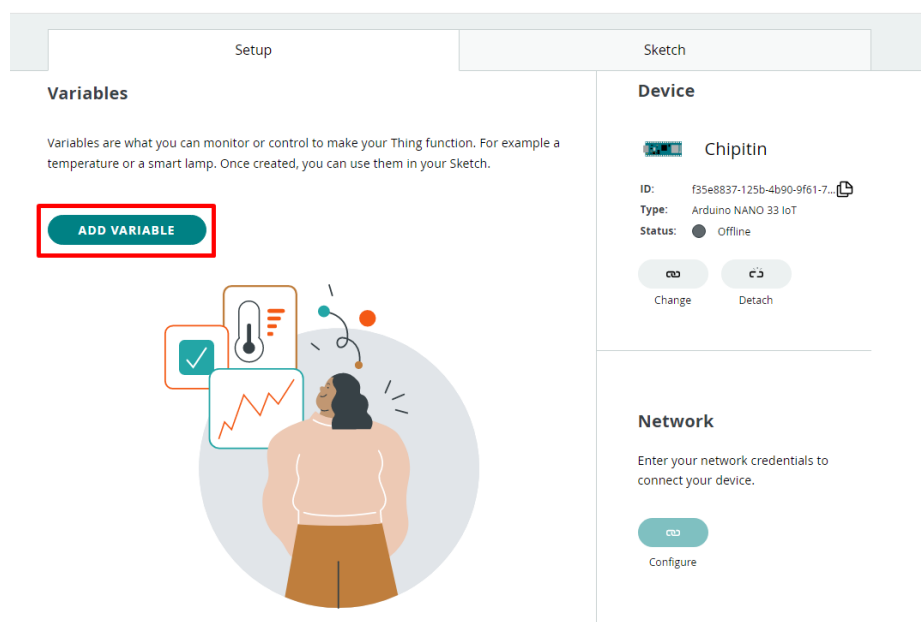
**Fig. 7** Captura de pantalla de la página web de [create.arduino.cc/iot/devices](https://create.arduino.cc/iot/devices), pantalla que muestra la configuración completada.

Y ahora, al cerrar el pop-up podremos ver que nuestro dispositivo se muestra en la lista de dispositivos, pulsamos sobre “create thing”.



**Fig. 8** Captura de pantalla de la página web de [create.arduino.cc/iot/devices](https://create.arduino.cc/iot/devices), pantalla que muestra la lista de dispositivos.

Ahora, desde esta ventana, pulsamos sobre “add variable”.



**Fig. 9** Captura de pantalla de la página web de [create.arduino.cc/iot/things](https://create.arduino.cc/iot/things), pantalla que muestra el dashboard de nuestro dispositivo.

Y ahora, completamos la especificación de la variable con los siguientes valores.

Add variable

×

Name

temperatura

↺

Sync with other Things

i

Integer Number

eg. 1

▼

Declaration

`int temperatura ;`

i

Variable Permission

i

☒ Read & Write

☐ Read Only

Variable Update Policy

i

☒ On change

☐ Periodically

Threshold

0

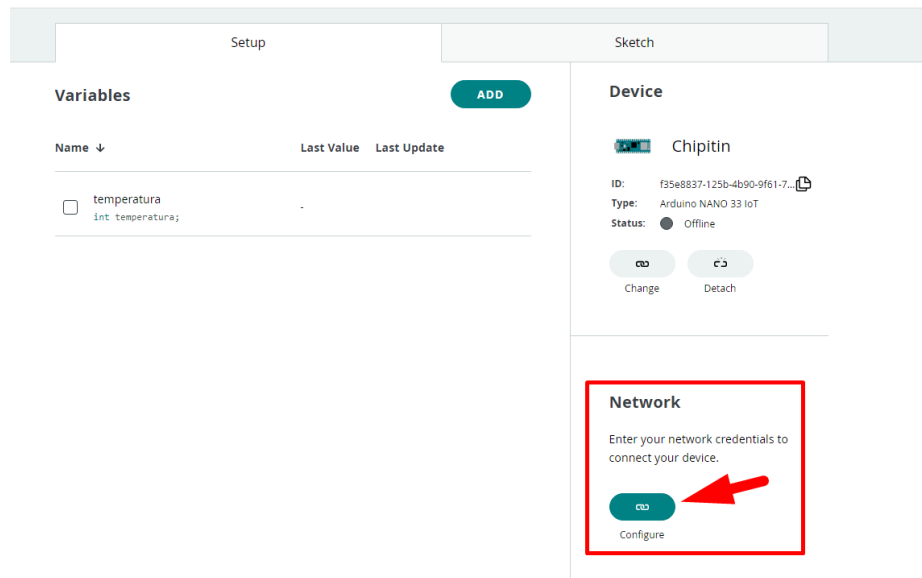
ADD VARIABLE

CANCEL

**Fig. 10** Captura de pantalla de la página web de [create.arduino.cc/iot/things](https://create.arduino.cc/iot/things), pantalla que muestra el formulario de creación de una variable rastreada.

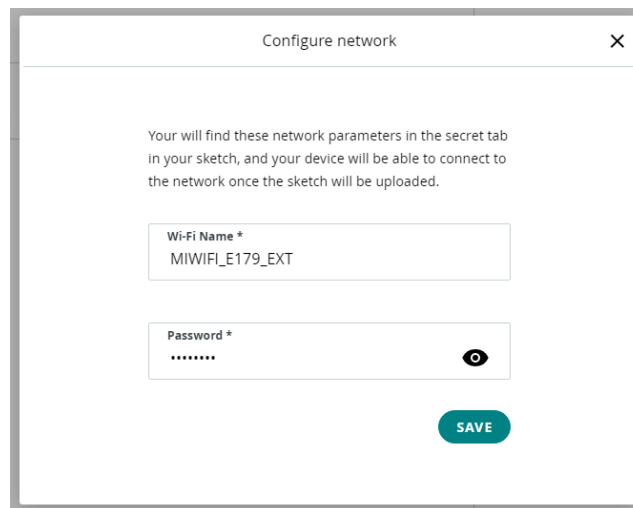


Ahora, podemos ver el estado de nuestra variable a la izquierda. Ahora vamos a conectar nuestro dispositivo a internet, para ello pulsamos en el botón “Configure” de la sección “Network”..



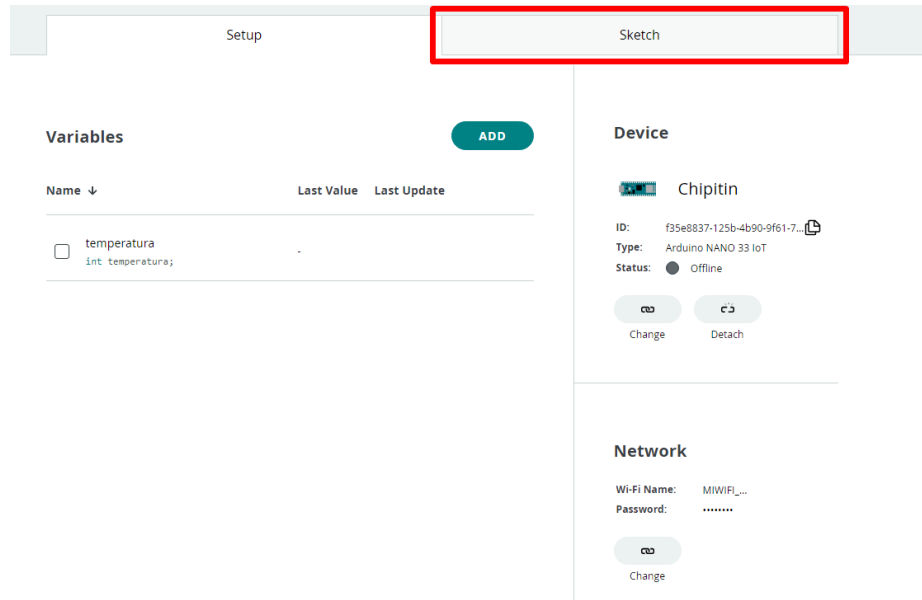
**Fig. 11** Captura de pantalla de la página web de [create.arduino.cc/iot/things](https://create.arduino.cc/iot/things), pantalla que muestra la variable recién creada, y el botón de configuración de red.

Rellenamos el formulario con las credenciales de nuestra red WiFi.



**Fig. 12** Captura de pantalla de la página web de [create.arduino.cc/iot/things](https://create.arduino.cc/iot/things), pantalla que muestra el formulario de autenticación de red del dispositivo.

Una vez configurada la red, nos movemos al editor en línea.



**Fig. 13** Captura de pantalla de la página web de [create.arduino.cc/iot/things](https://create.arduino.cc/iot/things) , pantalla que muestra el botón de acceso al editor en línea.

Ahora dentro del sketch, podemos empezar a tocar el código.

### 1.3 Código.

El código que se genera por defecto es necesario para poder conectarse con los servicios que proporciona nuestro Cloud.

Respecto a la variable “temperatura”, podemos denotar que no se define en ningún sitio, pero si nos fijamos bien, los comentarios predefinidos nos dicen:

“Las siguientes variables se generan y actualizan automáticamente cuando se realizan cambios en el proyecto.

```
int temperatura;
```

Las variables que están marcadas como READ/WRITE en Cloud también tendrán funciones que se llaman cuando se modifican sus valores desde el Dashboard. Estas funciones se generan con el proyecto y se agregan al final de este boceto.”

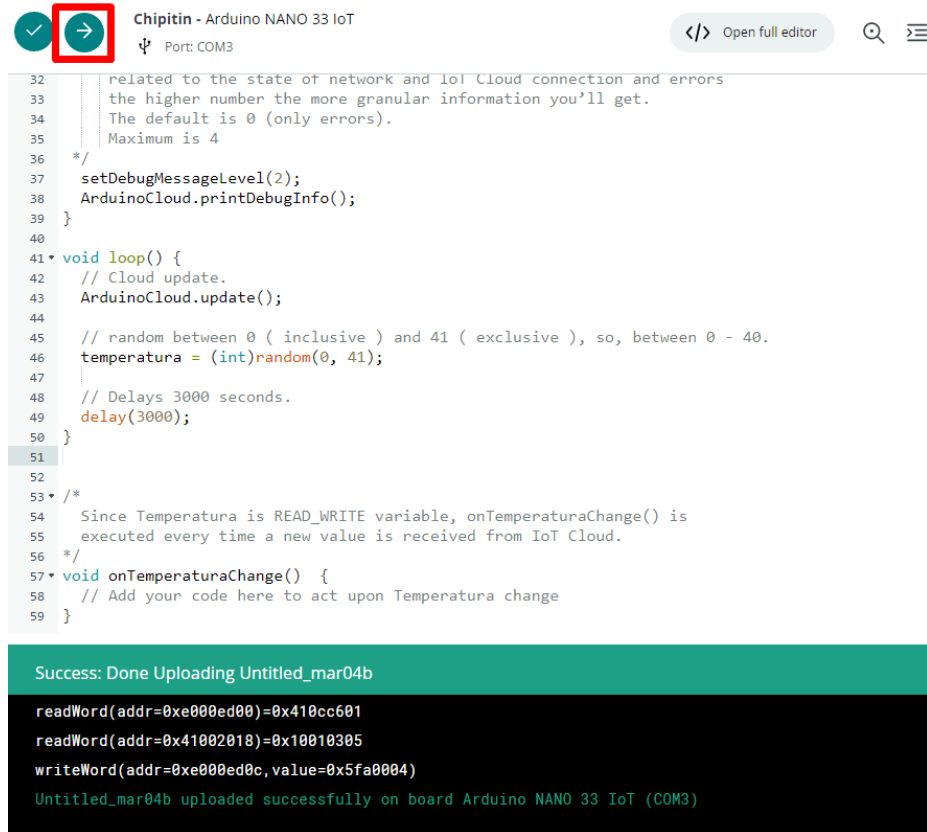
Así que, no debemos preocuparnos por definirla, además, podemos observar que nos viene por defecto otra función que nos permite realizar acciones cuando dicha variable cambia de estado.

De momento, lo que haremos será asignarle un valor del 0 al 40, y esperar 3 segundos, tal y como nos lo pide la práctica, y lo haremos dentro de la función “loop” debajo de “ArduinoCloud.update();” para que los cambios a dicha variable puedan ser correctamente rastreables.

```
void loop() {  
    // Cloud update.  
    ArduinoCloud.update();  
  
    // random between 0 ( inclusive ) and 41 ( exclusive ), so,  
    between 0 - 40.  
    temperatura = (int)random(0, 41);  
  
    // Delays 3000 seconds.  
    delay(3000);  
}
```

**Código. 1** Muestra la fracción de código de “loop” asociado a la práctica, además de la función ArduinoCloud.update que actualiza en la nube el estado de las variables.

Una vez modificado el código del sketch, apretamos el botón de verificar y subir.



```
32      related to the state of network and IoT Cloud connection and errors
33      the higher number the more granular information you'll get.
34      The default is 0 (only errors).
35      Maximum is 4
36  */
37  setDebugMessageLevel(2);
38  ArduinoCloud.printDebugInfo();
39  }
40
41  void loop() {
42    // Cloud update.
43    ArduinoCloud.update();
44
45    // random between 0 ( inclusive ) and 41 ( exclusive ), so, between 0 - 40.
46    temperatura = (int)random(0, 41);
47
48    // Delays 3000 seconds.
49    delay(3000);
50  }
51
52
53  /*
54   Since Temperatura is READ_WRITE variable, onTemperaturaChange() is
55   executed every time a new value is received from IoT Cloud.
56  */
57  void onTemperaturaChange() {
58    // Add your code here to act upon Temperatura change
59  }
```

Success: Done Uploading Untitled\_mar04b

```
readWord(addr=0xe000ed00)=0x410cc601
readWord(addr=0x41002018)=0x10010305
writeWord(addr=0xe000ed0c,value=0x5fa0004)
Untitled_mar04b uploaded successfully on board Arduino NANO 33 IoT (COM3)
```

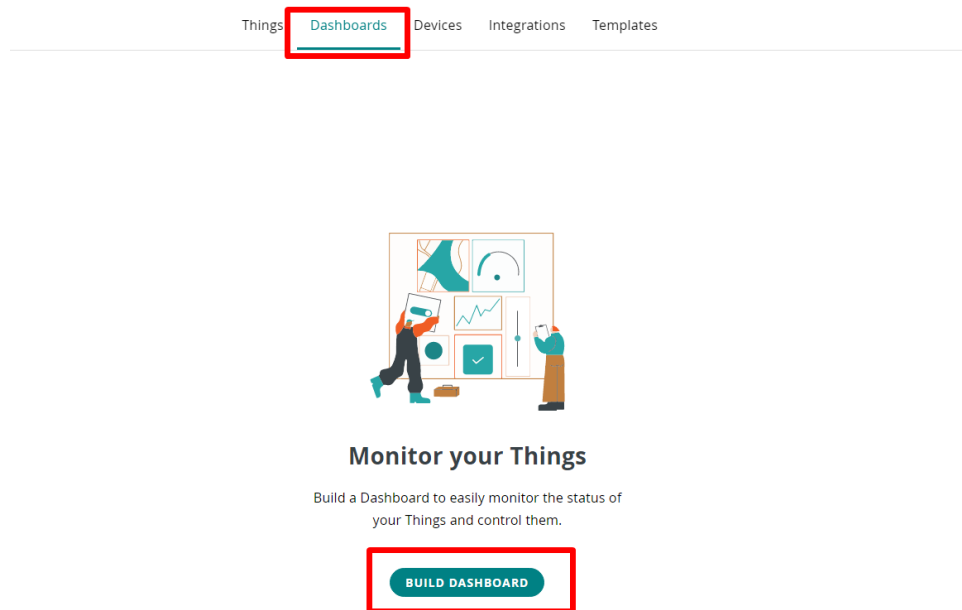
**Fig. 14** Captura de pantalla de la página web de [create.arduino.cc/iot/things/](https://create.arduino.cc/iot/things/) , pantalla que muestra el editor en línea “sketch” (garabato) recalcando el botón de “verificar y subir”.

Esto verifica, compila el programa y subirá el código al dispositivo, incluyendo la configuración de red previamente realizada.

El estado del dispositivo pasará a “online” cuando hayamos cargado el programa al dispositivo.

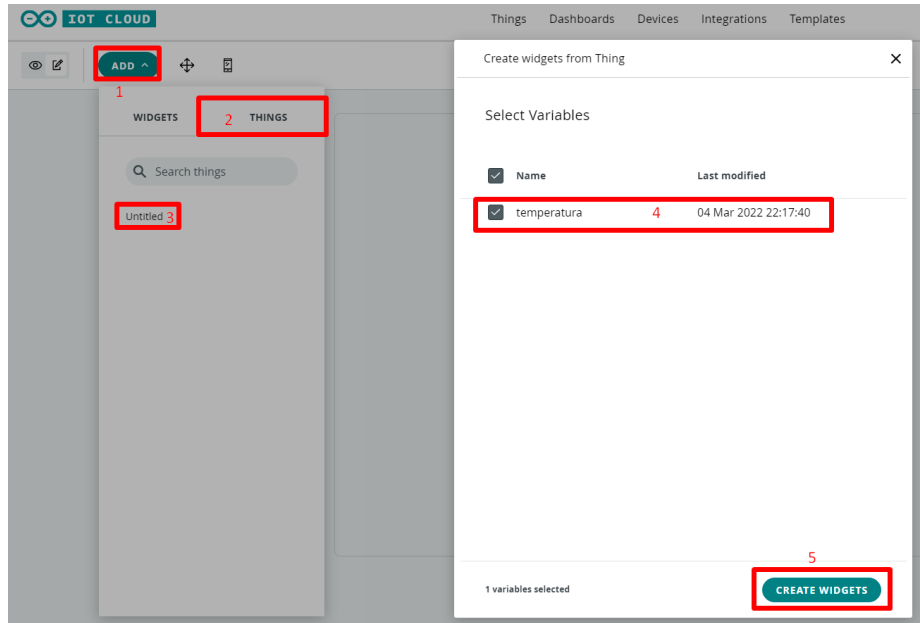
#### 1.4 Dashboard.

Ahora, vamos al apartado “Dashboard” y procederemos a crear una pequeña interfaz para representar el valor de la variable “temperatura” de forma discreta.



**Fig. 15** Captura de pantalla de la página web de [create.arduino.cc/iot/dashboards](https://create.arduino.cc/iot/dashboards), pantalla que muestra la página de Dashboards vacía, resaltando el botón de “crear dashboard”.

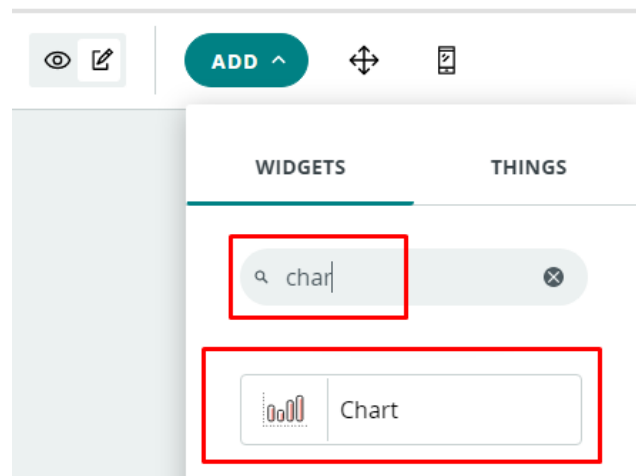
En mi caso, crearé un widget automático, enlazado a la variable del proyecto creado para Chipitin, “temperatura”, primero le damos a “add” > “things” > “Untitled ( o nombre del proyecto que se haya definido ) > Seleccionamos la variable > “Create Widget”.



**Fig. 16** Captura de pantalla de la página web de [create.arduino.cc/iot/dashboards](https://create.arduino.cc/iot/dashboards), pantalla que muestra los pasos a seguir para crear un widget en el dashboard.

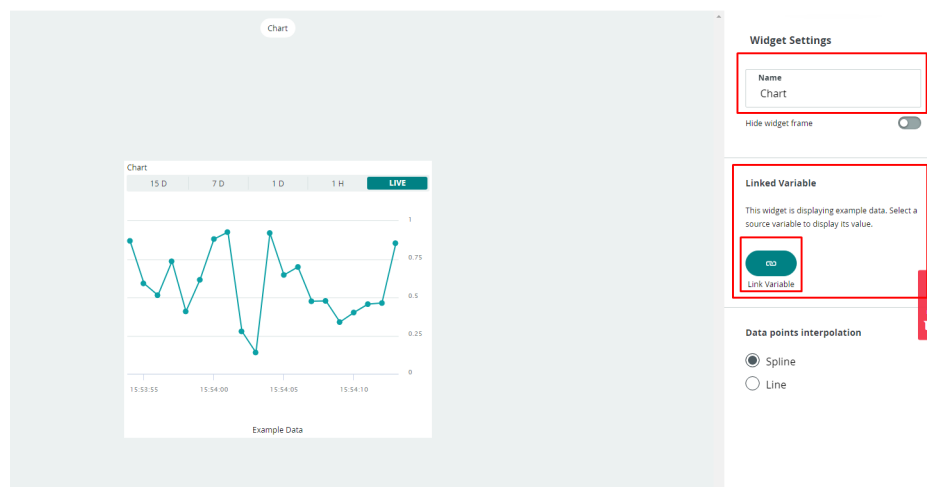
Con esto habríamos añadido un widget de la variable, que se va actualizando al momento que recibe los datos del arduino.

Ahora vamos a crear un chart, para ello, apretamos otra vez en el botón add, y buscamos “charts”.



**Fig. 17** Captura de pantalla de la página web de [create.arduino.cc/iot/dashboards](https://create.arduino.cc/iot/dashboards), pantalla que muestra los pasos a seguir para crear un chart en el dashboard.

Enlazamos el chart a nuestra variable, para ello apretamos el botón “add” en la sección “Linke variable”.

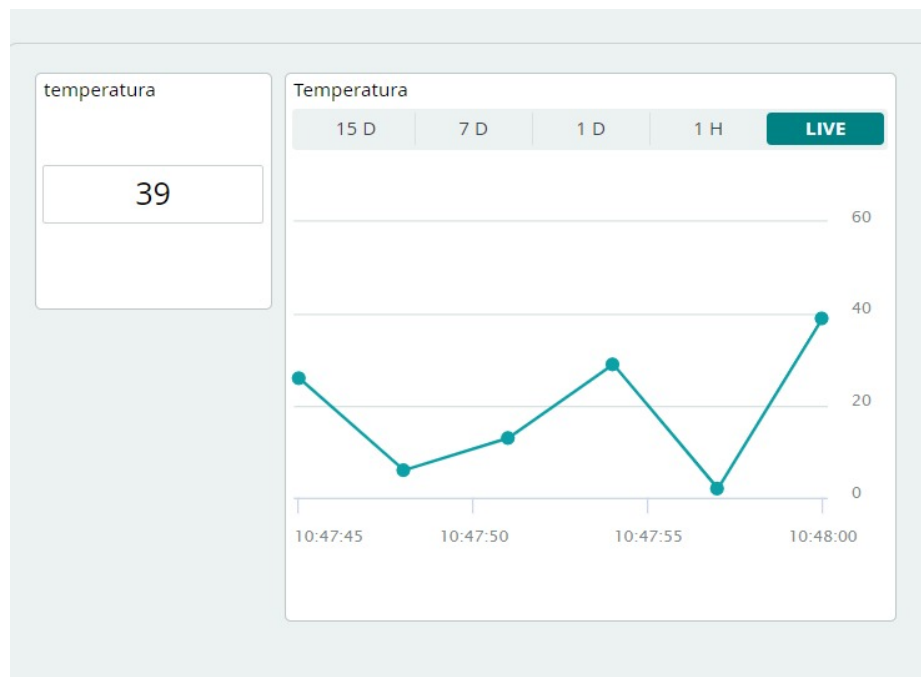


**Fig. 18** Captura de pantalla de la página web de [create.arduino.cc/iot/dashboards](https://create.arduino.cc/iot/dashboards), pantalla que muestra los pasos a seguir para crear un chart en el dashboard.

Seleccionamos nuestra variable, y listo!



**Fig. 19** Captura de pantalla de la página web de [create.arduino.cc/iot/dashboards](https://create.arduino.cc/iot/dashboards), pantalla que muestra los pasos a seguir para crear un chart en el dashboard.



**Fig. 20** Captura de pantalla de la página web de [create.arduino.cc/iot/dashboards](https://create.arduino.cc/iot/dashboards), pantalla que muestra los pasos a seguir para crear un chart en el dashboard.



Desde nuestra app móvil podemos acceder a las dashboards que hemos creado desde el panel web. Para ello, primero instalaremos la aplicación “Arduino IoT Cloud Remote”

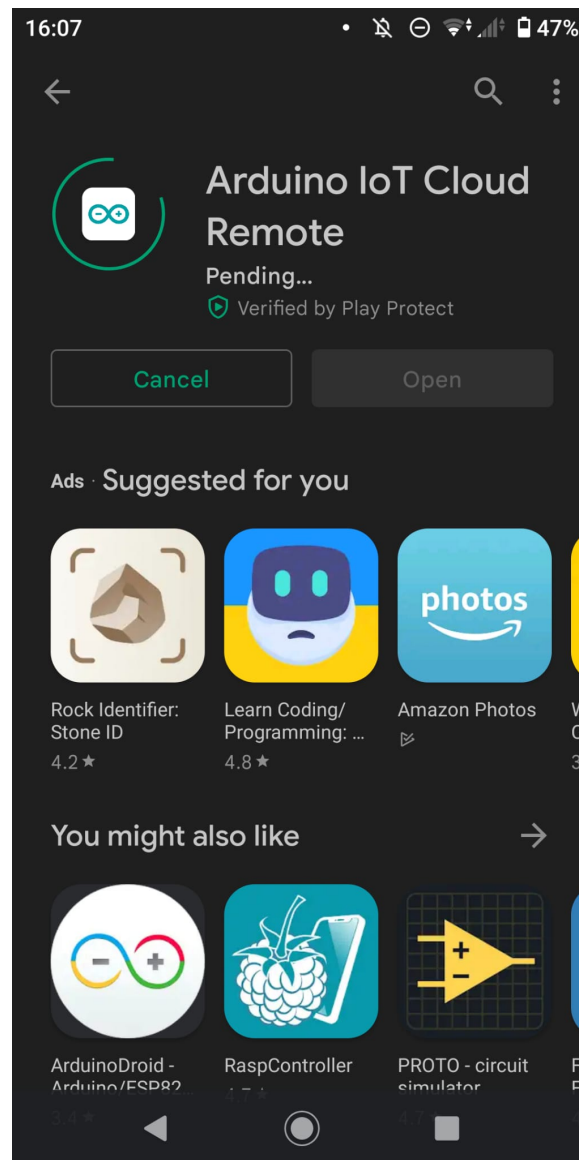
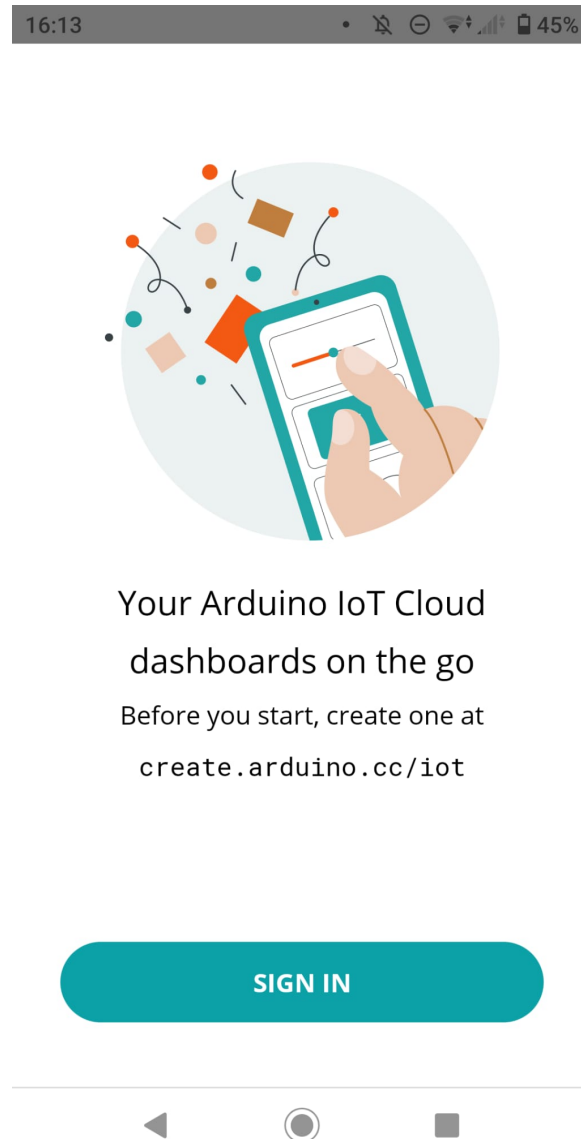
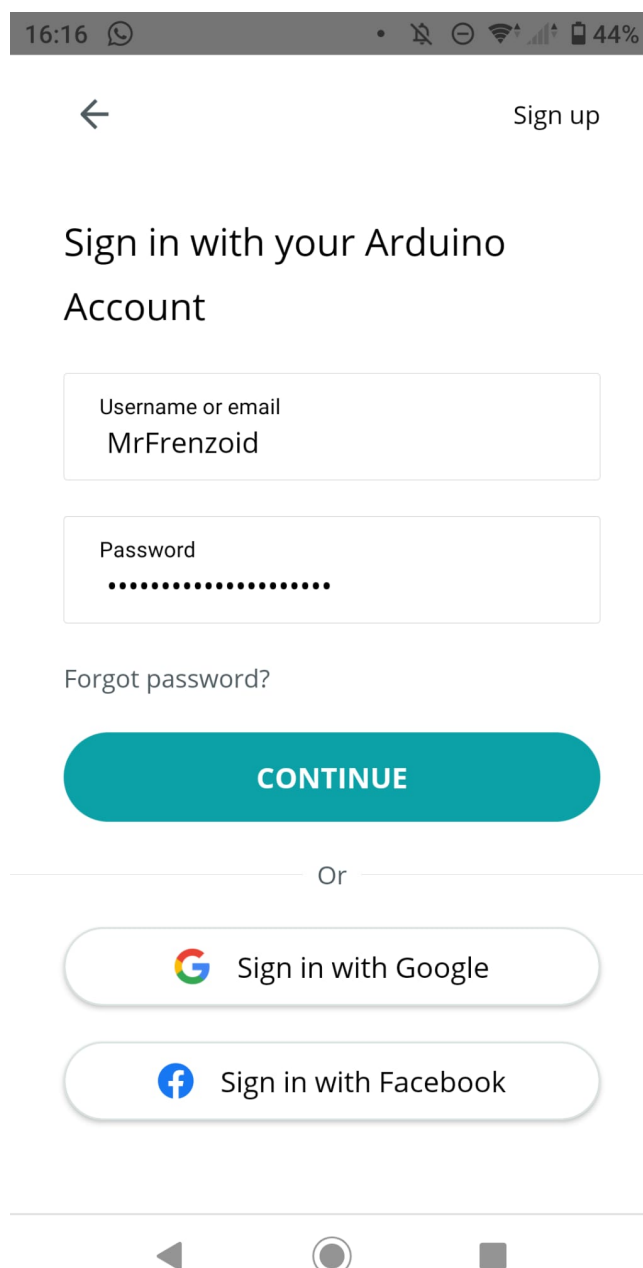


Fig. 21 Captura de pantalla de la aplicación a instalar.

Una vez instalada, la arrancamos, y nos pedirá que iniciemos sesión.



**Fig. 22** Pantalla de inicio de la aplicación.



16:16

Sign up

## Sign in with your Arduino Account


Username or email  
MrFrenzoid


Password  
.....

[Forgot password?](#)

**CONTINUE**

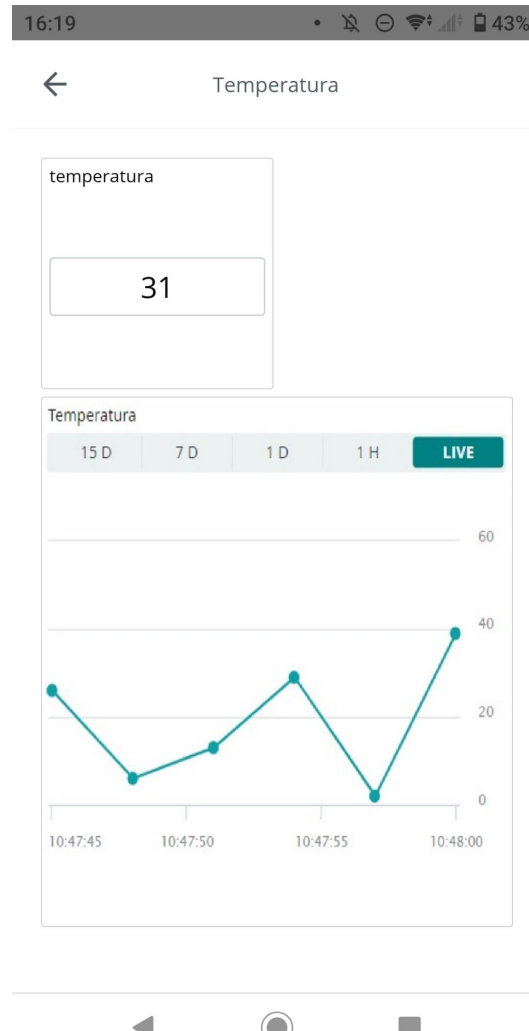
Or

 Sign in with Google

 Sign in with Facebook

**Fig. 21** Captura de pantalla de la aplicación, formulario de inicio de sesión.

Una vez introducidas las credenciales, podremos ver nuestro dashboard.



## 2 Conclusiones.

Hemos visto cómo usar nuestro dispositivo arduino para enlazarlo a un servicio en cloud, y recibir datos en vivo, demostrando que de esta manera, podemos tener una gestión centralizada de los datos de nuestro dispositivo.