

## COMUNICACIÓN INÁLAMBRICA WIFI POR MEDIO DE LA TARJETA DE DESARROLLO ESP32

Las herramientas o software por utilizar son las siguientes:

- Arduino IDE 1.8.19 ó posterior
- Plataforma IoT (Ubidots)
- Tarjeta de desarrollo ESP WROOM 32
- Cuenta gratuita de Ubidots



### Introducción

El objetivo de la presente guía es brindar las herramientas necesarias para realizar la comunicación vía Wifi de un dispositivo hacia un servidor que permita monitorizar variables físicas, biológicas o químicas. Una de las ventajas de esta comunicación es que permite la monitorización y el control remoto de dispositivos en movimiento o en ubicaciones remotas, además del cifrado de datos y autenticación, lo que permite proteger la comunicación entre los microcontroladores y otros dispositivos en la red, por ultimo y no menos importante es que permite el almacenamiento, procesamiento y análisis de datos en la nube. Hoy en día existen diferentes tipos de servidores, lo cuál permite flexibilidad para controlar el sistema diseñado, ya no es necesario programar alguna plataforma que permite monitorizar las variables, sino que el mismo servidor almacena los datos recibidos y pueden ser configurados por el usuario para que puedan ser visualizados en forma gráfica. Con esto, permite que el diseñador del sistema tenga un mayor enfoque en la programación del microcontrolador y en lo que compete a la electrónica del sistema.

En esta ocasión se permitirá hacer uso de la tarjeta de desarrollo ESP32, el cual tiene como característica peculiar la conectividad inalámbrica; cuenta con la integración de los módulos Wifi y Bluetooth, además de ser de muy bajo costo a comparación de otras tarjetas que realizan menos funciones. Además de ello, se usará el servidor Ubidots, el cual tiene la flexibilidad para conectar dispositivos y almacenar los datos que se envían.

### Instalación de software, firmware y librería.

Para comenzar, es necesario contar con el entorno de Arduino IDE, el cual puede descargarse a través de la siguiente dirección: <https://www.arduino.cc/en/software>

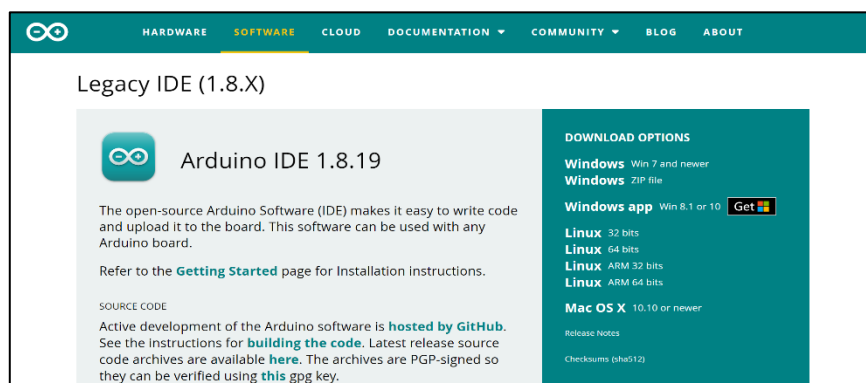


Figura 1

Actualmente se encuentra disponible la versión 1.8.19 de Arduino IDE, sin embargo, recientemente se lanzó una versión mejorada, la versión 2.0.4, que también puede ser descargada en la misma dirección antes mencionada.

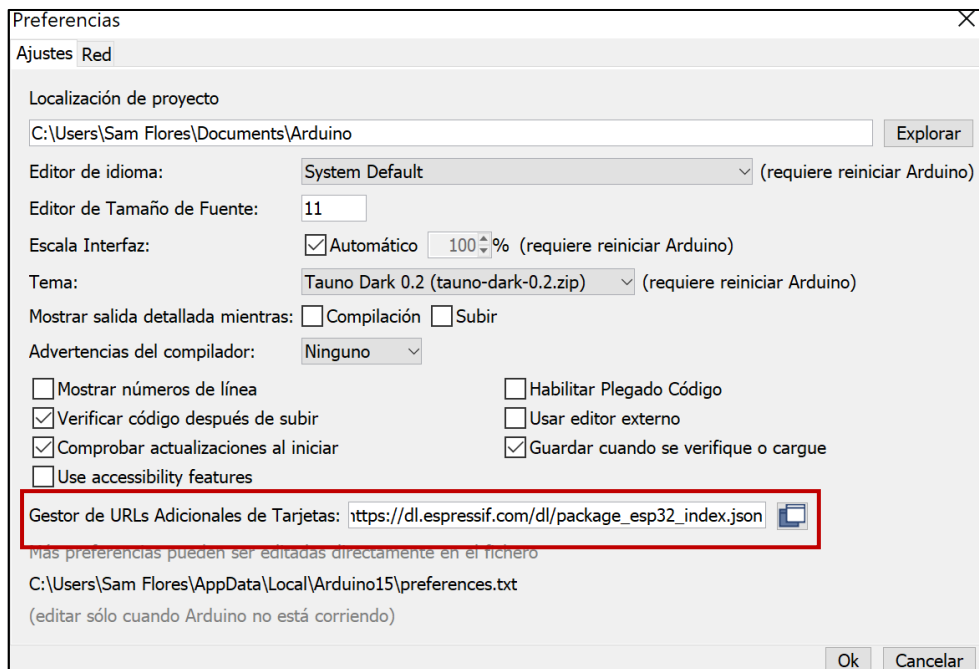


Figura 2

Una vez descargado el entorno de Arduino, es necesario contar con el firmware de la tarjeta de desarrollo ESP32, para ello es necesario abrir la aplicación de Arduino IDE y dirigirse hacia la sección de Archivo > Preferencias y en el campo de texto “Gestor de URLs Adicionales de Tarjetas” se ingresa la siguiente dirección: [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json), como se puede observar en la Figura 2.

Una vez que se haya ingresado la dirección en el “Gestor de URLs” se selecciona el botón “Ok” para guardar las modificaciones efectuadas.

Con la Url ya ingresada, se puede seleccionar la tarjeta a utilizar, que en este caso es la tarjeta de desarrollo ESP32. Para seleccionarla, se debe dirigir hacia los apartados o secciones del IDE de Arduino. En la sección de “Herramientas” y en donde se encuentra “Placa” se selecciona y se debe dirigir hacia el “Gestor de tarjetas”. Ver Figura 3.

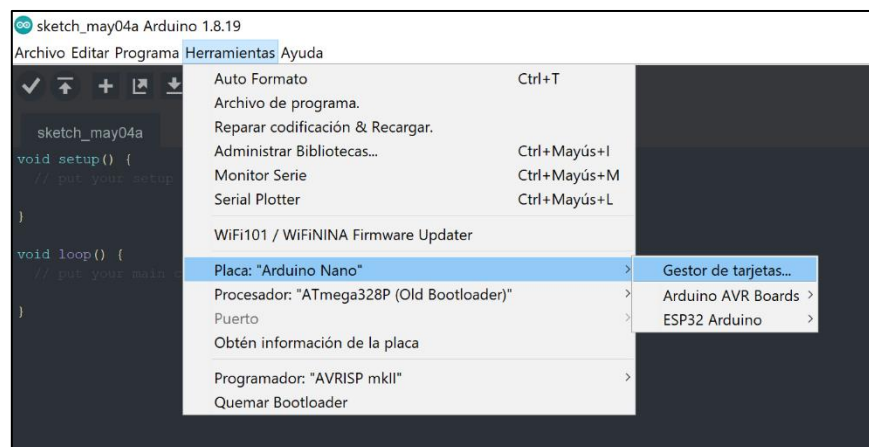


Figura 3

En el “Gestor de tarjetas”, se selecciona el buscador y se escribe la palabra “esp32”. Se procede a instalar el paquete y se espera a que termine el proceso de instalación, una vez terminado, se selecciona el botón “cerrar”. Ver Figura 4.

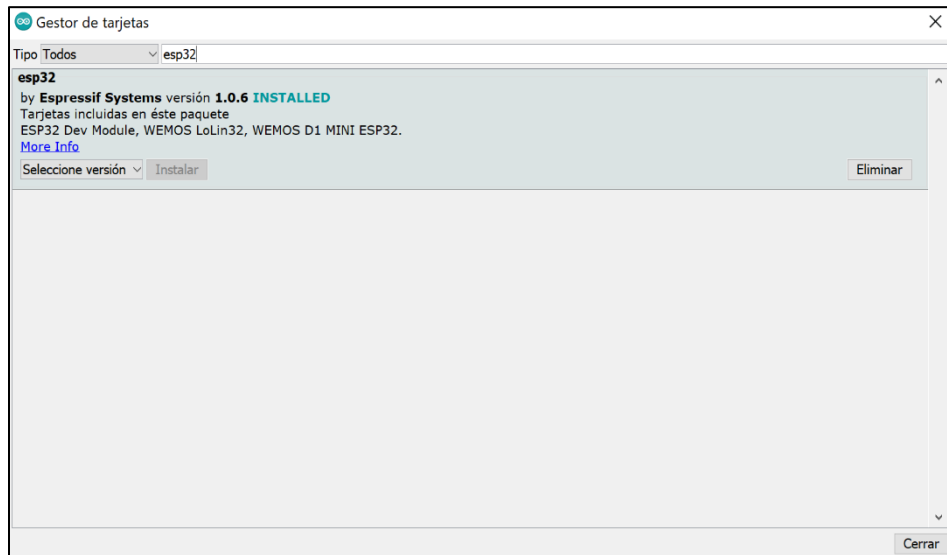


Figura 4

Ya teniendo la tarjeta instalada, se procede a instalar la librería que se usará para este proyecto. En el siguiente repositorio de Github, podemos encontrar la librería que se necesita: <https://github.com/ubidots/ubidots-esp32>

En la Figura 5 se encuentra visualizado el repositorio donde se debe ingresar, y en la parte de requerimientos se encuentra “Ubidots library”, se selecciona y la descarga comenzará inmediatamente. La librería será descargada en formato zip.

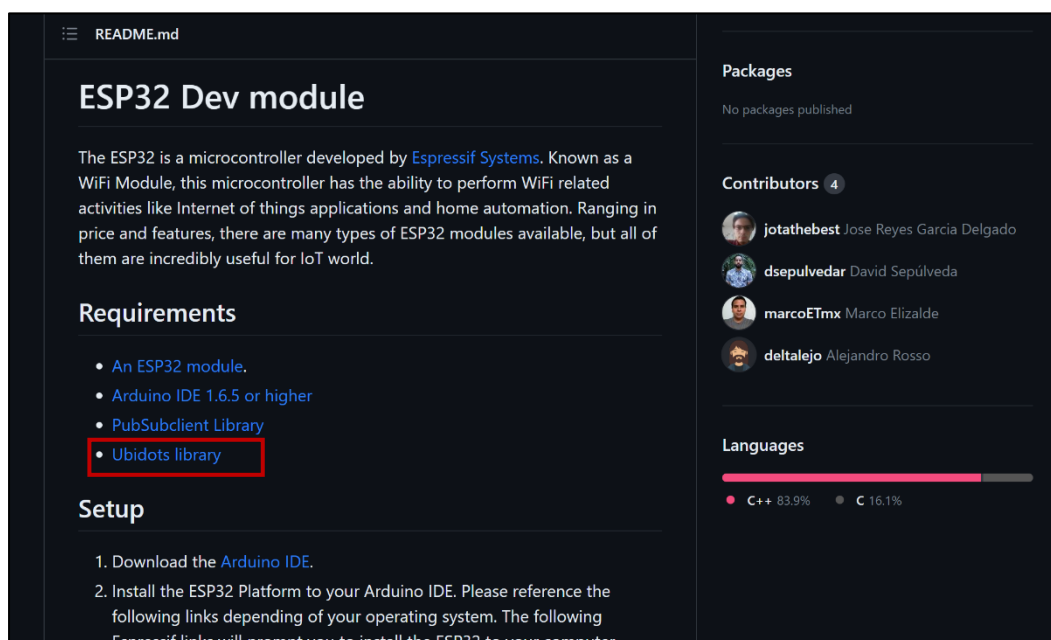


Figura 5

Una vez descargado el archivo, en el IDE de Arduino se incluye la librería de Ubidots. En la sección de Programa > Incluir Librería > Añadir biblioteca .ZIP, se selecciona el archivo .zip que se acaba de descargar. Ver Figura 6.

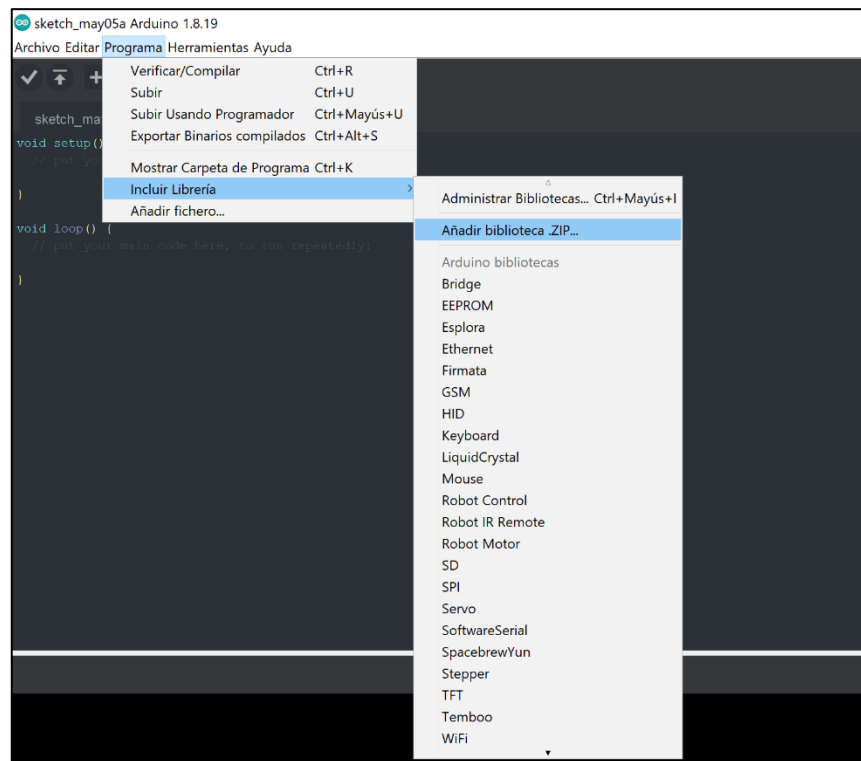


Figura 6

### Creación y configuración de cuenta de Ubidots.

Para crear una cuenta de Ubidots, se ingresa a la página oficial del servidor en la siguiente dirección: <https://es.ubidots.com/> y se selecciona la sección de “Regístrate”.



Figura 7

Ubidots da la opción de una prueba gratuita de 30 días, pero también da acceso a estudiantes y profesores para usar su plataforma. Para dirigirse a la plataforma gratuita es necesario seleccionar el apartado que dice “plataforma GRATUITA” y automáticamente se redirigirá a la sección de registro de usuario para estudiantes y profesores.



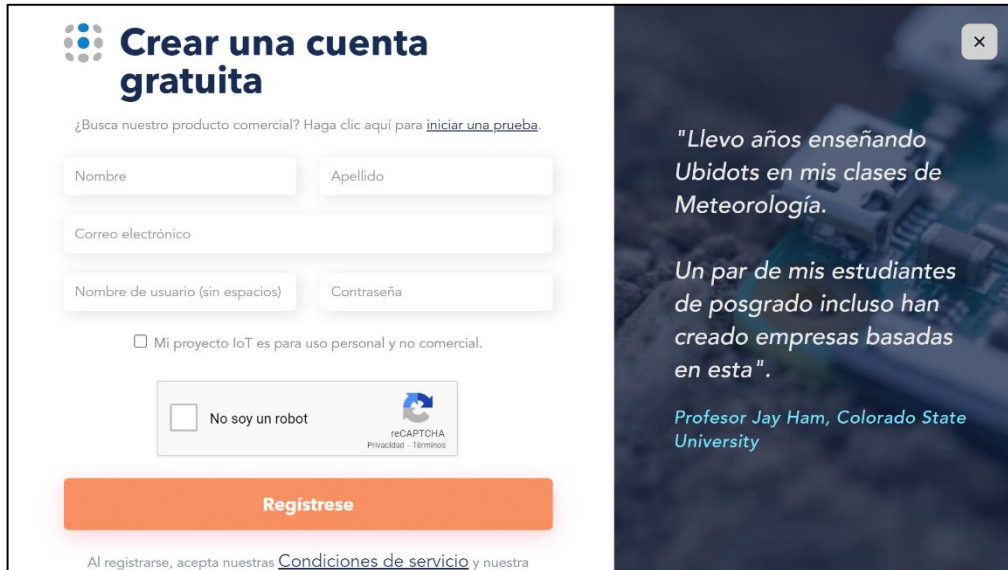
Figura 8

Al momento de seleccionar la opción de ingreso como estudiantes, será redireccionado a una plataforma de Ubidots especialmente para estudiantes. Seleccionar “Crear cuenta STEM”. Ver Figura 9.



Figura 9

Se crea una cuenta, llenando todos los datos que se solicitan y posteriormente se enviará un mensaje de confirmación a su cuenta de correo. Ver Figura 10.



**Crear una cuenta gratuita**


¿Busca nuestro producto comercial? Haga clic aquí para [iniciar una prueba](#).

Nombre  Apellido

Correo electrónico

Nombre de usuario (sin espacios)  Contraseña

☐ Mi proyecto IoT es para uso personal y no comercial.

☐ No soy un robot  reCAPTCHA Privacidad - Términos

**Regístrate**

Al registrarse, acepta nuestras [Condiciones de servicio](#) y nuestra

*"Llevo años enseñando Ubidots en mis clases de Meteorología."*

*Un par de mis estudiantes de posgrado incluso han creado empresas basadas en esta".*

*Profesor Jay Ham, Colorado State University*

Figura 10

Después de confirmar por cuenta de correo. Ingrese nuevamente a la plataforma en el siguiente link: <https://stem.ubidots.com/accounts/signin/> e inicie sesión. Posteriormente visualizará un menú como el que se muestra en la Figura 11.

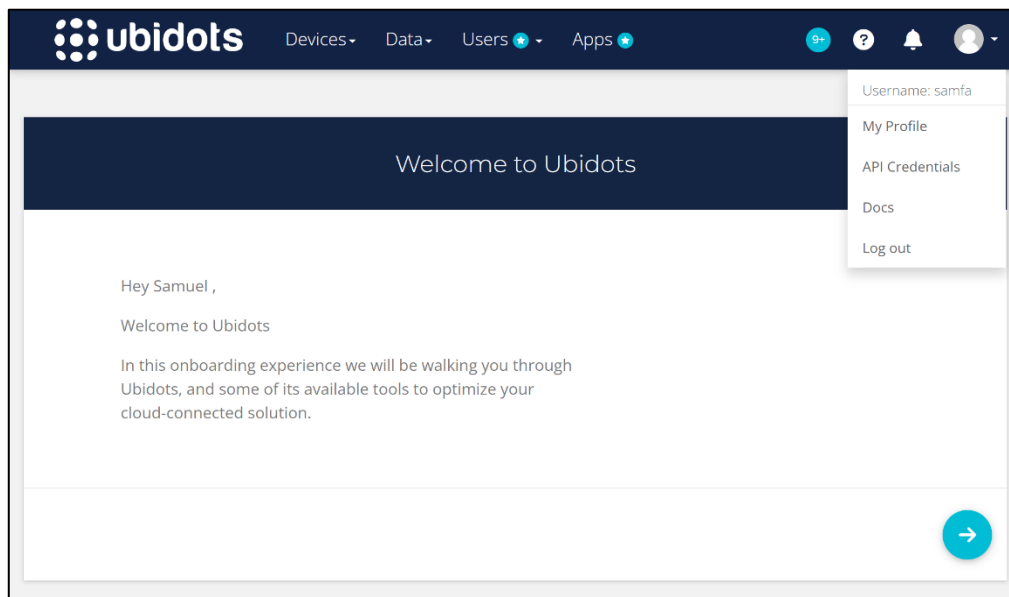


Figura 11

### Comunicación de la tarjeta de desarrollo ESP32 al servidor Ubidots

Para la comunicación vía WiFi primeramente se crea un nuevo proyecto en Arduino IDE y posteriormente se incluye la librería Ubidots que se descargó anteriormente al código. Para ello se diseñó la siguiente plantilla que se puede utilizar como base para el desarrollo de proyectos. Ver Figura 12.

```
#include <Ubidots.h>

const char* UBIDOTS_TOKEN = "";
const char* WIFI_SSID = "";
const char* WIFI_PASS = "";

Ubidots ubidots(UBIDOTS_TOKEN, UBI_HTTP);

void setup() {
    Serial.begin(9600);
    ubidots.wifiConnect(WIFI_SSID, WIFI_PASS);
}

void loop() {

    ubidots.add("Parametro1", sensor1);
    ubidots.add("Parametro2", sensor2);

    ubidots.send();
    delay(2000);
}
```

Figura 12

A continuación, se explicará con mayor detalle lo que realiza cada línea de código para su mayor comprensión.



Primeramente, se inserta la librería “Ubidots.h” el cuál contiene todas las funciones necesarias para la comunicación WiFi con el servidor.

```
#include <Ubidots.h>
```

Se establecen tres constantes del tipo char y serán nombrados de acuerdo con la función que tendrán.

```
const char* UBIDOTS_TOKEN = "";  
const char* WIFI_SSID = "";  
const char* WIFI_PASS = "";
```

Esta constante a establecer permite almacenar la contraseña de red Wfi a la que se conectará el sistema para realizar la comunicación.

Esta constante a establecer permite almacenar un código especial que será proporcionado por el servidor Ubidots.

Esta constante a establecer permite almacenar el nombre de red Wfi a la que se conectará el sistema a desarrollar para realizar la comunicación.

La siguiente línea establece la comunicación con el servidor por medio del token que se estableció anteriormente y del código que proporciona Ubidots.

```
Ubidots ubidots(UBIDOTS_TOKEN, UBI_HTTP);
```

En la función void setup () permite establecer la comunicación serial a 9600 baudios y por medio de la función ubidots.wifiConnect establece la comunicación con la red wifi.

```
void setup() {  
  Serial.begin(9600);  
  ubidots.wifiConnect(WIFI_SSID, WIFI_PASS);  
}
```

En la función void loop () permite agregar los valores leídos por los sensores y enviarlos hacia el servidor de manera constante e iterada.

```
void loop() {  
  ubidots.add("Parametro1", sensor1);  
  ubidots.add("Parametro2", sensor2);  
  ubidots.send();  
  delay(2000);  
}
```

La función ubidots.add recibe como parámetro el valor de cada sensor

La función ubidots.send() envía los valores leídos de los sensores al servidor ubidots.

La función delay administra el tiempo del proceso en que se envían los datos sensados.



### Ejemplo: Monitoreo de sensor ultrasónico y sensor de temperatura LM35

Teniendo en cuenta lo antes mencionado, para este ejemplo se realizará un pequeño sistema de monitoreo que permita leer dos variables, el sensor ultrasónico medirá la distancia, mientras que el sensor LM35 medirá la temperatura.

Las conexiones se pueden visualizar en el siguiente esquema. Ver Figura 13.

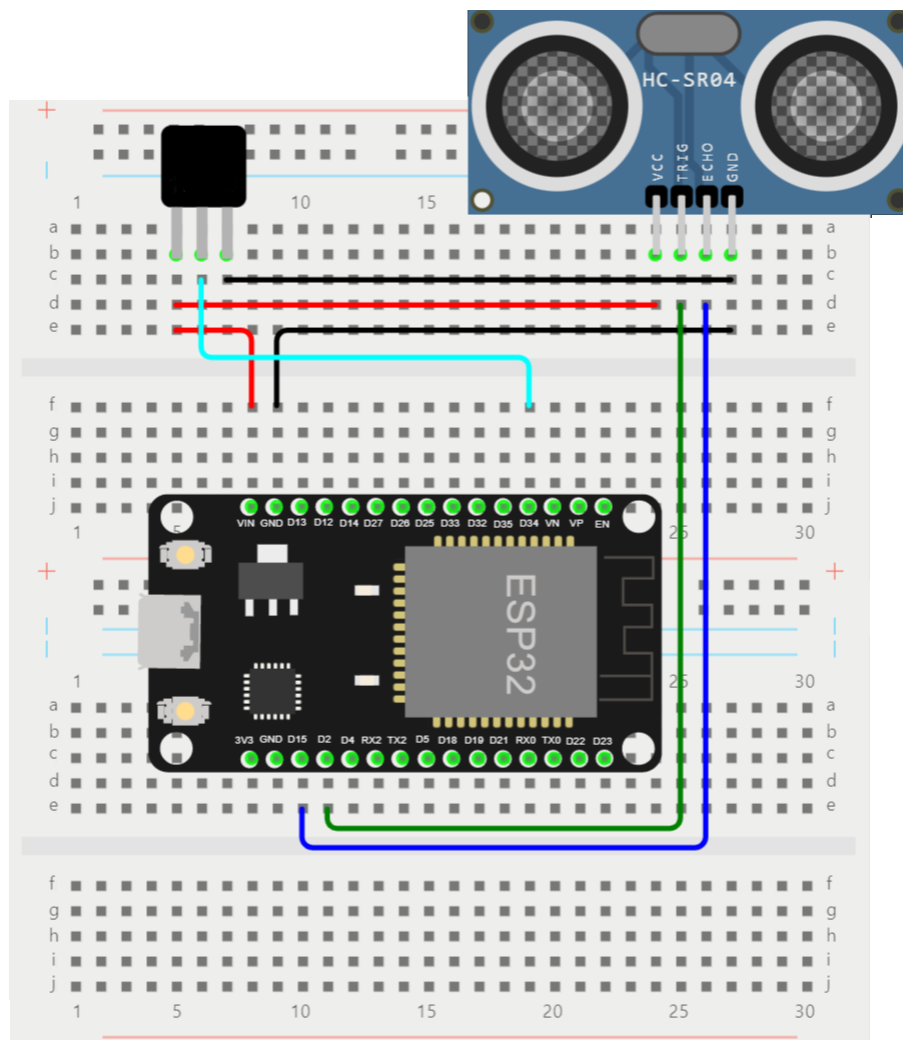


Figura 13

Después de realizar las conexiones para el monitoreo de los sensores, se carga el código realizado a la tarjeta de desarrollo ESP32. El código para este ejemplo se encuentra en la Figura 14.

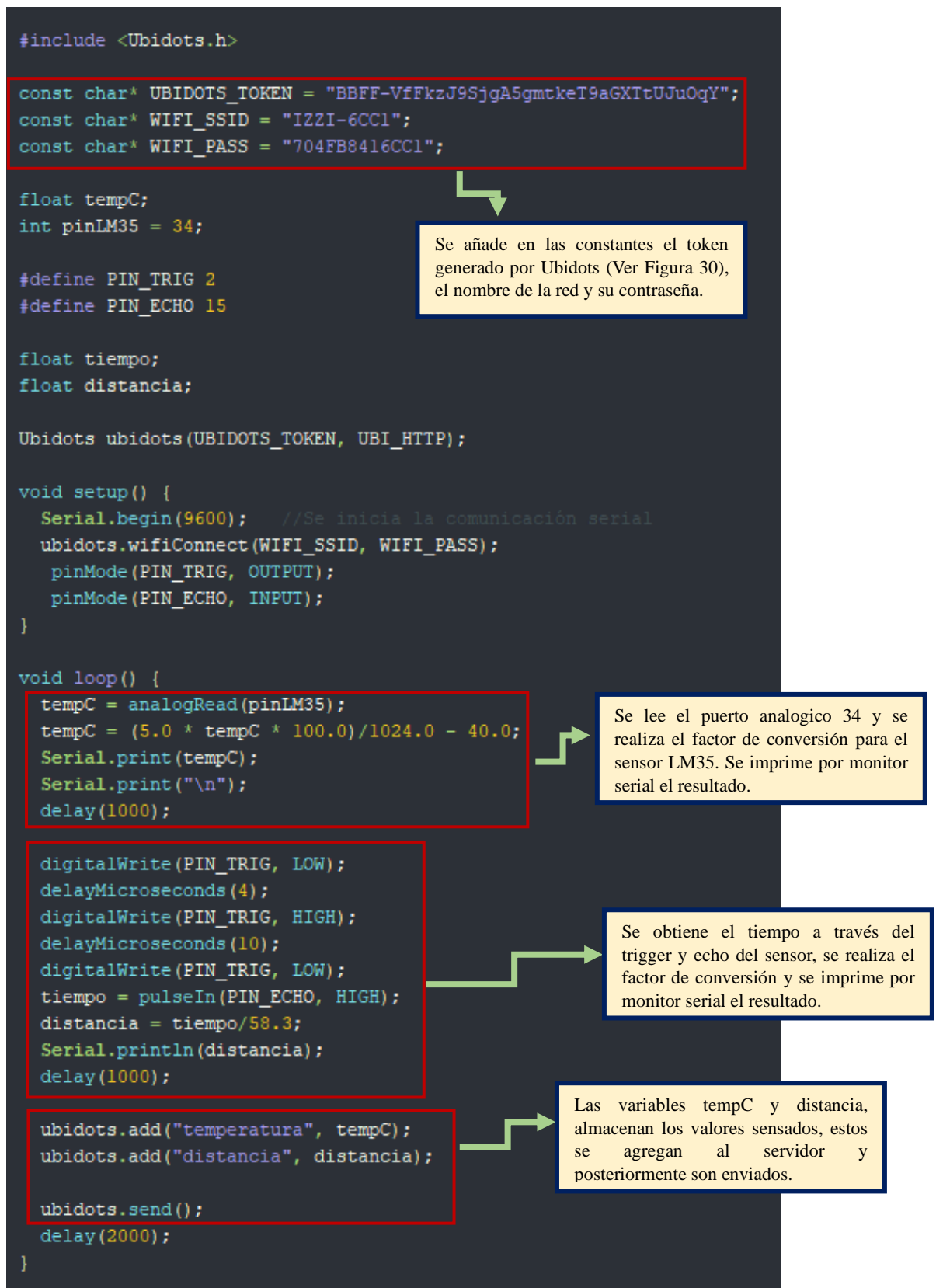


Figura 15

### Creación de dashboard en Ubidots

Después de realizar las conexiones para el monitoreo, y tener el código a compilar, se creará una dashboard dentro de la plataforma Ubidots, para ello se iniciará sesión con la cuenta ya creada. Ver Figura 16.

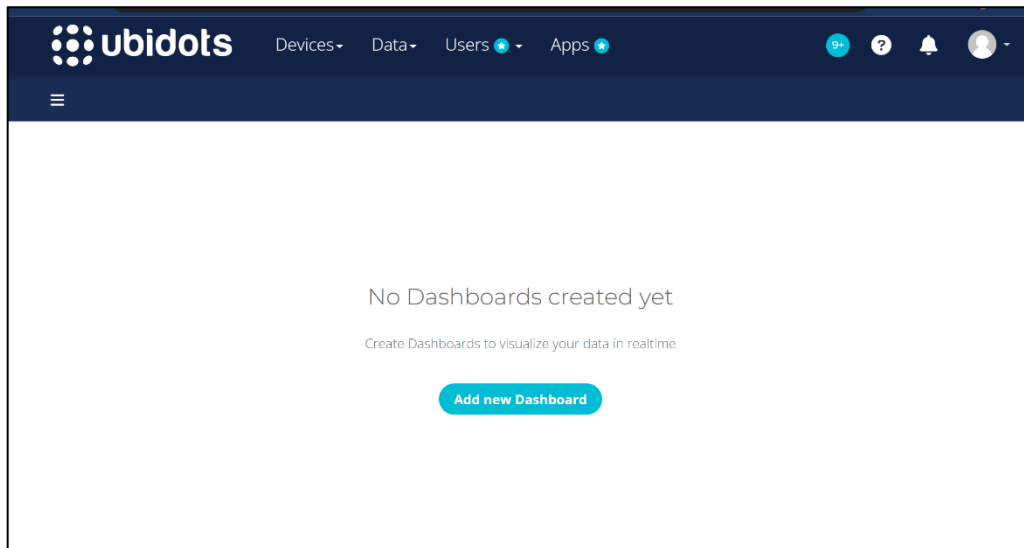


Figura 16

Se añade una nueva dashboard, se le puede cambiar el nombre, además de hacerle algunas modificaciones a la apariencia. Para este caso, no es necesario muchas modificaciones, se puede establecer la configuración que viene por defecto. Ver Figura 17.

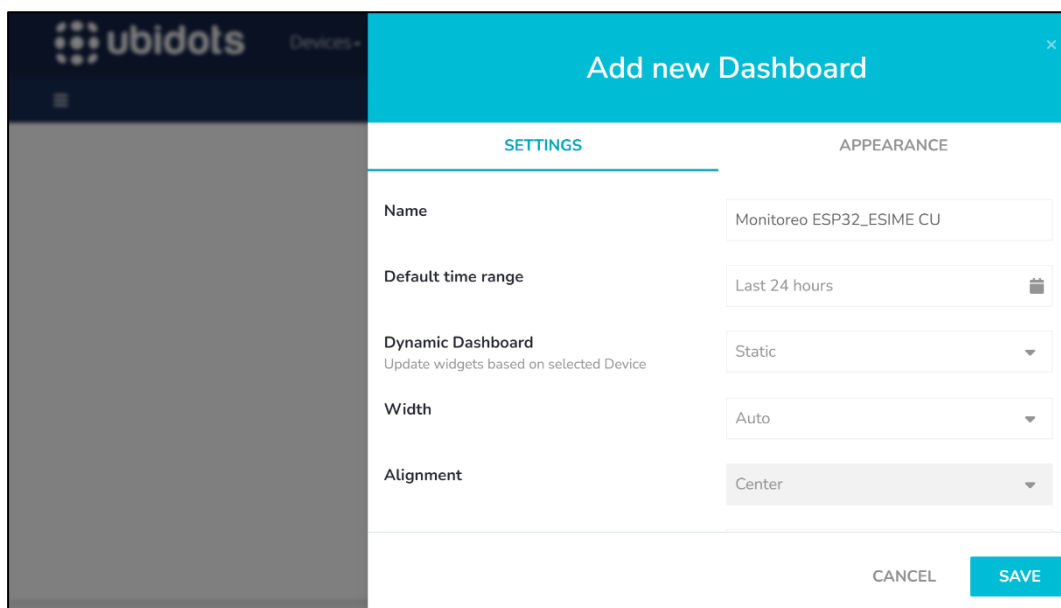


Figura 17

Después de ya tener la dashboard creada, se crea un nuevo dispositivo, para ello se selecciona la opción “Devices” que viene en el menú superior que se encuentra en la plataforma. Ver Figura 18.

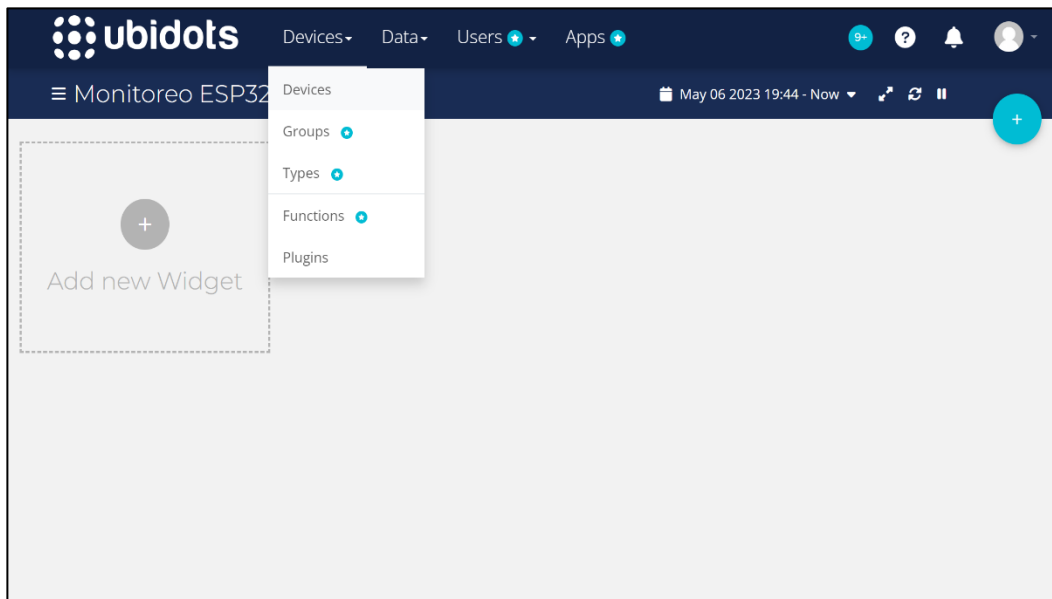


Figura 18

Se agrega un nuevo dispositivo y se selecciona la opción de “Blank Device”. Ver Figura 19.

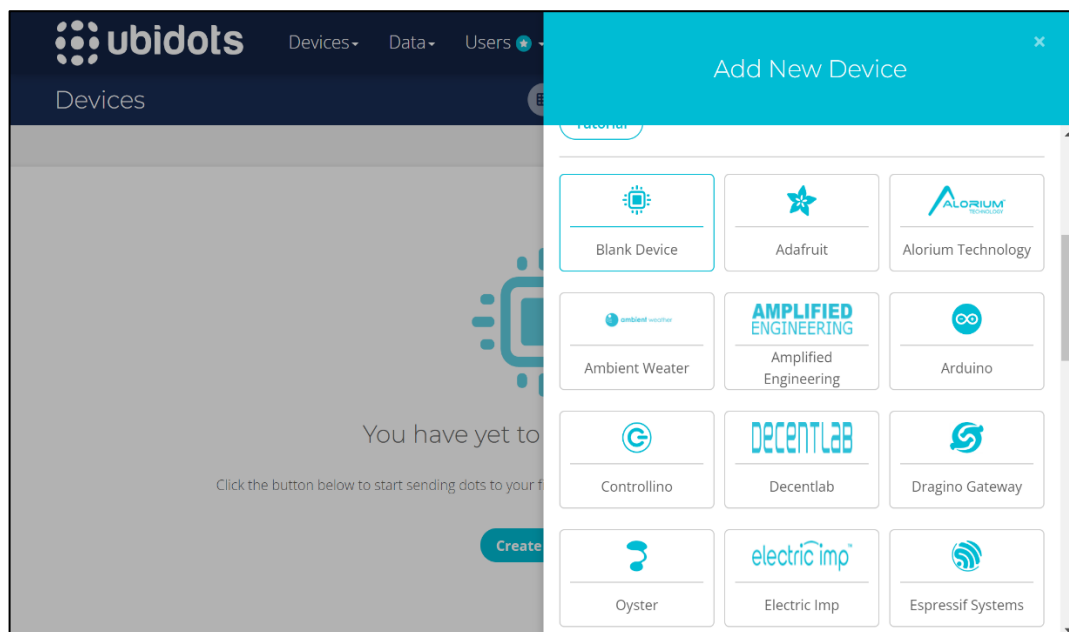


Figura 19

Se agrega el nombre del dispositivo, para este caso, como se utilizará la tarjeta de desarrollo ESP32, se le escribirá de igual forma. Ver Figura 20.

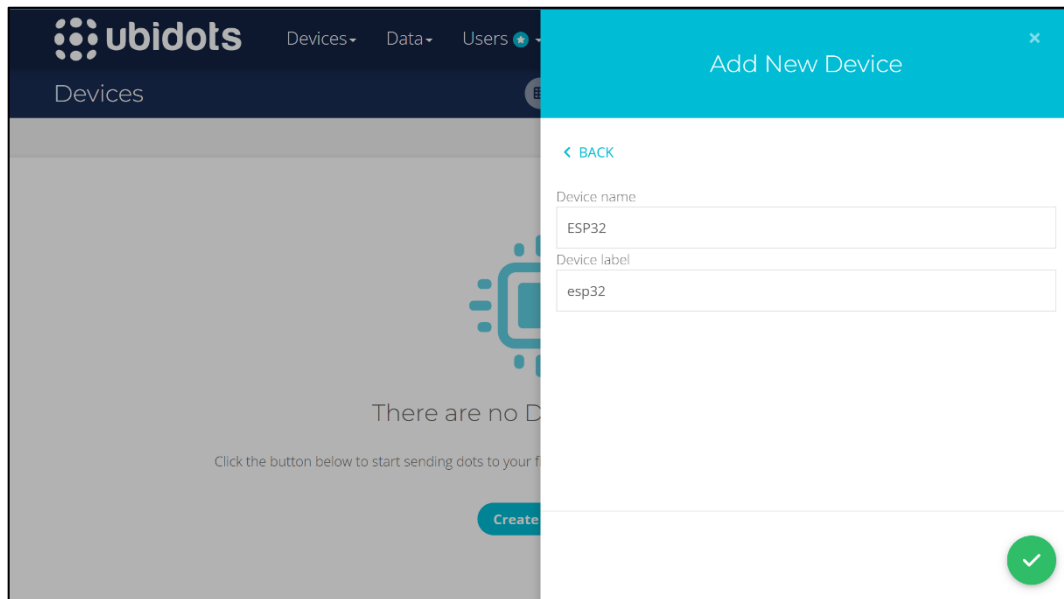


Figura 20

Ya creado el dispositivo, se crean las variables en donde se visualizaran los datos enviados del microcontrolador. Para ello, se selecciona la opción “Add Variable” y se define del tipo Raw.

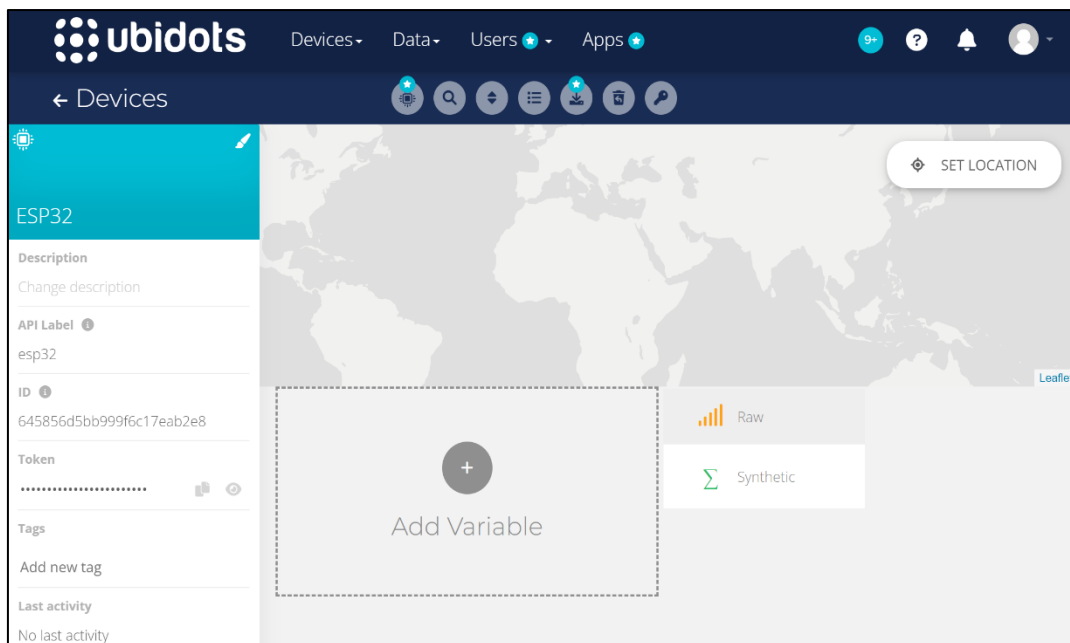


Figura 21

Teniendo ya las variables creadas, se observará de forma similar a la Figura 22.

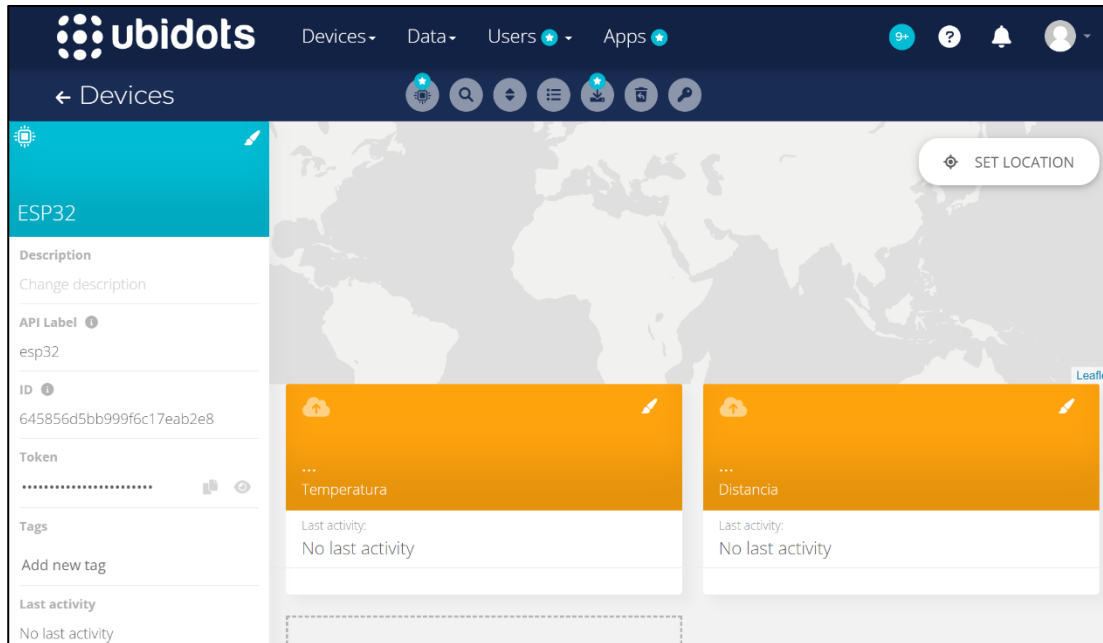


Figura 22

Se dirige nuevamente a la dashboard, que se puede acceder a Data y seleccionar en Dashboards. Ver Figura 23.

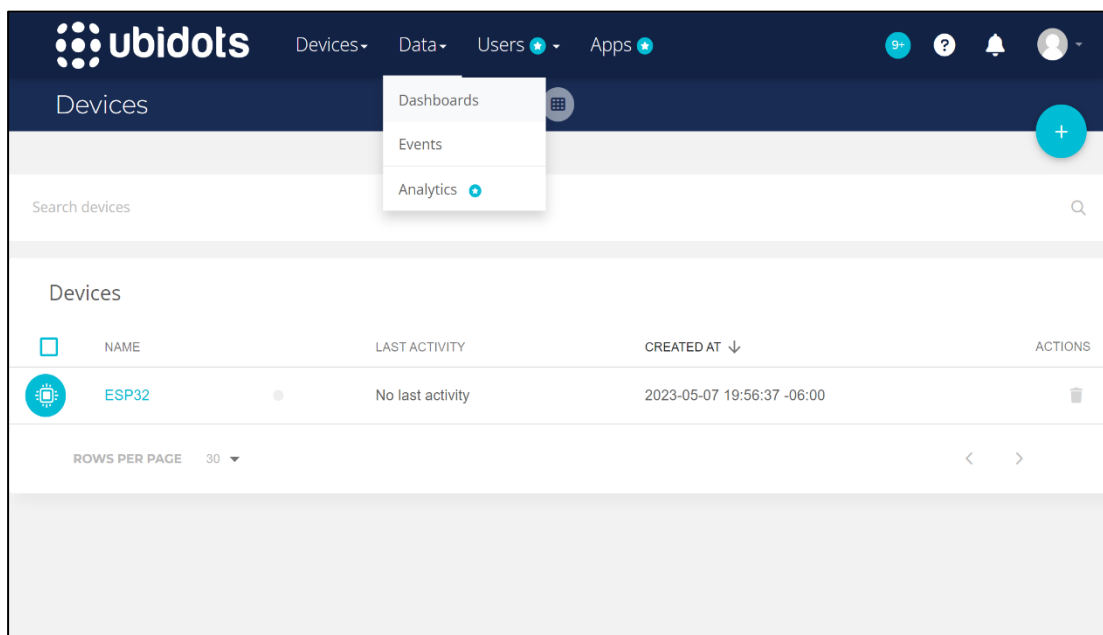


Figura 23

Se agrega un widget en la dashboard. Ver Figura 24.

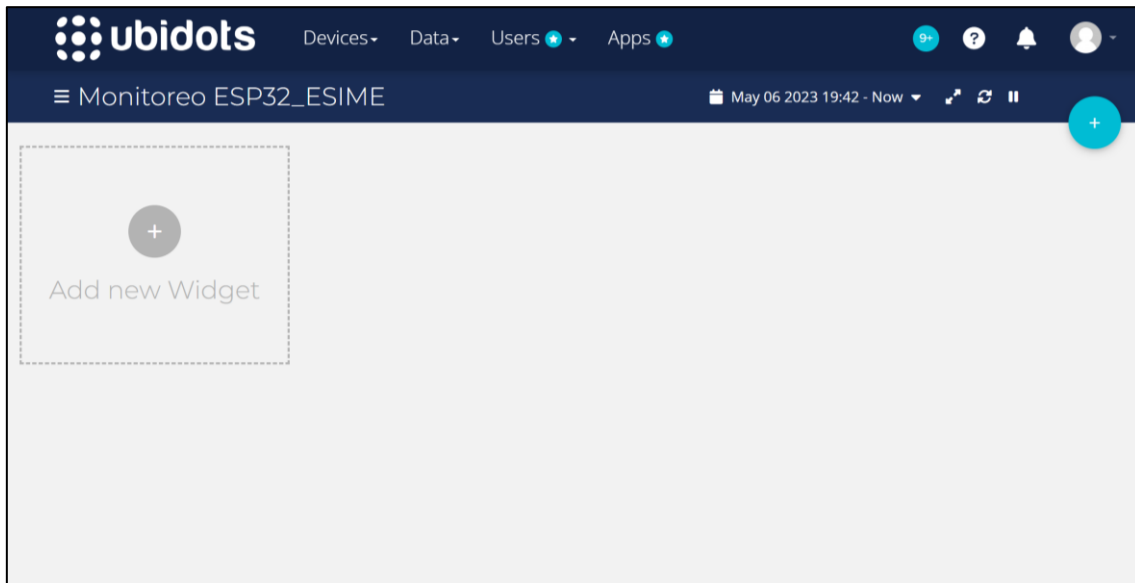


Figura 24

Se realiza la configuración de los widgets, dependiendo el valor sensado se puede escoger la imagen a visualizar y tipo de gráfico. Ver Figura 25.

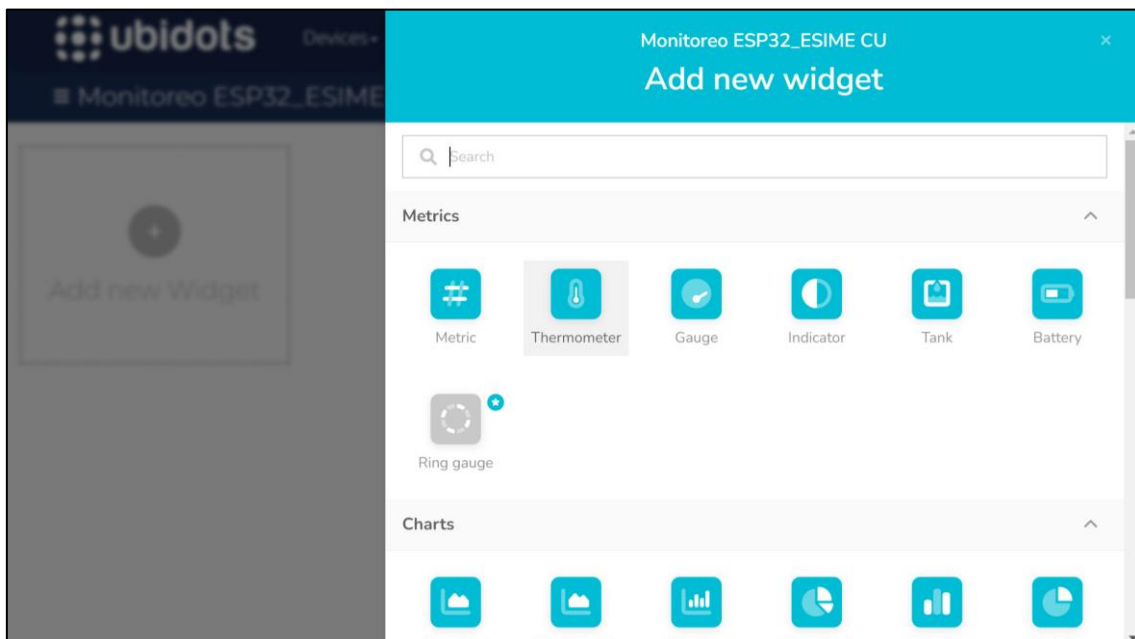


Figura 25



Se agrega una variable al widget en creación. Ver Figura 26.

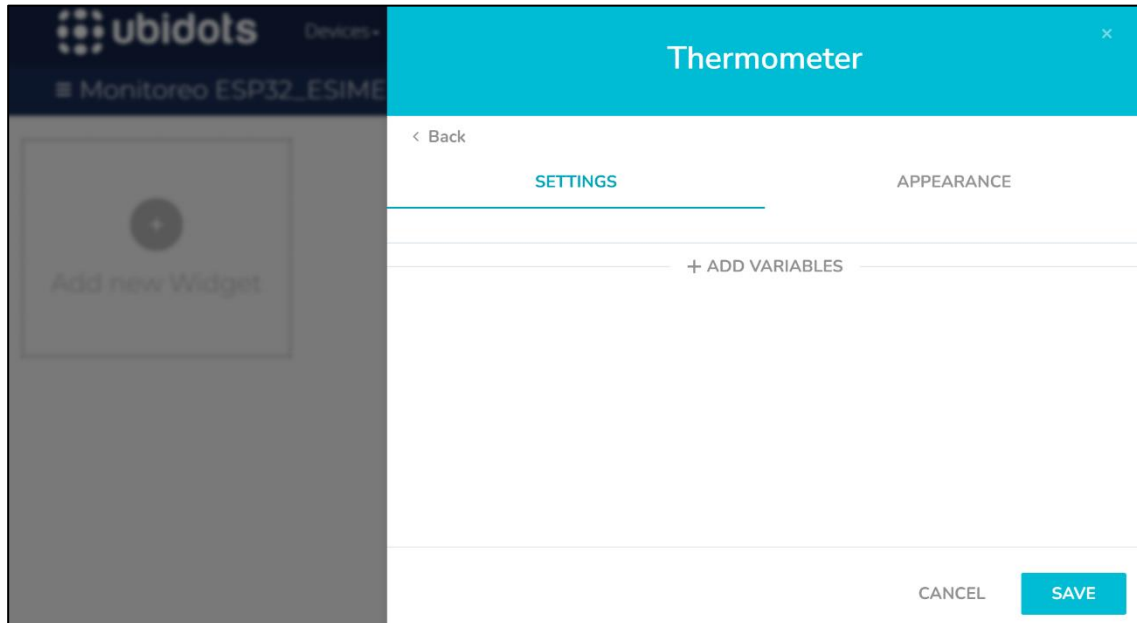


Figura 26

Al agregar la variable, Ubidots permite relacionar las variables creadas de otros dispositivos, en este caso se escoge la variable Temperatura del dispositivo ESP32. Ver Figura 27.

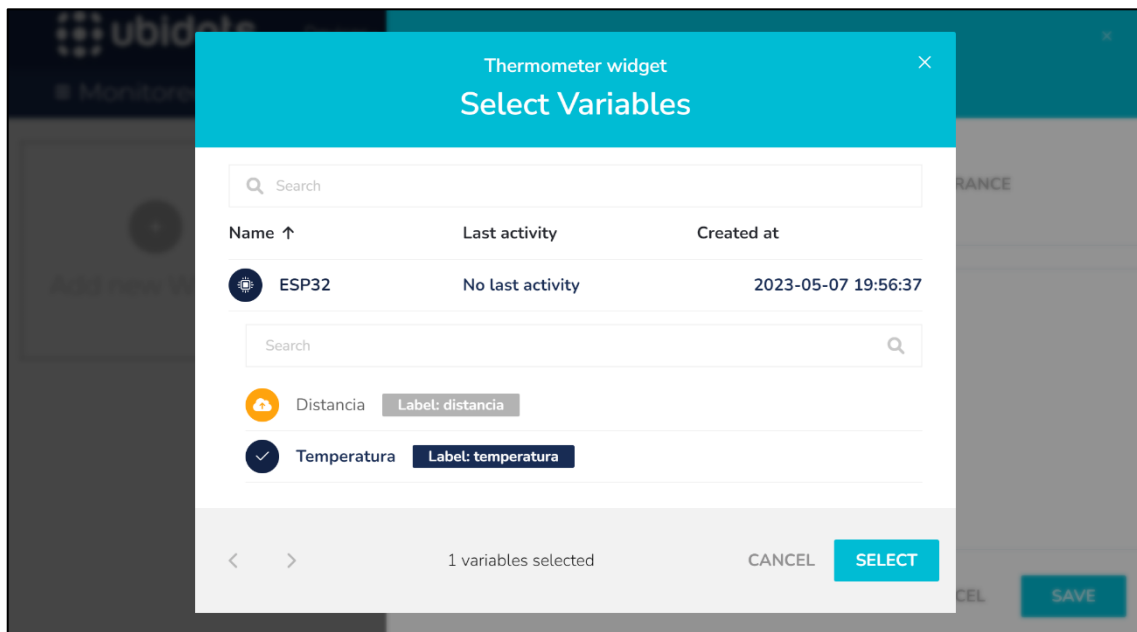


Figura 27

Después de haber seleccionado la variable del dispositivo, se puede cambiar alguna configuración, además de su apariencia. Por ultimo se guardan los cambios con el botón “save”. Ver Figura 28.

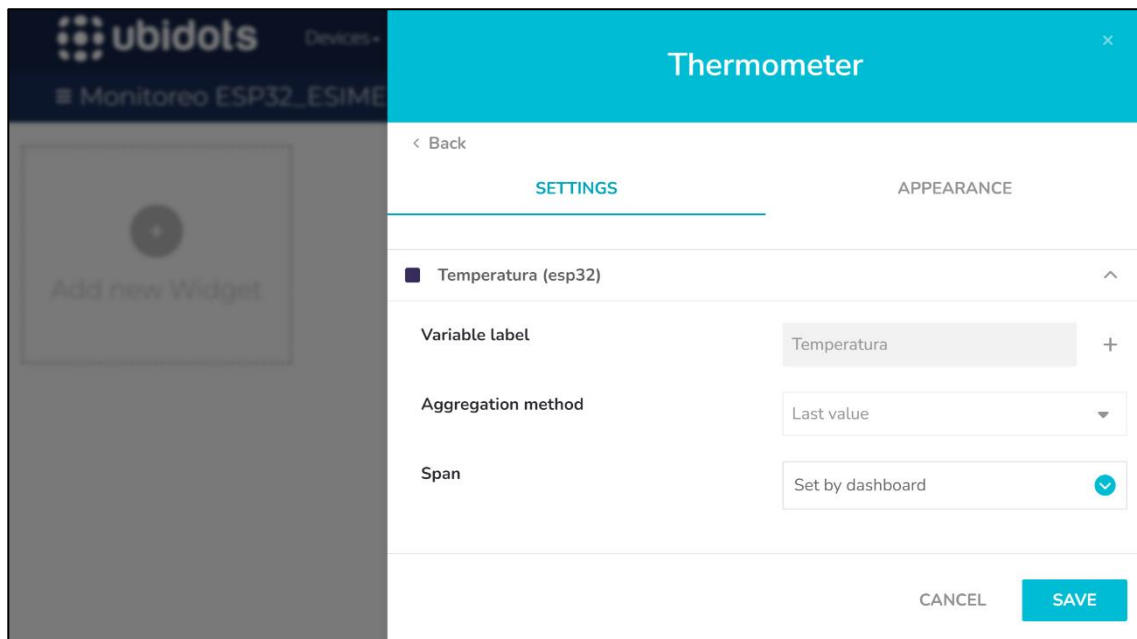


Figura 28

Después de haber realizado los pasos anteriores, se podrá visualizar la variable en la dashboard. Ver Figura 29.

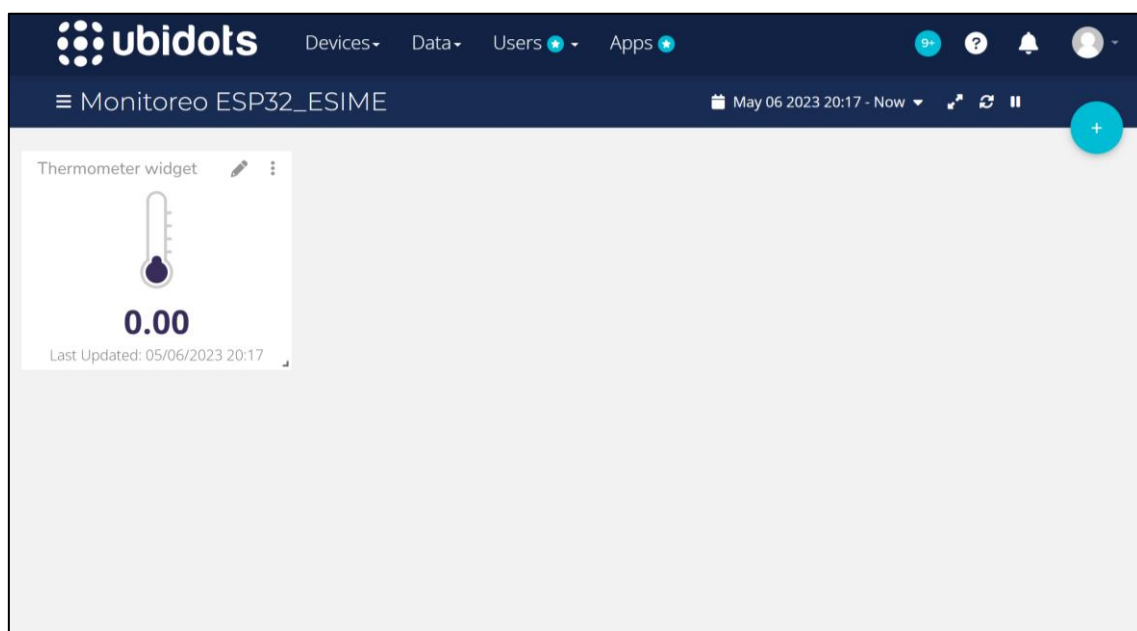


Figura 29

Después de tener agregadas las variables, otro punto importante es obtener el token que proporciona Ubidots y que será indispensable para realizar la comunicación WiFi, está se puede obtener en la sección de dispositivos y se selecciona el dispositivo creado, que en este caso fue “ESP32”. El token se puede visualizar y se puede copiar para agregarlo al código. Ver Figura 30.

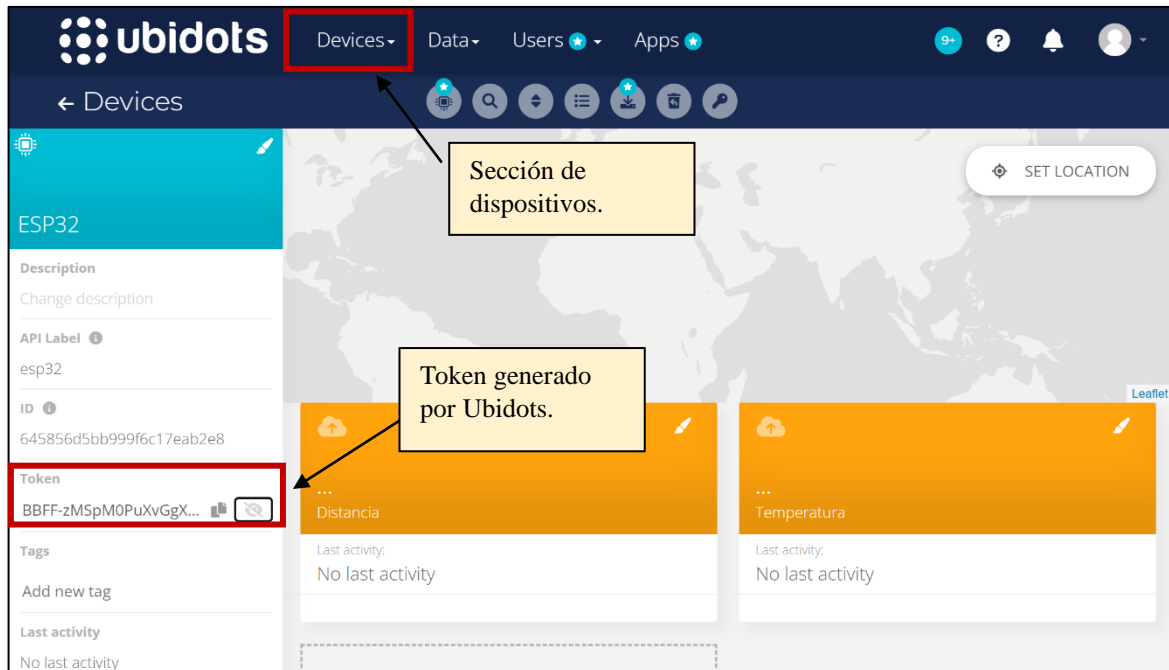


Figura 30

### Compilación y programación del microcontrolador

Una vez que se tiene listo el dispositivo en Ubidots y el código listo, se realiza la compilación y programación de la ESP32, para ello se selecciona la placa en la sección de herramientas y se verifica que este conectado la placa correspondiente. Ver Figura 31.

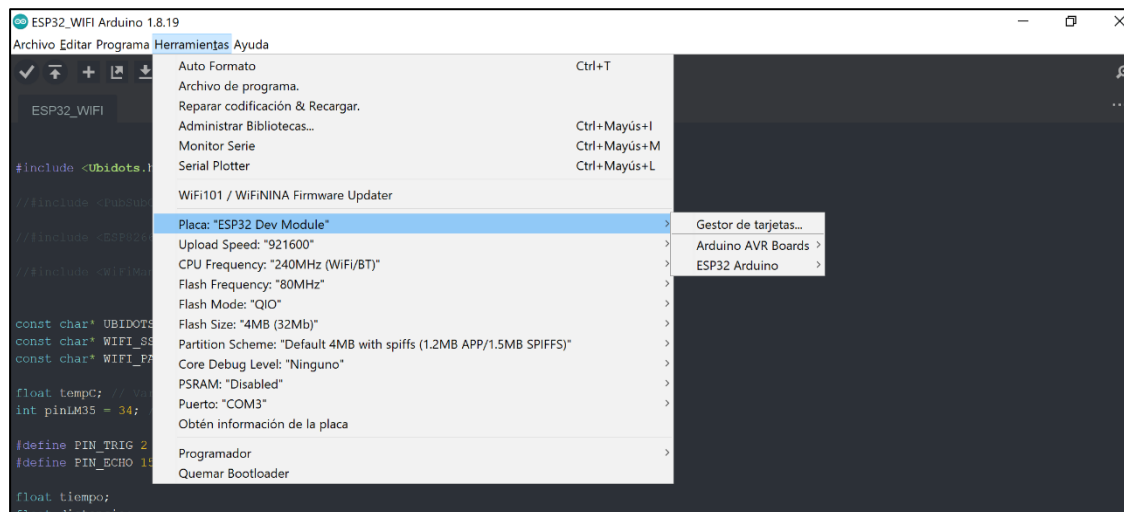
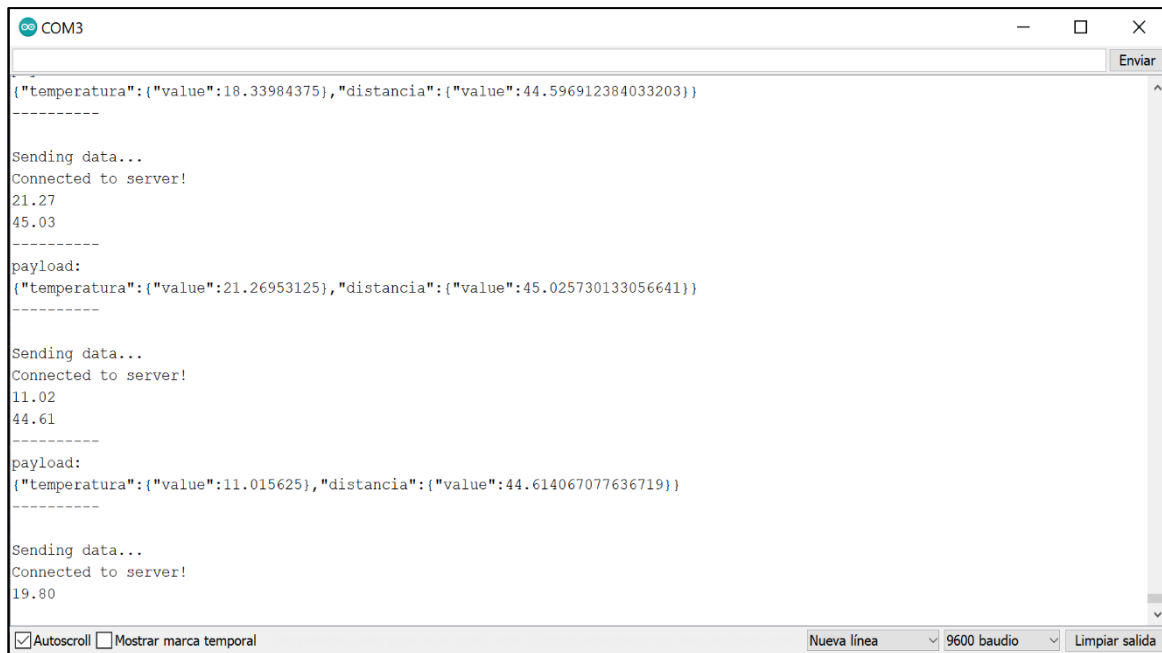


Figura 31

Se compila el código y cuando finalice la compilación puede verificar el envío de datos a través del monitor serial de Arduino. Ver Figura 32.



```
COM3
{"temperatura":{"value":18.33984375},"distancia":{"value":44.596912384033203}}
-----
Sending data...
Connected to server!
21.27
45.03
-----
payload:
{"temperatura":{"value":21.26953125},"distancia":{"value":45.025730133056641}}
-----
Sending data...
Connected to server!
11.02
44.61
-----
payload:
{"temperatura":{"value":11.015625},"distancia":{"value":44.614067077636719}}
-----
Sending data...
Connected to server!
19.80

[Autoscroll] [Mostrar marca temporal] Nueva línea 9600 baudio Limpiar salida
```

Figura 33

Para visualizar el monitoreo en Ubidots, se ingresa a la sección de Data y se selecciona la dashboard creada. De acuerdo con los widgets añadidos se podrá visualizar un tablero como el que se muestra en la Figura 34.



Figura 34

Con esto, es posible conectar diferentes sensores por medio de la ESP32 hacia el servidor Ubidots, solamente hace falta considerar el ingreso y creación de variables dentro de la dashboard así como la lectura de los sensores a través de la programación en Arduino.

### **Conclusión**

La presente explicación se realizó con el fin de facilitar la conectividad que ofrece la ESP32 a través de sus módulos WiFi y Bluetooth, con ello se tiene el objetivo de desarrollar proyectos que tengan mayor eficacia y eficiencia para el envío de datos hacia servidores web como lo es en el caso de Ubidots.

La plataforma de Ubidots tiene la peculiaridad de brindar servicio gratuito para estudiantes y profesores con fines meramente educativos, con el fin de facilitar el aprendizaje, esto hace que pueda ser accesible para experimentación de prototipos con miras a tener un mayor alcance.

Además de ello, la programación en Arduino permite la integración de bibliotecas ya preestablecidas y que facilitan el uso del programador.

### **Repositorio en Github**

Para facilidad del estudiante, puede revisar el siguiente repositorio con los códigos utilizados en la presente guía: [https://github.com/samfag/Comunicacion\\_Inalambrica\\_ESP32.git](https://github.com/samfag/Comunicacion_Inalambrica_ESP32.git)