

# MANUAL TÉCNICO

(Microestación Meteorológica.)

## [Preparación Inicial]

**Proyecto 1: Estación  
meteorológica Openweather**  
**Proyecto 2: Instrumento  
meteorológico Thinkspeak**



## Índice del Manual Técnico

### 1. Preparación Inicial

- 2.1. Descargar e Instalar Arduino IDE
- 2.2. Instalación del Controlador USB a Serie CH340
- 2.3. Configuración de la Placa ESP32 en Arduino IDE
- 2.4. Registro en OpenWeatherMap y Obtención de la Clave API
- 2.5. Registro en ThingSpeak y Configuración del Canal

### 2. Proyecto 1: Estación Meteorológica con OpenWeatherMap

- 3.1. Esquema Eléctrico
- 3.2. Importar el Código Fuente
- 3.3. Configuración del Nombre y Contraseña WiFi
- 3.4. Actualización de la Clave API de OpenWeatherMap
- 3.5. Configuración de los Parámetros en Arduino IDE
- 3.6. Carga del Código y Visualización de Resultados

### 3. Proyecto 2: Instrumento Meteorológico con ThingSpeak

- 4.1. Esquema Eléctrico
- 4.2. Importar el Código Fuente
- 4.3. Configuración del Nombre y Contraseña WiFi
- 4.4. Actualización de la Clave API de ThingSpeak
- 4.5. Configuración de los Parámetros en Arduino IDE
- 4.6. Carga del Código y Visualización de Resultados

### 4. Solución de Problemas Comunes

- 5.1. Problemas de Conexión WiFi
- 5.2. Problemas con la Pantalla OLED
- 5.3. Problemas con las Claves API
- 5.4. Problemas con los Sensores
- 5.5. Problemas de Reconocimiento del ESP32 en Arduino IDE
- 5.6. Problemas de Envío de Datos a ThingSpeak
- 5.7. Problemas con la Carga del Código en el ESP32

### 5. Glosario

- 6.1. Microcontrolador ESP WROOM 32
- 6.2. Sensor DHT11
- 6.3. Sensor DHT22
- 6.4. Sensor BH1750
- 6.5. Sensor BMP180
- 6.6. Plataforma OpenWeatherMap



- 6.7. Plataforma ThingSpeak
- 6.8. CH340
- 6.9. Wi-Fi
- 6.10. API (Application Programming Interface)
- 6.11. Firmware
- 6.12. MQTT (Message Queuing Telemetry Transport)
- 6.13. Fuente de Alimentación

## 6. Contacto

- 8.1. Correo Electrónico de Soporte
- 8.2. Información del Desarrollador



## 1. Descargar Arduino IDE para Windows

Versión clásica recomendada 1.8.19:

Descargar URL: <https://downloads.arduino.cc/arduino-1.8.19-windows.exe>

## 2. Instale el controlador USB a serie CH340 para Windows

a) Necesitamos que la computadora reconozca el Puerto Serial de la Placa, instale el driver CH340.intente cada uno de los siguientes 2 métodos:

- Utilice el motor de búsqueda para encontrar el controlador CH340



Buscar con Google      Voy a tener suerte

Google disponible en: [Español \(Latinoamérica\)](#)

*Buscar en Google el Driver a instalar*

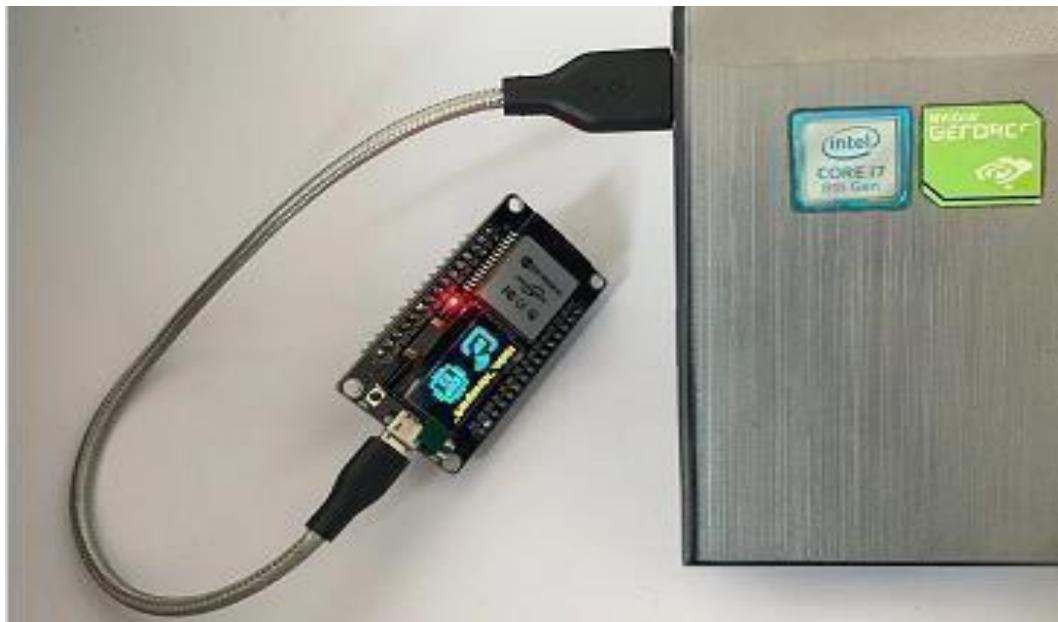
- O Visite el sitio web oficial del controlador CH340 para descargar  
<https://www.wch-ic.com/search?q=CH341SER&t=downloads>

file category	file content	version	upload time
Driver&Tools	CH340/CH341 USB to serial port One-Key installation VCP vendor driver for Windows, supports Windows 11/10/8.1/8/7/VISTA/XP/2000/98	3.8	2023-03-16
	CH340/CH341 USB to serial port Windows driver, supports Windows	3.8	2023-03-16

*En este sitio puedes instalar el driver (mejor opción)*

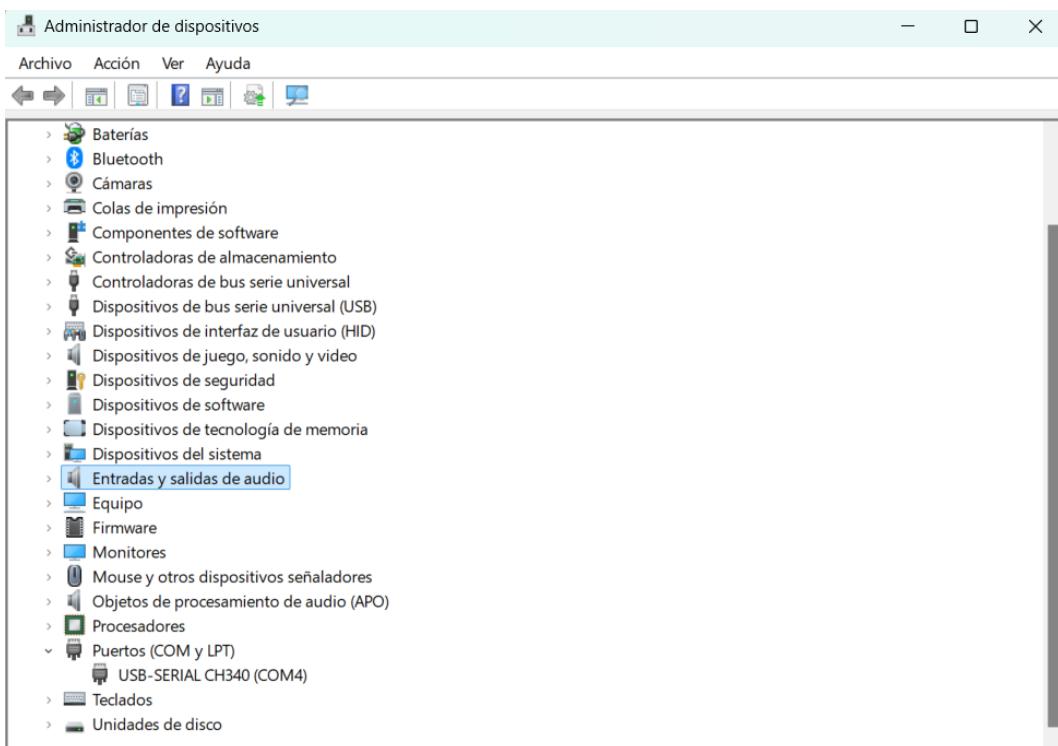


b) ESP32: O Visite el sitio web oficial del controlador CH340 para descargar



*Conectar el micro controlador a un puerto USB del computador*

c) Puede verlo en Mi PC -> Administrador de dispositivo



*Comprobar que si está reconociendo el puerto serial CH340*



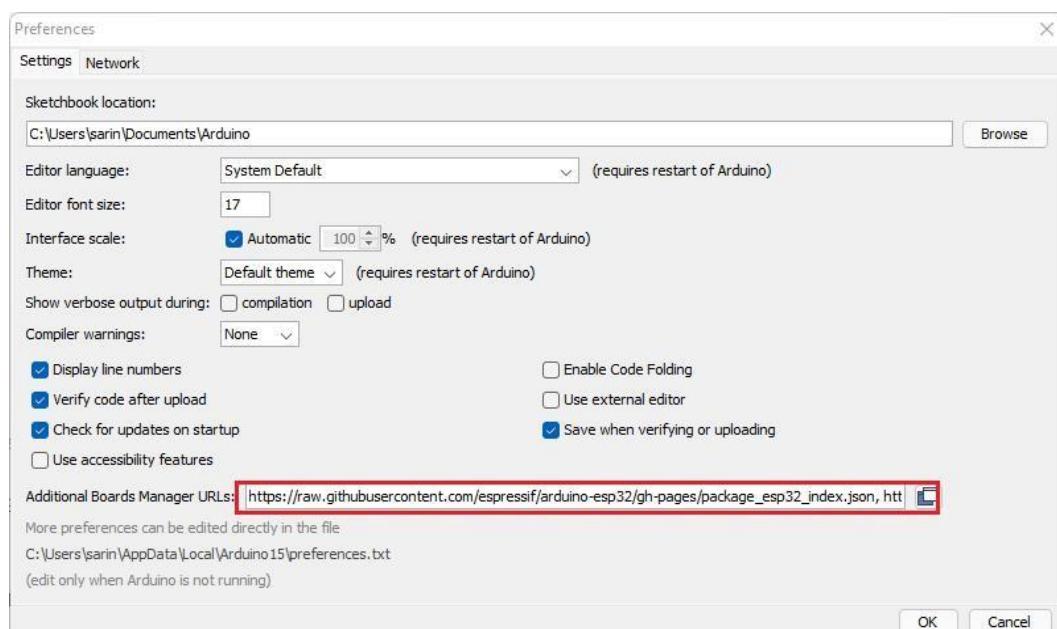
### 3. Configurar la placa ESP32 en Arduino IDE

- Abrir Arduino IDE
- Archivo -> Preferencias

Copie los 2 siguientes enlaces azules en el cuadro de texto «URLs adicionales del administrador de tableros:

[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)

[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)

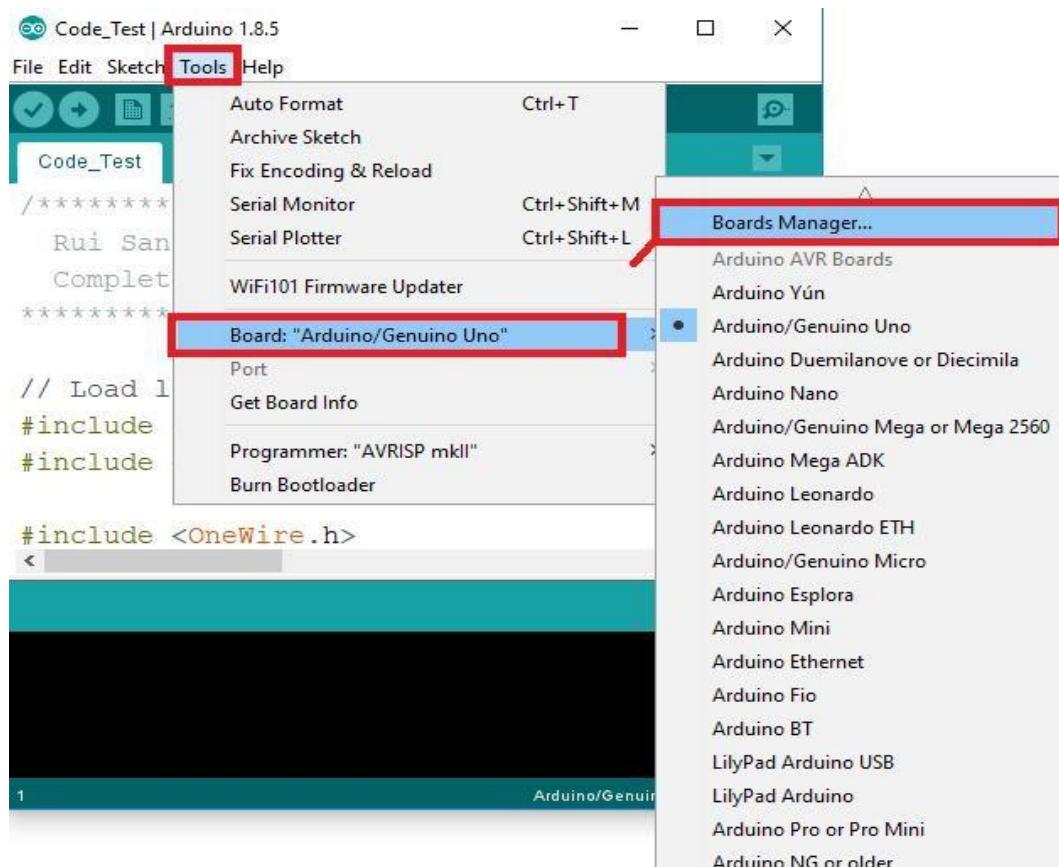


*Instalar las librerías necesarias*

- Haga clic en «Aceptar» para cerrar el cuadro de diálogo.

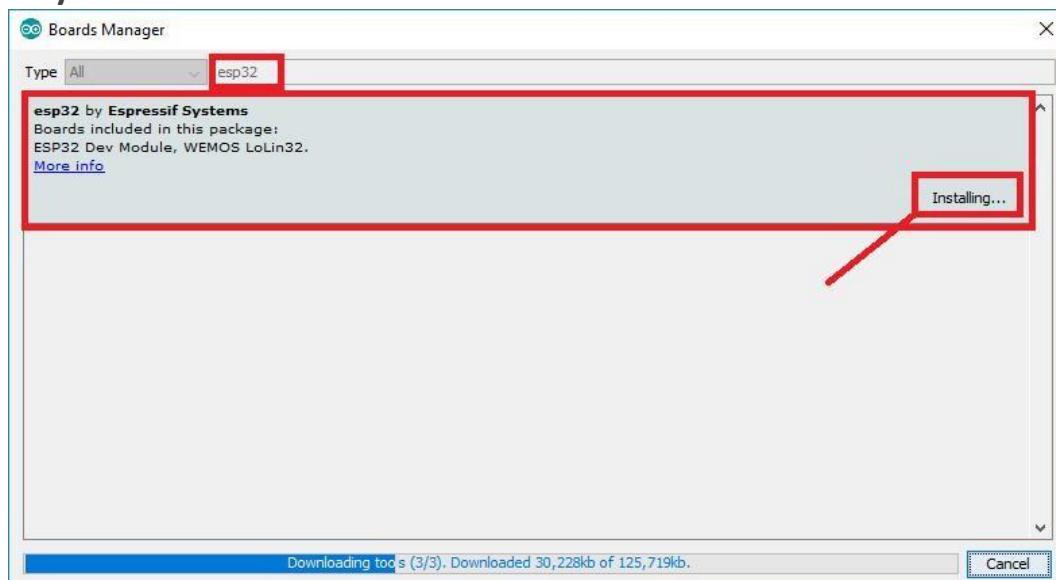


- d) Abra el Gestor de Pizarras. Vaya a **Herramientas > Tablero > Administrador de tableros...**



*Configurando para reconocer la placa de esp32*

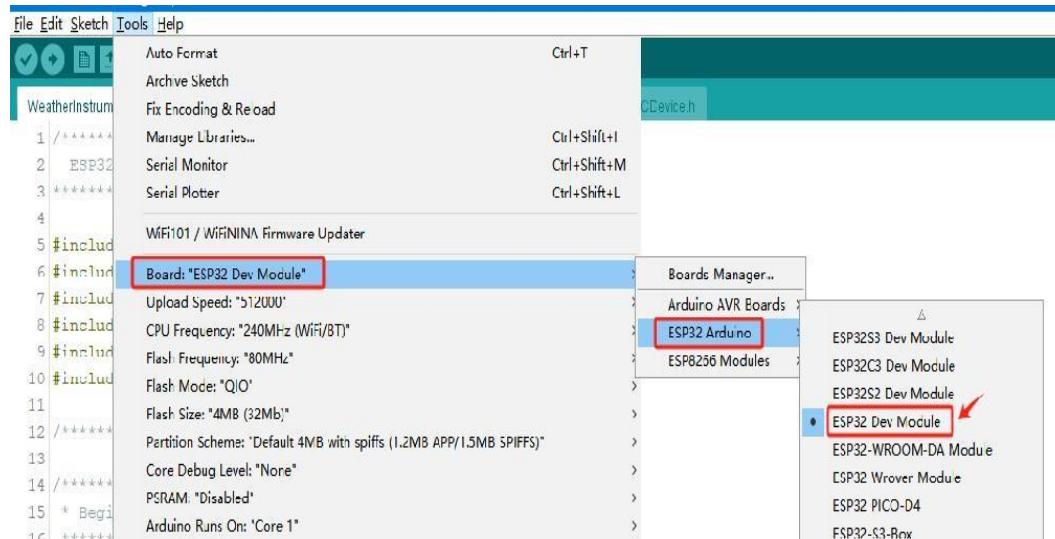
- e) Busque «**ESP32**» y pulse el botón de instalación para el «**ESP32 por Espressif Systems**»:



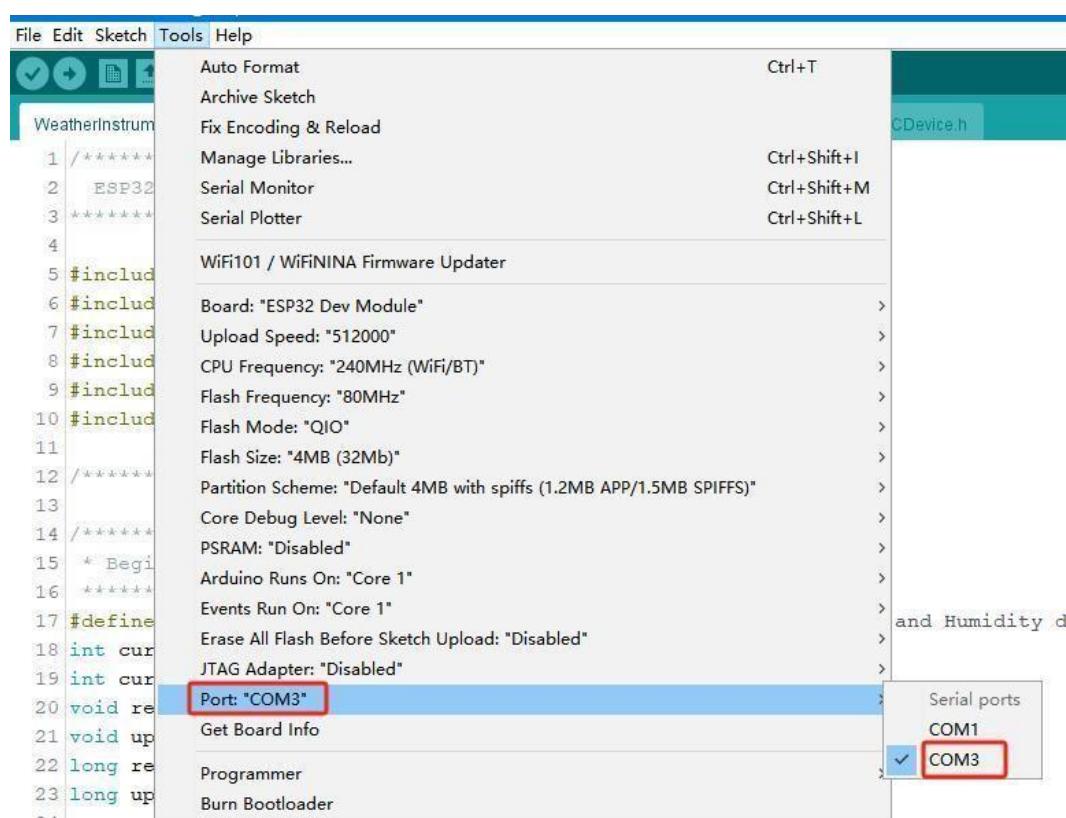
*Descargar librería del esp32 de Espressif Systems*



- f) Haga clic en el botón «Instalar» para instalar la **versión estable «2.0.11»**
- j) Una vez finalizado el proceso de instalación (puede tardar unos minutos), puede cerrar el cuadro de diálogo pulsando el botón «Cerrar».
- k) Vaya a **Herramientas > Placas**, Seleccione «**ESP32 Dev Modulo**

*Configurar la placa de ESP32 Dev Module*

- j) Elija el puerto correcto

*Puerto COM correcto donde está conectado la placa*



## 1) Registrar OpenWeathermap, obtener clave API

- a) Registrar una nueva cuenta OpenWeathermap:  
[https://home.openweathermap.org/users/sign\\_up](https://home.openweathermap.org/users/sign_up)

Create New Account

Username

Enter email

Password

Repeat Password

We will use information you provided for management and administration purposes, and for keeping you informed by mail, telephone, email and SMS of other products and services from us and our partners. You can proactively manage your preferences or opt-out of communications with us at any time using Privacy Centre. You have the right to access your data held by us or to request your data to be deleted. For full details please see the OpenWeather [Privacy Policy](#).

I am 16 years old and over

I agree with [Privacy Policy](#), [Terms and conditions of sale](#) and [Websites terms and conditions of use](#)

I consent to receive communications from OpenWeather Group of Companies and their partners:

*Regístrate para obtener la Api Key*

- b) Obtenga las claves API de OpenWeathermap para el “Proyecto 1: OpenWeatherMap”  
[https://home.openweathermap.org/api\\_keys](https://home.openweathermap.org/api_keys)

New Products Services API keys Billing plans Payments Block logs My orders My profile

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name	Status	Actions	Create key
fc22...		Default Active	<input checked="" type="button"/>	<input checked="" type="button"/>

API key name:

Generate

*Ten presente la Api Key para el proyecto 1 con Openweather*



## 2) Registrar ThingSpeak, establecer canal, obtener clave API

- a) Regístrate en ThingSpeak con una nueva cuenta y, a continuación, inicie sesión:

<https://thingspeak.com/>

To use ThingSpeak, you must sign in with your existing MathWorks account or create a new one.

Non-commercial users may use ThingSpeak for free. Free accounts offer limits on certain functionality. Commercial users are eligible for a time-limited free evaluation. To get full access to the MATLAB analysis features on ThingSpeak, log in to ThingSpeak using the email address associated with your university or organization.

To send data faster to ThingSpeak or to send more data from more devices, consider the [paid license options](#) for commercial, academic, home and student usage.

Create MathWorks Account

Email Address

Location

First Name

Last Name

Continue

Cancel

This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

*Creación de la cuenta en ThinkSpeak necesario para el proyecto 2*

- b) Cree un nuevo canal y reciba los datos de **temperatura, humedad, luz y atmósfera** del «Proyecto 2: Instrumento meteorológico». del «Proyecto 2: Instrumento Meteorológico». (EnableField1 y nombre «**Temperatura**» EnableField2 y nombre «**Humedad**» EnableField3 y nombre «**Luz**» EnableField4 y nombre «**Atmósfera**»)

<https://thingspeak.com/channels/new>

Channel Settings

Percentage complete: 50%

Channel ID: 549976

Author: ideaspark

Access: Public

ESP8266 and DHT11

Name: ESP8266-DHT11

Description: ESP8266 and DHT11

Field 1: Temperature

Field 2: Humidity

Field 3: Light

Field 4: Atmosphere

Field 5:

Field 6:

Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- **Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- **Tags:** Enter keywords that identify the channel. Separate tags with commas.
- **Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- **Show Channel Location:**
  - **Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
  - **Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.

*Crear los campos para poder graficar todos los datos*

**c) Poner el canal en Público**

Channel ID: 558971  
Author: ideaspark  
Access: Private

Sharing

Channel Sharing Settings

- Keep channel view private
- Share channel view with everyone
- Share channel view only with the following users:

Email Address Enter email here Add User

Help

ThingSpeak allows you to control who can view the data in your channel. Irrespective of the settings on this tab, reading data from or writing data to the fields of a channel requires the appropriate API key for the channel.

Channel Sharing Settings

- Keep channel view private: Selecting this option keeps your channel private. Only you will be able to see the channel view.
- Share channel view with everyone: Selecting this option makes the public view of your channel viewable by anyone browsing the ThingSpeak website.
- Share channel view only with the following users: Selecting this option shares the private view of your channel only with specific ThingSpeak users.

*Configura el canal en “share channel view with everyone”*

**d) Obtenga la clave API de escritura del canal para el «Proyecto 2: ThinkSpeak».**

Channel ID: 558971  
Author: ideaspark  
Access: Public

API Keys

Write API Key

Key D7IT [REDACTED] 3F

Generate New Write API Key

Read API Keys

Key HH [REDACTED] DD

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- Write API Key: Use this key to write data to a channel. If you feel your key has been compromised, click [Generate New Write API Key](#).
- Read API Keys: Use this key to allow other people to view your private channel feeds and charts. Click [Generate New Read API Key](#) to generate an additional read key for the channel.
- Note: Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

API Requests

Update a Channel Feed

*La llave de escritura será importante para configurar el proyecto 2 “Write API key”*

g

**Proyecto 1**

**[OpenWeatherMap]**



## 1. Esquema eléctrico

ESP32 0.96'' OLED Board	Computador
Micro USB Port	USB Port

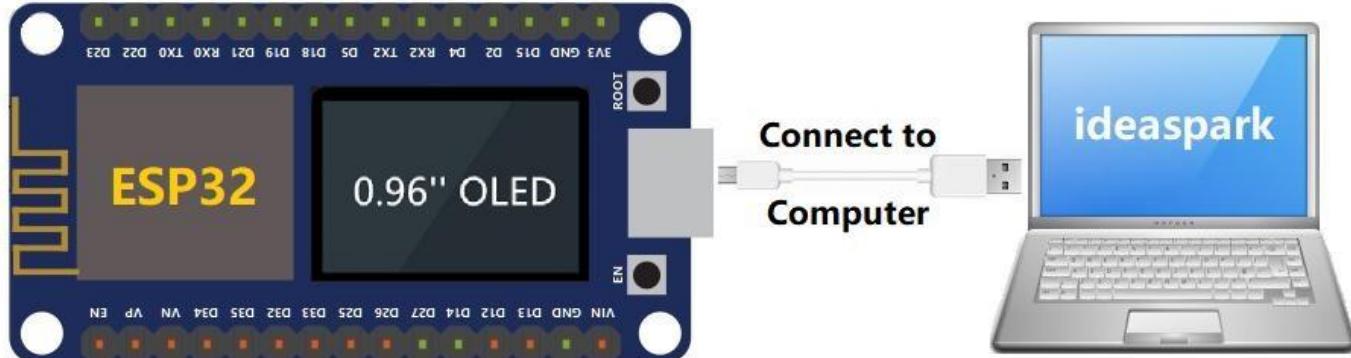


Imagen gráfica de conexión

### 1. Importar el código fuente de la estación meteorológica

a) Busque el archivo «.ino» en la carpeta 00\_DocumentacionMicroEstacionMeteorologica /Códigos(Sketch) /documentacionEstacionMeteorologica:

Archivo: **sketch-project-1-openweathermap.ino**

OpenWeatherMapForecast	21/01/2025 9:41 a. m.	Archivo de origen ...
OpenWeatherMapOneCall	21/01/2025 9:41 a. m.	Archivo de origen ...
OpenWeatherMapOneCall	21/01/2025 9:41 a. m.	Archivo de origen ...
sketch-project-1-openweathermap	21/01/2025 9:41 a. m.	Arduino file
SSD1306Wire	21/01/2025 9:41 a. m.	Archivo de origen ...
SunMoonCalc	21/01/2025 9:41 a. m.	Archivo de origen ...
SunMoonCalc	21/01/2025 9:41 a. m.	Archivo de origen ...

Busca el archivo: **sketch-project-1-openweathermap.ino**

a) Importar sketch-project-1-openweathermap.ino a Arduino IDE

```

1 // ****
2   ESP32 Weather Station
3 ****/
4 #include <Arduino.h>
5 #include <WiFi.h>
6 #include "JsonListener.h"
7 #include <time.h>
8 #include <sys/time.h>
9

```

Importar el sketch-project-1-openweathermap

(Verá que el archivo principal «sketch-project-1-openweathermap.ino» y otros archivos de referencia se importan automáticamente al IDE")



## 2. Actualiza tu nombre y contraseña WiFi

- a) Localice la pestaña «sketch-project-1-openweathermap», Sustituya el “\*\*\*\*\*” en el WiFi\_Name por su nombre WiFi Sustituya el “\*\*\*\*\*” en el WiFi\_Password por su contraseña de WiFi:

```
sketch-project-1-openweathermap Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda
sketch-project-1-openweathermap.ino AerisForecasts.cpp AerisForecasts.h AerisObservations.cpp AerisObservations.h
22 //*****
23 * WIFI Settings
24 *****/
25 const char* WiFi_Name = "*****"; /* Nombre de la red WIFI */
26 const char* WiFi_Password = "*****"; /* Contraseña de red WIFI */
27
```

*Actualizar y poner el nombre de la red WI-FI además de la contraseña*

## 3. Actualizar las claves API de OpenWeathermap

- a) Localice la pestaña «sketch-project-1-openweathermap.ino», Sustituya “\*\*\*\*\*” en el OPEN\_WEATHER\_MAP\_APP\_ID por las claves API de OpenWeathermap que registró en los preparativos anteriores.

```
sketch-project-1-openweathermap Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda
sketch-project-1-openweathermap.ino AerisForecasts.cpp AerisForecasts.h AerisObservations.cpp AerisObservations.h AerisSunMoon.cpp AerisSun...
27
28 //*****
29 * Configuración de OpenWeatherMap
30 * Regístrate aquí para obtener una clave API: https://docs.thingpulse.com/how-tos/openweathermap-key/
31 *****/
32 const boolean IS_METRIC = true;
33 String OPEN_WEATHER_MAP_APP_ID = "*****"; /* Actualizar la api keys de openWeathermap */
34
```

*Copiar la clave API de Openweather*

- b) Localice el número en el link de la ciudad que busques o necesites  
Ejemplo: <https://openweathermap.org/city/3687952>

openweathermap.org/city/3687952

Weather in your city      Guide    API    Dashboard    Marketplace    Pricing

caldas      Search

Jan 21, 11:06am  
**Caldas, CO**

22°C



*Busca el municipio que necesitas y copia el número del link city/**368752***



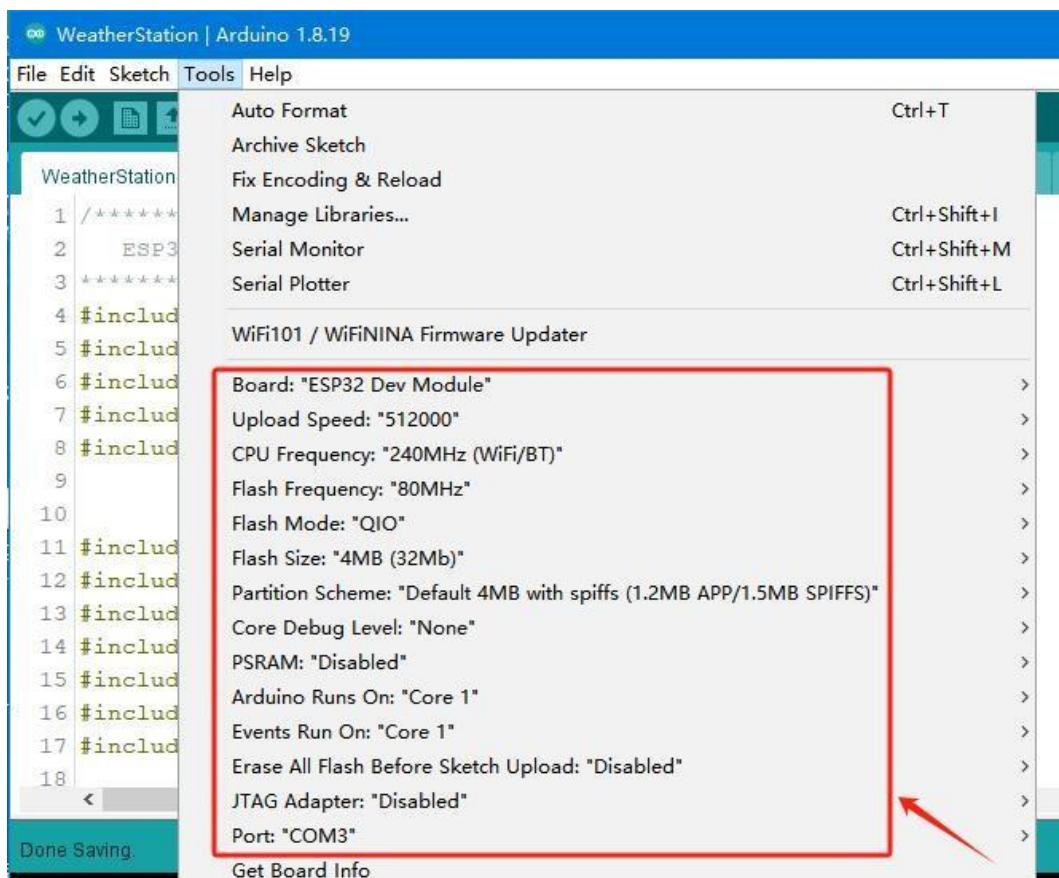
El número 3687952 y es el ID de caldas, va en OPEN\_WEATHER\_MAP\_LOCATION\_ID = "\*\*\*\*\*";

```
sketch-project-1-openweathermap Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda
sketch-project-1-openweathermap $ AerisForecasts.cpp AerisForecasts.h AerisObservations.cpp AerisObservations.h AerisSunMoon.cpp AerisSur
34
35 /*Vaya a https://openweathermap.org/find?q= y busque una ubicación. Recorra el
36 resultado y seleccione la entrada más cercana a la ubicación real que desea mostrar
37 datos. Será una URL como https://openweathermap.org/city/2657896. El número
38 al final es lo que se asigna a la constante a continuación.
39 */
40
41 String OPEN_WEATHER_MAP_LOCATION_ID = "*****"; /* Actualizar region, municipio para datos meteorologicos */
```

Pon el ID del municipio o región

## 1. Configurar el parámetro Arduino IDE

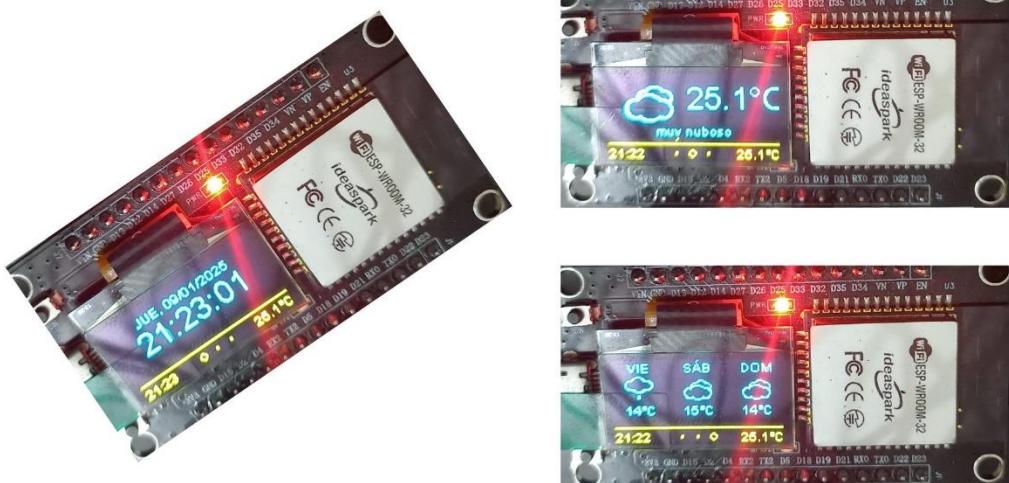
- a) Configure su IDE Arduino de acuerdo con los ajustes que se muestran en la siguiente figura:



Configura estos valores en Tools



## 2. Cargue el código y vea los resultados



## 3. Note

- a) Si la información de la pantalla ESP32 sigue atascado durante las previsiones de actualización:



*Cargando código al microcontrolador*



*Los datos de la previsión meteorológica están vacíos*



*Los datos de la previsión meteorológica están vacíos*

podría ser:

- a. La señal WiFi es débil, acerque la placa al router de la red.
- b. Congestión de la red. Por favor, cambie a una nueva red
- c. La clave API de OpenWeathermap no está disponible

# Proyecto 2

# [ThingSpeak]

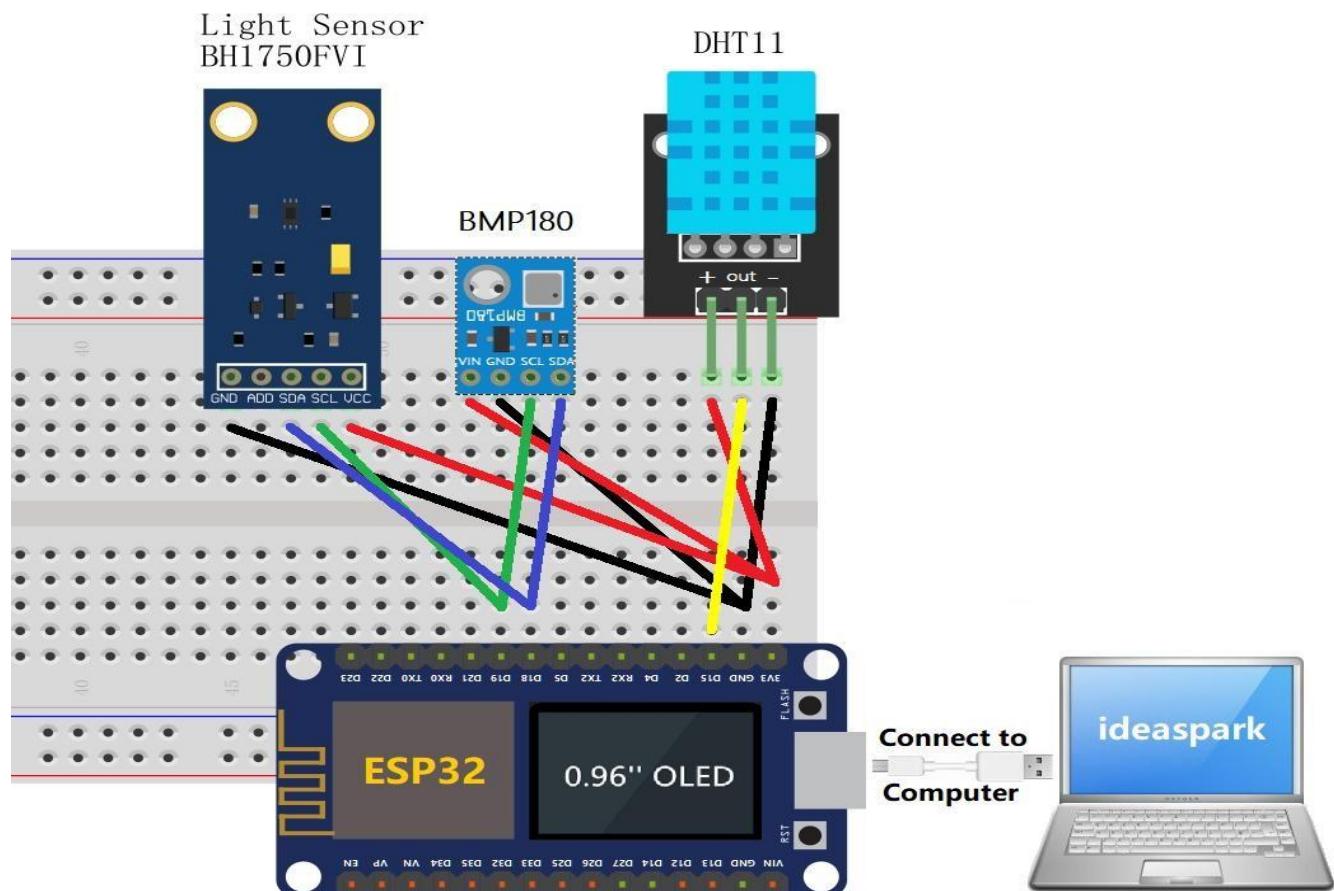


## ➤ Esquema eléctrico

ESP32 0.96'' OLED Board	DHT11 Sensor de temperatura y humedad
VCC(3.3V)	VCC
GND	GND
D15	OUT

ESP32 0.96'' OLED Board	BMP180 Sensor de atmósfera
VCC(3.3V)	VCC
GND	GND
D19	SCL
D18	SDA

ESP32 0.96'' OLED Board	BH1750 Ajustes del sensor de luz
VCC(3.3V)	VCC
GND	GND
D19	SCL
D18	SDA

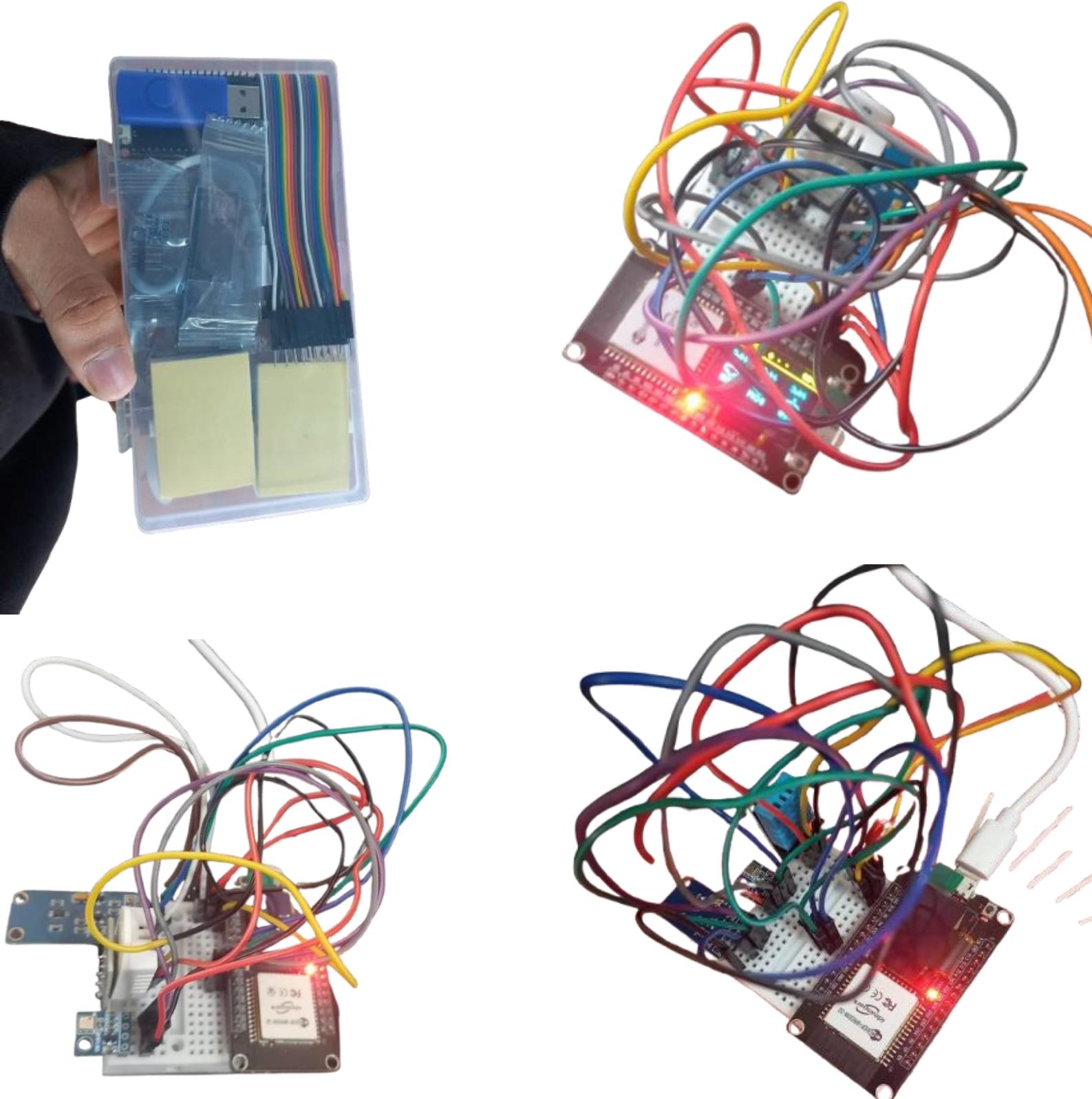


Esquema de conexiones del proyecto 2



Kit para armar la estación meteorológica en su hardware (componentes físicos) además de ejemplos de cómo se ve armado.

**Video Tutorial para armarlo:** [https://youtu.be/Gx2EkeZB\\_A](https://youtu.be/Gx2EkeZB_A)



*Imágenes referentes a su parte física armado*



## ➤ Importar el código fuente del instrumento meteorológico

- Busque el archivo «00\_DocumentacionMicroEstacionMeteorologica /Codos(Sketch) /sketch\_project\_2\_ThinkSpeak.ino» en la carpeta documentacionEstacionMeteorologica:

Nombre	Fecha de modificación	Tipo	Tamaño
sketch_project_2_ThinkSpeak	21/01/2025 1:45 p. m.	Arduino file	5 KB

*Importar el sketch\_project\_2\_ThinkSpeak.ino a Arduino IDE*

- Importar sketch\_project\_2\_ThinkSpeak.ino a Arduino IDE

```
sketch_project_2_ThinkSpeak.ino
1 #include <Wire.h>
2 #include <Adafruit_BMP085.h>
3 #include "DHT.h"
4 #include <BH1750.h>
5 #include <WiFi.h>
6 #include <HTTPClient.h>
7
8 // Credenciales de tu red WiFi
9 const char* ssid = "*****"; /* Nombre de la red WIFI */
10 const char* password = "*****"; /* Contraseña de red WIFI */
```

*Código ya importado en Arduino IDE*

## ➤ Actualice su Nombre WiFi y Contraseña WiFi

- Localice la pestaña «sketch\_project\_2\_ThinkSpeak.ino», Sustituya “\*\*\*\*\*” en el WiFi\_Name por su nombre WiFi Sustituya “\*\*\*\*\*” en el WiFi\_Password por contraseña WiFi

```
sketch_project_2_ThinkSpeak.ino
8 // Credenciales de tu red WiFi
9 const char* ssid = "*****"; /* Nombre de la red WIFI */
10 const char* password = "*****"; /* Contraseña de red WIFI */
```

*Cambiar por el nombre de tu red WiFi y contraseña*



## ➤ Actualizar la clave API de ThingSpeak Write

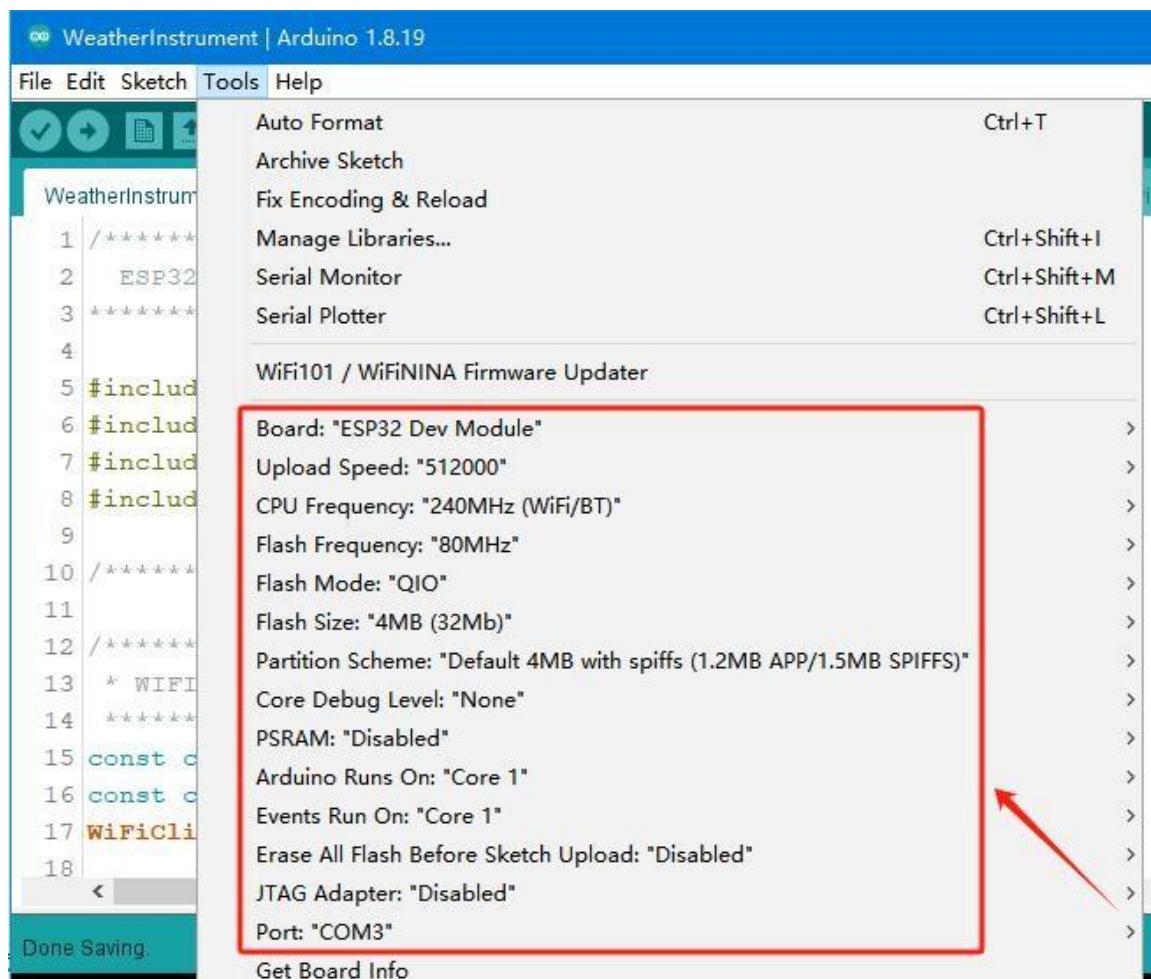
- Localice la pestaña “**sketch\_project\_2\_ThinkSpeak.ino**”,  
Reemplace “\*\*\*\*\*” en thingspeak\_API\_key con su Clave API de Escritura de Canal ThingSpeak que registró en las preparaciones previas.

```
sketch_project_2_ThinkSpeak Arduino 1.8.19
Archivo Editar Programa Herramientas Ayuda
sketch_project_2_ThinkSpeak §
12 // Configuración de ThingSpeak
13 const char* server = "http://api.thingspeak.com";
14 String apiKey = "*****"; /* Write API Key del channel de thinkspeak */
15

Cambiar por la Write API Key de ThinkSpeak
```

## ➤ Configurar el parámetro Arduino IDE

- Configure su IDE Arduino de acuerdo con los ajustes que se muestran en la siguiente figura:



Configurar estos valores en Tools



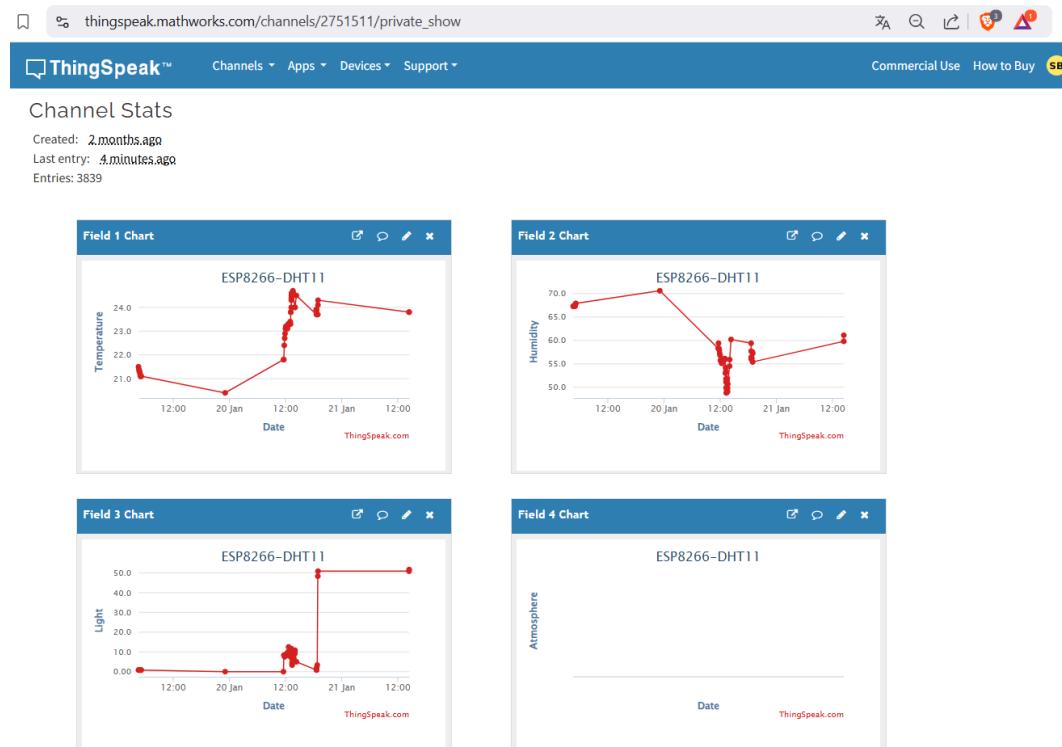
## ➤ Cargar el código y ver los resultados

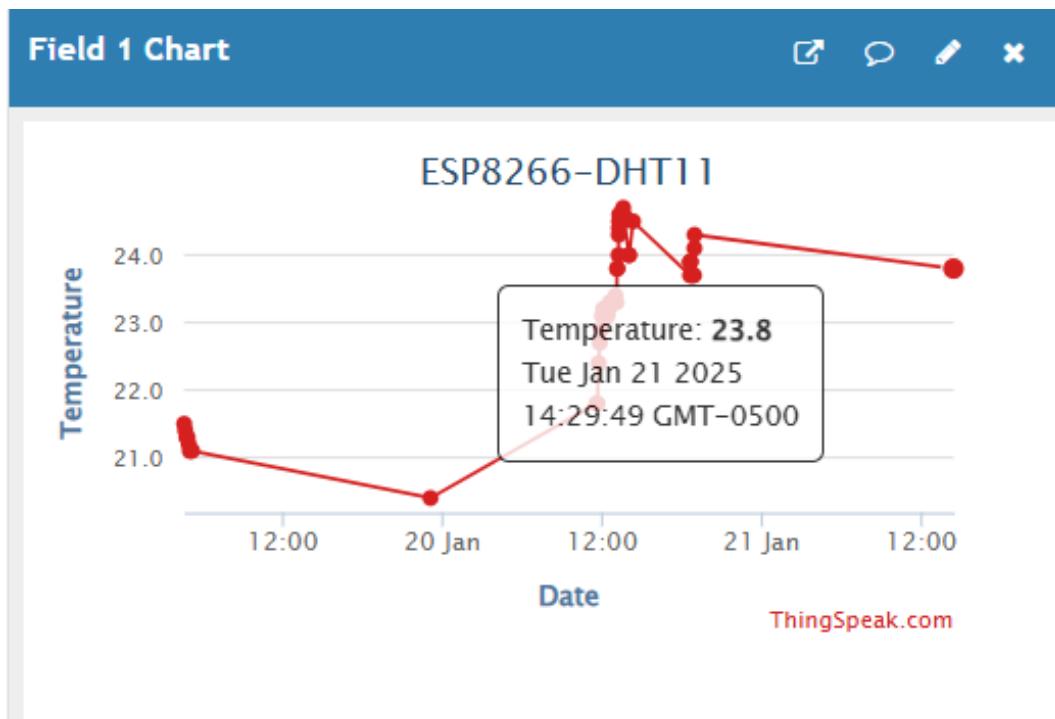
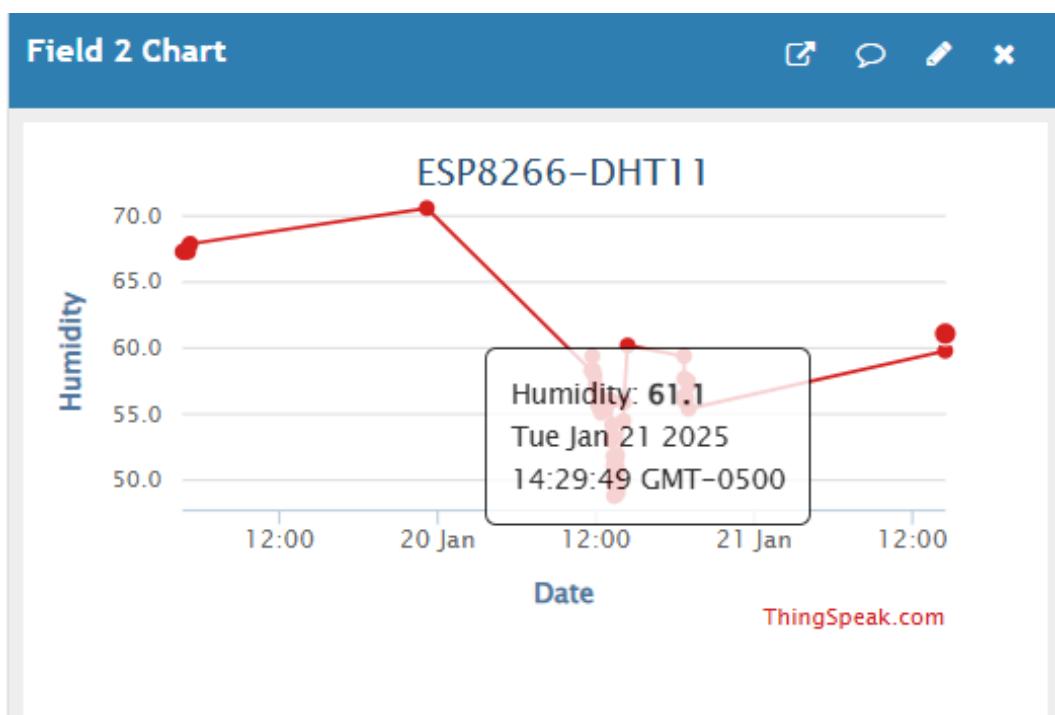
```

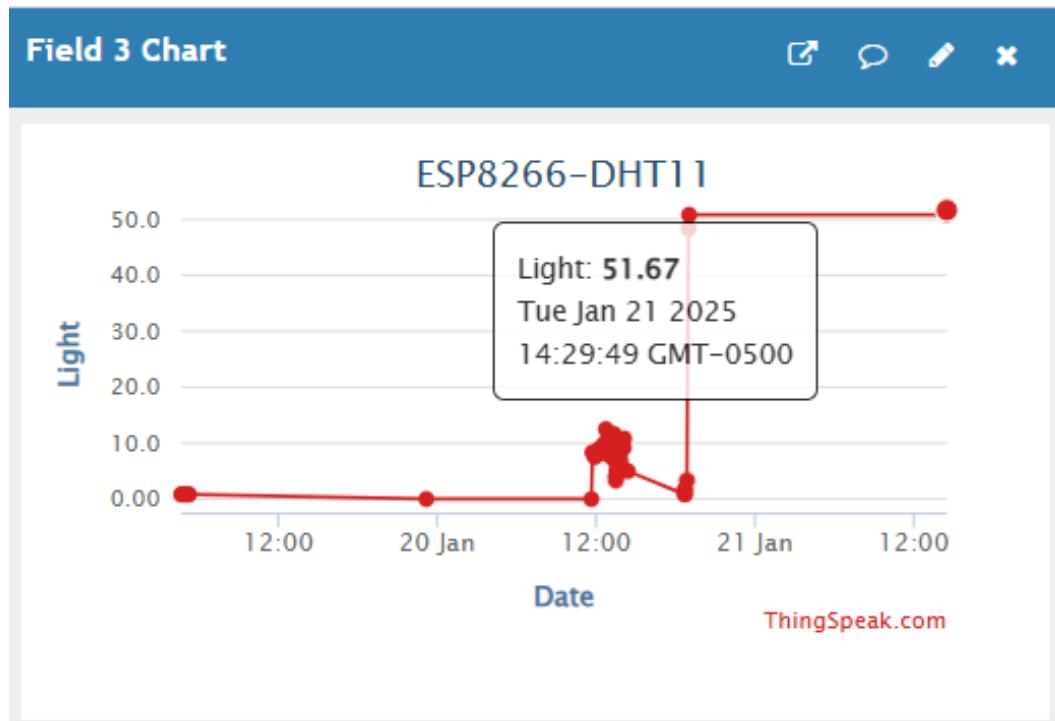
14:29:47.715 -> ets Jul 29 2019 12:21:46
14:29:47.715 ->
14:29:47.715 -> rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
14:29:47.715 -> configsip: 0, SPIWP:0xee
14:29:47.715 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
14:29:47.715 -> mode:DIO, clock div:1
14:29:47.715 -> load:0x3fff0030,len:1344
14:29:47.715 -> load:0x40078000,len:13964
14:29:47.715 -> load:0x40080400,len:3600
14:29:47.715 -> entry 0x400805f0
14:29:48.045 -> Error: Sensor BMP180 no detectado.
14:29:48.093 -> BH1750 inicializado correctamente.
14:29:48.093 -> Conectando a WiFi: villaventura0124
14:29:49.226 -> .
14:29:49.226 -> ;Conexión WiFi establecida!
14:29:49.226 -> Lecturas de los sensores:
14:29:49.226 -> Temperatura: 23.80 °C
14:29:49.273 -> Humedad: 61.10 %
14:29:49.273 -> Luz: 51.67 lx
14:29:49.273 -> Presión: nan Pa
14:29:49.699 -> Datos enviados correctamente a ThingSpeak.
14:29:49.699 -> Código HTTP: 200
14:29:49.699 -> Wi-Fi desconectado para ahorrar energía.

```

Autoscroll  Mostrar marca temporal  Nueva línea  115200 baudio  Limpiar salida

*Resultados de monitor serial**Graficas generales con datos*

*Grafica de temperatura**Grafica de humedad*

*Grafica de luz**Grafica de atmosfera*

# [Solución de Problemas Comunes]



## 1. ¿Qué hacer si el ESP32 no se conecta al WiFi?

### Posibles causas y soluciones:

- **Verifica que el nombre y la contraseña del WiFi estén correctamente escritos:**
  - Asegúrate de que el nombre de la red (SSID) y la contraseña estén correctamente ingresados en el código. Revisa que no haya espacios adicionales o caracteres especiales que puedan causar errores.
- **Asegúrate de que la red WiFi esté funcionando:**
  - Verifica que el router esté encendido y que la red WiFi esté activa. Intenta conectarte con otro dispositivo para confirmar que la red está disponible.
- **Acerca el ESP32 al router para mejorar la señal:**
  - Si la señal WiFi es débil, intenta mover el ESP32 más cerca del router o elimina obstáculos que puedan interferir con la señal.

---

## 2. ¿Qué hacer si no se muestran datos en la pantalla OLED?

### Posibles causas y soluciones:

- **Verifica las conexiones del sensor y la pantalla OLED:**
  - Asegúrate de que los cables estén correctamente conectados entre el ESP32, los sensores y la pantalla OLED. Revisa que no haya cables sueltos o mal conectados.
- **Asegúrate de que el código esté correctamente cargado en el ESP32:**
  - Verifica que el código se haya cargado correctamente en el ESP32. Si hay errores durante la carga, revisa el mensaje de error en el IDE de Arduino y corrige cualquier problema en el código.
- **Revisa la configuración de la pantalla OLED en el código:**
  - Asegúrate de que la dirección I2C de la pantalla OLED esté correctamente configurada en el código. Si es necesario, utiliza un escáner I2C para verificar la dirección correcta.

---

## 3. ¿Qué hacer si la clave API no funciona?

### Posibles causas y soluciones:

- **Verifica que la clave API esté correctamente copiada y pegada:**
  - Asegúrate de que la clave API esté correctamente copiada y pegada en el código. Revisa que no haya espacios adicionales o caracteres faltantes.



- **Asegúrate de que la clave API esté activa y no haya expirado:**
  - Algunas claves API tienen un período de validez limitado. Verifica en la plataforma de OpenWeatherMap o ThingSpeak que la clave API esté activa y no haya expirado. Si es necesario, genera una nueva clave API.
- **Verifica el límite de uso de la API:**
  - Algunas APIs tienen un límite de solicitudes por día. Si has excedido el límite, es posible que la API deje de funcionar temporalmente. Revisa el uso de la API en la plataforma correspondiente y espera a que se restablezca el límite.

## 4. ¿Qué hacer si los sensores no están proporcionando datos correctos?

Posibles causas y soluciones:

- **Verifica la alimentación de los sensores:**
  - Asegúrate de que los sensores estén correctamente alimentados. Algunos sensores requieren un voltaje específico (por ejemplo, 3.3V o 5V). Revisa las especificaciones del sensor y asegúrate de que esté recibiendo el voltaje correcto.
- **Revisa las conexiones de los sensores:**
  - Asegúrate de que los cables estén correctamente conectados y que no haya cortocircuitos o conexiones sueltas. Revisa también que los pines de conexión (SCL, SDA, VCC, GND) estén correctamente asignados en el código.
- **Calibra los sensores si es necesario:**
  - Algunos sensores, como el BMP180 (presión atmosférica) o el DHT11 (temperatura y humedad), pueden requerir calibración para obtener lecturas precisas. Consulta la documentación del sensor para realizar la calibración adecuada.

## 5. ¿Qué hacer si el ESP32 no se reconoce en el Arduino IDE?

Posibles causas y soluciones:

- **Verifica que el controlador CH340 esté correctamente instalado:**
  - Asegúrate de que el controlador CH340 esté correctamente instalado en tu computadora. Si no está instalado, descárgalo e instálalo desde el sitio web oficial.
- **Revisa el puerto COM seleccionado:**
  - En el Arduino IDE, asegúrate de que el puerto COM correcto esté seleccionado en **Herramientas > Puerto**. Si no ves el puerto COM, intenta desconectar y volver a conectar el ESP32.



- **Reinstala las librerías del ESP32:**

- Si el ESP32 no se reconoce, reinstala las librerías del ESP32 en el Arduino IDE. Ve a **Herramientas > Placa > Administrador de tarjetas** y busca "ESP32". Luego, instala la versión más reciente.
- 

## 6. ¿Qué hacer si los datos no se están enviando a ThingSpeak?

### Posibles causas y soluciones:

- **Verifica la clave API de ThingSpeak:**

- Asegúrate de que la clave API de ThingSpeak esté correctamente copiada y pegada en el código. Revisa que no haya errores de escritura.

- **Revisa la conexión a Internet:**

- Asegúrate de que el ESP32 esté conectado a Internet y que la red WiFi esté funcionando correctamente. Si la conexión es débil, intenta mover el ESP32 más cerca del router.

- **Verifica los campos en ThingSpeak:**

- Asegúrate de que los campos en ThingSpeak estén correctamente configurados y que coincidan con los nombres de los campos en el código. Si es necesario, revisa la configuración del canal en ThingSpeak.
- 

## 7. ¿Qué hacer si la pantalla OLED muestra caracteres extraños o no se enciende?

### Posibles causas y soluciones:

- **Verifica la conexión I2C:**

- Asegúrate de que los cables SDA y SCL estén correctamente conectados entre la pantalla OLED y el ESP32. Revisa que no haya cables cruzados o mal conectados.

- **Revisa la dirección I2C:**

- Algunas pantallas OLED tienen direcciones I2C diferentes. Asegúrate de que la dirección I2C en el código coincida con la dirección real de la pantalla. Puedes usar un escáner I2C para encontrar la dirección correcta.

- **Verifica la alimentación de la pantalla:**

- Asegúrate de que la pantalla OLED esté recibiendo el voltaje correcto (generalmente 3.3V o 5V). Si la pantalla no se enciende, revisa las conexiones de alimentación.
-



## 8. ¿Qué hacer si el código no se carga en el ESP32?

### Potenciales causas y soluciones:

- **Verifica que el ESP32 esté en modo de carga:**
  - Para cargar el código en el ESP32, es necesario que el microcontrolador esté en modo de carga. Esto generalmente se hace manteniendo presionado el botón "BOOT" mientras se conecta el ESP32 al computador.
- **Revisa el puerto COM:**
  - Asegúrate de que el puerto COM correcto esté seleccionado en el Arduino IDE. Si no ves el puerto COM, intenta desconectar y volver a conectar el ESP32.
- **Reinstala las librerías del ESP32:**
  - Si el código no se carga, reinstala las librerías del ESP32 en el Arduino IDE. Ve a **Herramientas > Placa > Administrador de tarjetas** y busca "ESP32". Luego, instala la versión más reciente.

# [Glosario]



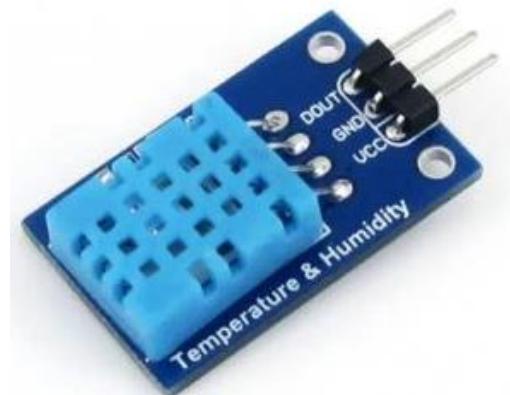
## ➤ Glosario

- **Microcontrolador ESP WROOM 32:** Módulo de microcontrolador basado en el ESP32, con conectividad Wi-Fi y Bluetooth integrada. Es utilizado para proyectos de IoT debido a su bajo consumo de energía y su capacidad de procesamiento.



*Microcontrolador ESP WROOM 32*

- **Sensor DHT11:** Sensor de temperatura y humedad de bajo costo y precisión media. Ideal para aplicaciones básicas de monitoreo ambiental.



*Sensor DHT11*

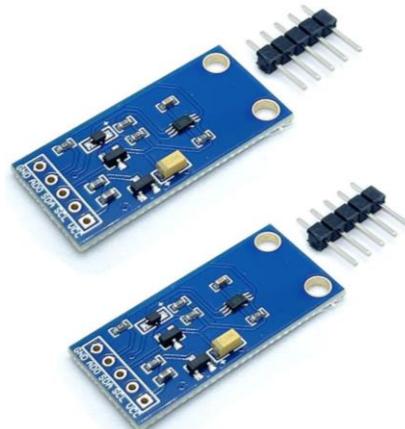
- **Sensor DHT22:** Versión mejorada del DHT11, con mayor precisión y rango de medición más amplio para temperatura y humedad.



*Sensor DHT22*

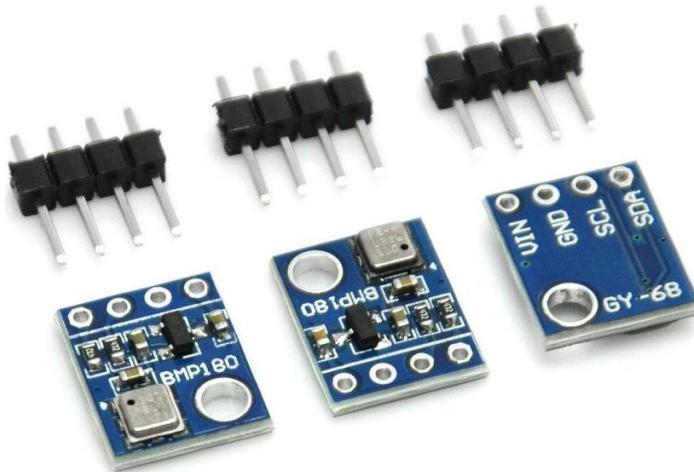


- **💡 Sensor BH1750:** Sensor digital de luz que mide la intensidad de la luz en lux. Utilizado para monitorear la luminosidad del entorno.



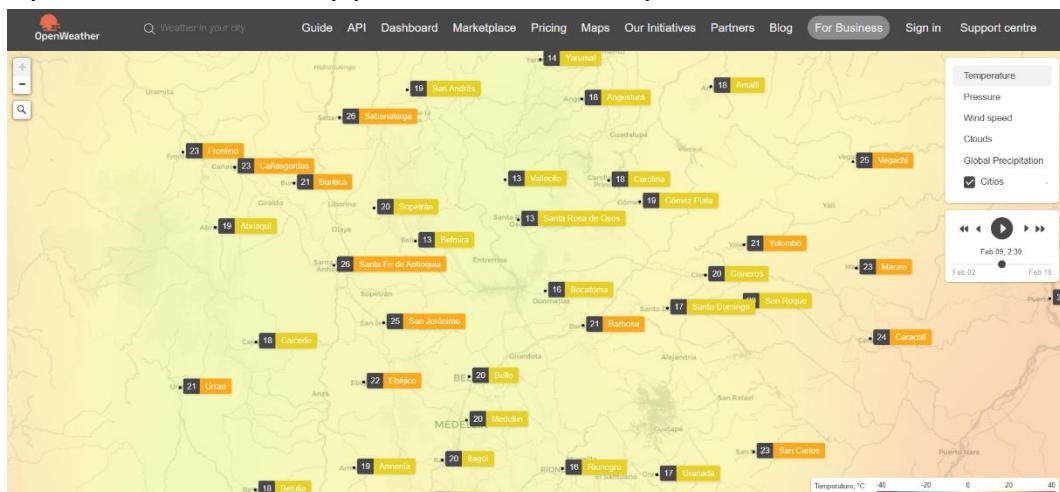
*Sensor BH1750*

- **🌡 Sensor BMP180:** Sensor de presión barométrica y temperatura. Permite medir la presión atmosférica y estimar la altitud relativa.



*Sensor BMP180*

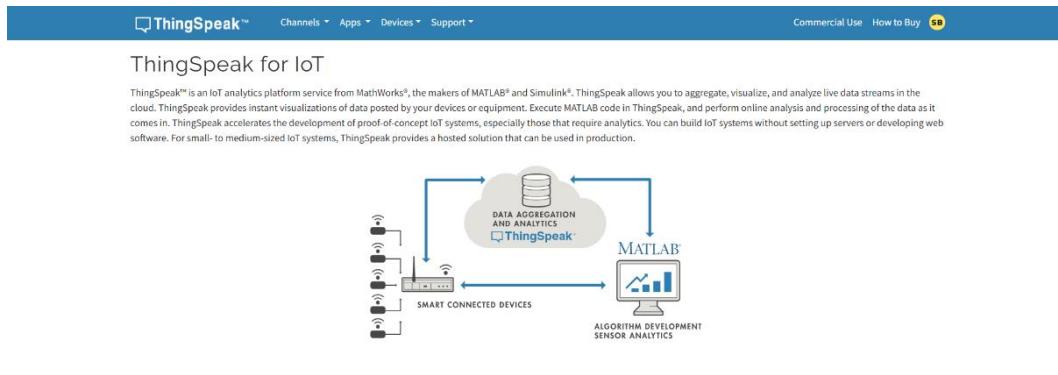
- **🌐 Plataforma OpenWeatherMap:** Servicio en línea que proporciona datos meteorológicos, como temperatura, humedad y pronóstico del tiempo, mediante APIs.



*Plataforma OpenWeatherMap*



-  **Plataforma ThingSpeak:** Plataforma de análisis y visualización de datos en tiempo real, diseñada para aplicaciones IoT. Permite recopilar, analizar y mostrar datos de sensores conectados a Internet.



#### ThingSpeak Key Capabilities

ThingSpeak allows you to aggregate, visualize and analyze live data streams in the cloud. With ThingSpeak, your data is stored in channels. Each channel stores up to 8 fields of data. You can create as many channels as you need for your application.

#### *Plataforma ThingSpeak*

-  **CH340:** Convertidor USB a serie utilizado para facilitar la comunicación entre el microcontrolador y el puerto USB del ordenador. Es comúnmente necesario para cargar programas en microcontroladores.

file category	file content	version	upload time
Driver&Tools	CH341SER.EXE CH340/CH341 USB to serial port One-Key installation VCP vendor driver for Windows, supports Windows 11/10/8.1/8/7/VISTA/XP/2000/98	3.8	2023-03-16
	CH341SER.ZIP CH340/CH341 USB to serial port Windows driver, supports Windows	3.8	2023-03-16

#### *Driver CH340*

-  **Wi-Fi:** Tecnología inalámbrica que permite la transmisión de datos entre dispositivos a través de una red local o Internet.
-  **API (Application Programming Interface - Interfaz de Programación de Aplicaciones):** Conjunto de definiciones y protocolos que permiten que diferentes aplicaciones se comuniquen entre sí.
-  **Firmware:** Software específico para el hardware que controla el funcionamiento del dispositivo. En el caso del ESP32, el firmware es el programa que ejecuta el microcontrolador.



- **MQTT (Message Queuing Telemetry Transport):** Protocolo de comunicación ligero para el intercambio de mensajes en dispositivos IoT.
- **⚡ Fuente de alimentación:** Dispositivo que proporciona la energía eléctrica necesaria para que la micro estación meteorológica funcione correctamente.

# [Requisitos de software y Hardware]



## ➤ Requisitos de Software y Hardware

---

### 1. Requisitos de Hardware

#### ➤ Lista de Componentes

A continuación, se detalla la lista de componentes necesarios para la implementación de la microestación meteorológica:

- **Microcontrolador:** ESP32 Dev Module
- **DHT11 / DHT22** (Temperatura y Humedad)
- **BH1750** (Sensor de Luz)
- **BMP180** (Presión Atmosférica y Altitud)

#### ➤ Conexiones y Accesorios:

- **Cables Dupont** (Hembra-Hembra, Hembra-Macho).
- **Mini-Protoboard** para conexiones.
- **Fuente de Alimentación:** Power Bank TL-PB10000

---

### 2. Requisitos de Software

#### ➤ Entorno de Desarrollo

- **Arduino IDE** (Versión 1.8.19 o superior).
- **Configuración:** Agregar la URL del gestor de tarjetas ESP32:  
[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json).

#### ○ **Librerías necesarias:**

- *DHT.h* (Para sensores DHT11/DHT22).
- *Adafruit BMP085.h* (Para el sensor BMP180).
- *BH1750.h* (Para el sensor de luz).
- *Adafruit SSD1306.h* (Para la pantalla OLED).
- *WiFi.h* (Para la conectividad).
- *HTTPClient.h* (Para el envío de datos a la nube).
- *Openweather by Bodmer*
- *ESP8266 and ESP32 OLED driver for SSD1306 displays*
- *DHT sensor library*
- *BH1750VI*
- *BH1750*
- *ArtronShop\_BH1750*
- *ArduinoJson*
- *AM2302-Sensor*
- *Adafruit Unified Sensor*

[santigobedoyacorrales@gmail.com](mailto:santigobedoyacorrales@gmail.com)

Tel: 3102532893



- Adafruit SSD1306

➤ **Controladores:**

- **CH340:** Necesario para reconocimiento del ESP32 en Windows.

➤ **APIs y Plataformas**

- **OpenWeatherMap API:** Para obtener datos meteorológicos externos.

- **ThingSpeak API:** Para almacenamiento, visualización y análisis de datos.

---

➤ **Pruebas y Validación**

- **Prueba de conectividad WiFi:** Verificar conexión estable a la red.

- **Prueba de sensores:** Comprobar que cada sensor entrega datos correctos.

- **Prueba de transmisión de datos:** Validar envío exitoso a ThingSpeak.

- **Prueba de transmisión de datos:** Validar la visualización de OpenWeatherMap

- **Prueba de visualización:** Asegurar datos correctos en la pantalla OLED.
- 

*Desarrollador:  
Santiago Bedoya  
Corrales*

**Correo electrónico puedes escribir ante cualquier duda o sugerencia**  
**Correo: microestacionmeteorologica@gmail.com**