

## ***Manual básico de pruebas del NodeMCU ESP8266, versión ESP-12E. Arduino IDE.***

### **Contenido:**

1. Introducción al NodeMCU, ESP-12E
2. Conexión de componentes.
3. Bajar a su computadora el IDE Arduino.
4. Integrar (Add On) al IDE el sistema NodeMCU ESP8266, ESP-12E.
5. Carga del driver CH340G USB-Serial.
6. Carga de 2 programas de aplicación simples.

## 1. Introducción al NodeMCU ESP8266, versión ESP-12E:

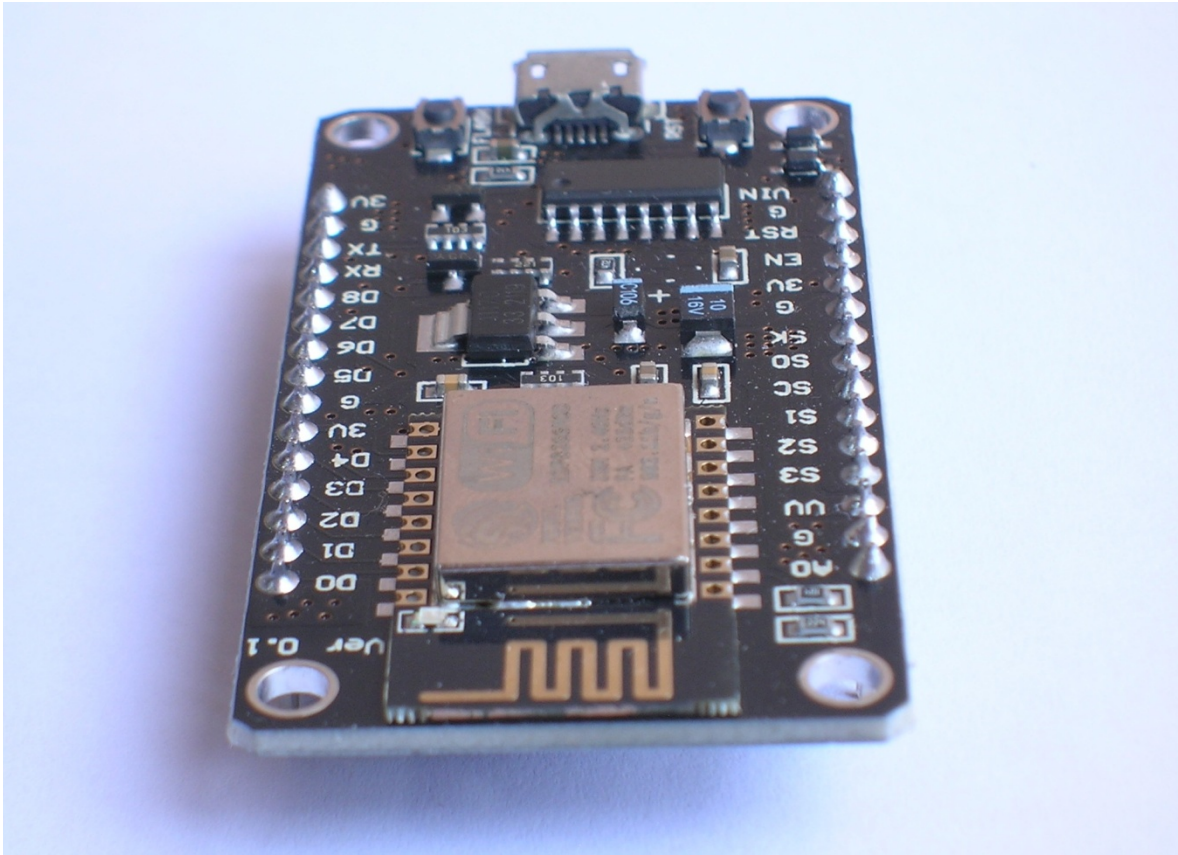


Figura 1

Un dispositivo microcontrolador nuevo ha ingresado al mundo de la electrónica, bajando los costos de sistemas que antes eran complejos en su desarrollo, como el llamado IoT (Internet of Things) y la automatización de casas (Home Automation). Se trata del nuevo NodeMCU ESP8266 en su versión ESP-12E V3.

El NodeMCU ESP-12E, es una plataforma completa de desarrollo, módulo SoC (system on chip), basado en el microcontrolador Tensílica L106 de 32 bits, funcionando a 80 Mhz, con una memoria Flash de 4MB, y 32 KB de SRAM. Tiene conectividad Wifi, estándar 802.11 b/g/n, para la implementación de dispositivos operando como servidores de internet (web server).

NodeMCU incorpora además un circuito CH340G, con lo cual es posible comunicarse durante la fase de desarrollo con el puerto serial del microcontrolador desde el puerto USB de la PC y usando un cable Micro USB.

El NodeMCU integra el protocolo TCP-IP y tiene conectividad Wifi, con antena de radiofrecuencia integrada, lo cual permite su conexión a internet a través de un router cercano al módulo (digamos hasta unos 3 metros), con la posibilidad de control remoto del tipo "Anywhere", desde cualquier parte. Esto significa que, ya sea a través de un teléfono celular ó una computadora conectados a internet, se pueden enviar comandos al NodeMCU para activar dispositivos y recibir información de sensores ó alarmas instaladas en casas y oficinas.

Durante la fase de desarrollo, NodeMCU se conecta directamente a la computadora con un cable micro USB, el cual proporciona la alimentación de 5v al módulo y permite la carga de nuevos programas a una velocidad típica de 115200 bps.

NodeMCU cuenta con 13 pines de entrada/salida, llamados GPIO (General Port Input Output), y puede funcionar de manera totalmente autónoma como servidor web y recibir comandos remotos para operar los pines mencionados, ya sea para activar dispositivos ó enviar status de sensores digitales. Incluye también pines con las funciones PWM, I2C, SPI, UART, y un convertidor A/D de 10 bits.

Puede alimentarse directamente a través del conector micro USB ó con una fuente externa. Su voltaje de alimentación es de 5v, el cual es reducido a 3.3v por medio de un regulador on board. El consumo promedio es de 100 ma. La corriente aumenta durante la transmisión y recepción Wifi, de tal manera que, si se alimenta con fuente externa, se recomienda una capacidad de 300 ma.

Los lenguajes de programación más usados para NodeMCU, son Arduino y Micropython. Ambos son de código abierto y cuentan con herramientas de desarrollo gratuitas.

En este manual, haremos uso del lenguaje Arduino, por ofrecer la mayor disponibilidad y sencillez de uso. Arduino cuenta con un IDE (Integrated Development Environment) completo, un lenguaje de programación sencillo y numerosas referencias a nuevos proyectos.

### Especificaciones técnicas:

- Modelo: ESP8266, versión ESP-12E, V3
- Función: plataforma de conectividad Wifi, TCP-IP, IoT.
- Procesador: Tensílica L106 de 32 bits, bajo consumo.
- Memoria: 4 MB Flash, 32 KB SRAM
- Radiofrecuencia: WiFi, 802.11 b/g/n, 2.4 Ghz, antena integrada.
- Protocolo: TCP/IP integrado, conectividad a internet.
- Comunicación con computadora: a través de interfaz USB-Serial CH340G
- Velocidad de comunicación por USB: 9600 hasta 961,600 bps
- Alimentación: 5v, a través del puerto micro USB ó con fuente externa.
- Corriente máxima consumida: 250 ma.
- 13 pines GPIO disponibles para entrada/salida.
- Pines con funciones de PWM, 1wire, I2C, SPI, UART, ADC.
- Dimensiones: 5.7 x 3.3 x 1.5 cms
- Distancia entre pines: estándar 2.54 mm

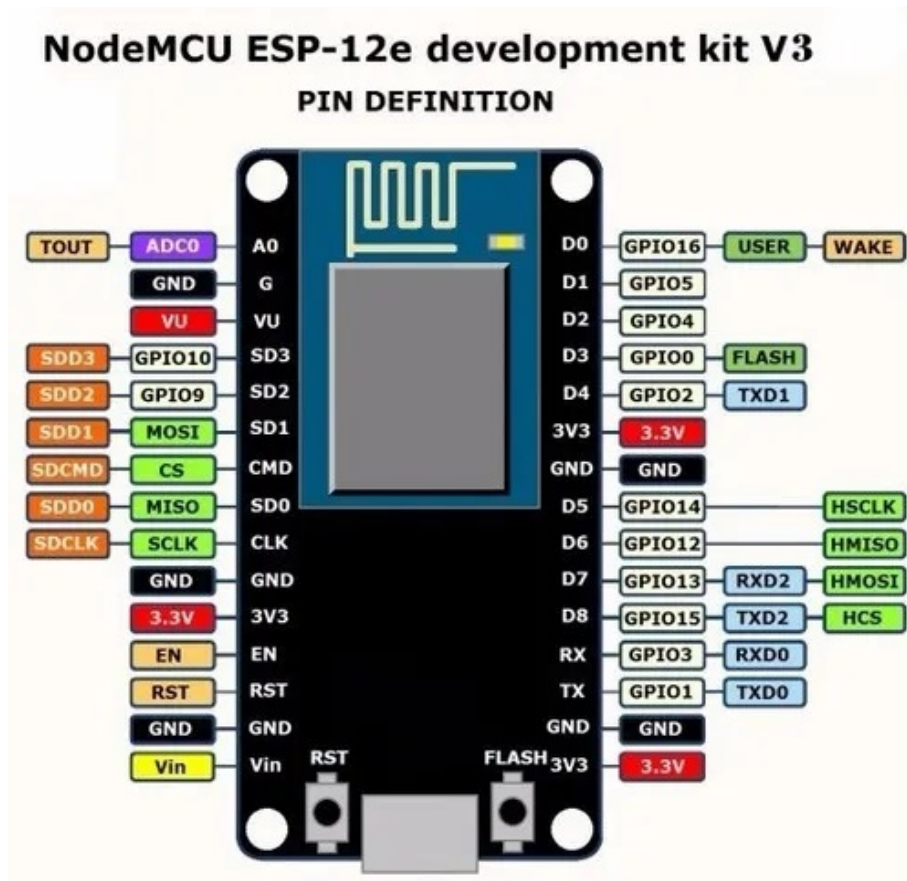


Figura 2



## 2. Conexión de componentes:

Para la realización de la instalación del software, firmware y las primeras pruebas al módulo, usted necesita:

- 1 módulo NodeMCU versión ESP-12E
- 1 cable Micro USB, a través del cual se le da la alimentación de 5v al NodeMCU y al mismo tiempo puede tener comunicación con la PC
- En la foto de abajo se incluye una base, la cual no es indispensable para las pruebas de este manual.

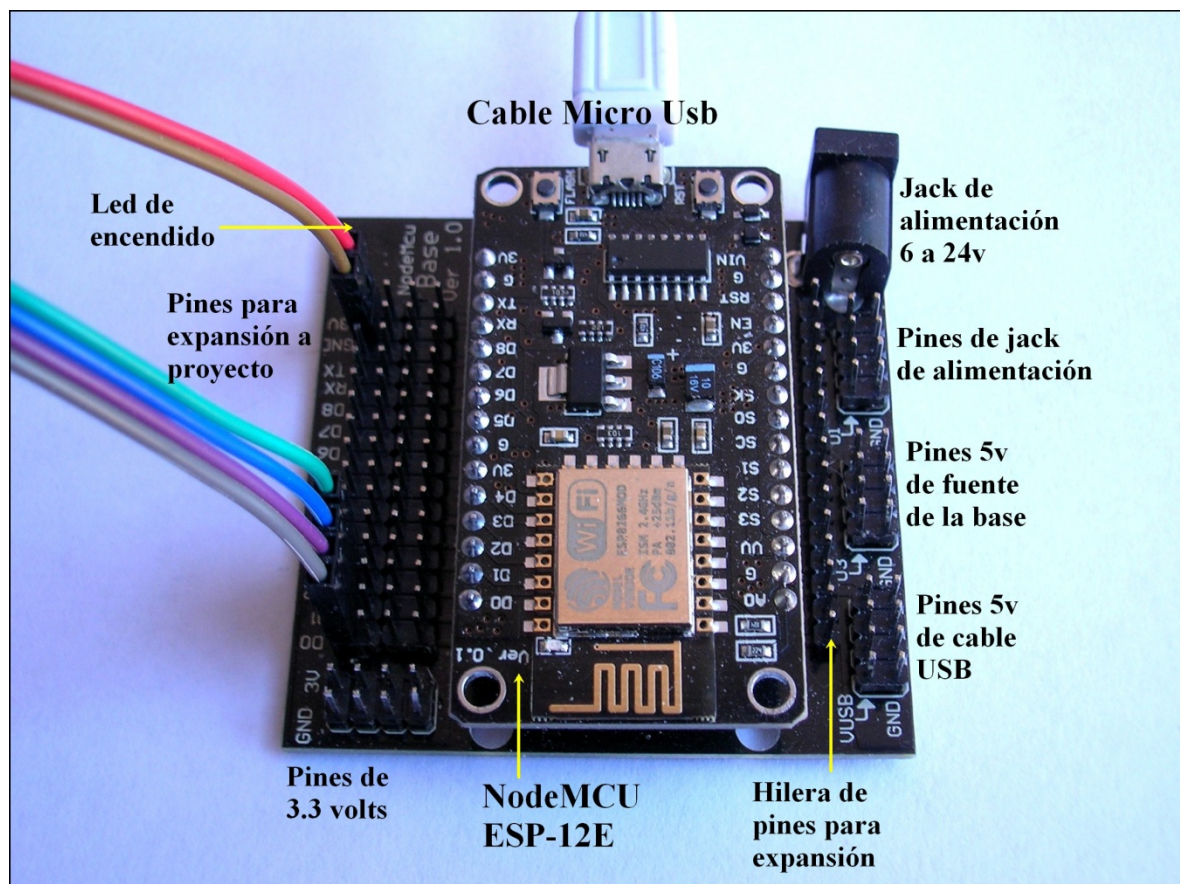


Figura 3

Inserte el cable Micro USB al NodeMCU, con lo cual alimentará el circuito. Deben encenderse el led rojo (si cuenta con la base) y brevemente el led azul del módulo (junto a la antena de WiFi). Por el momento no utilizaremos los cables de expansión mostrados en la foto.

### 3. Bajar a su computadora el IDE Arduino:

Es importante que usted cuente con la versión más reciente del IDE, ya que éste se está constantemente actualizando con nuevas librerías y placas de desarrollo. En este caso nuestra placa es la NodeMCU ESP8266, versión ESP-12E. El link es el siguiente:

[arduino.cc/en/Main/Software](https://arduino.cc/en/Main/Software)

Almacene en una carpeta el programa de instalación:

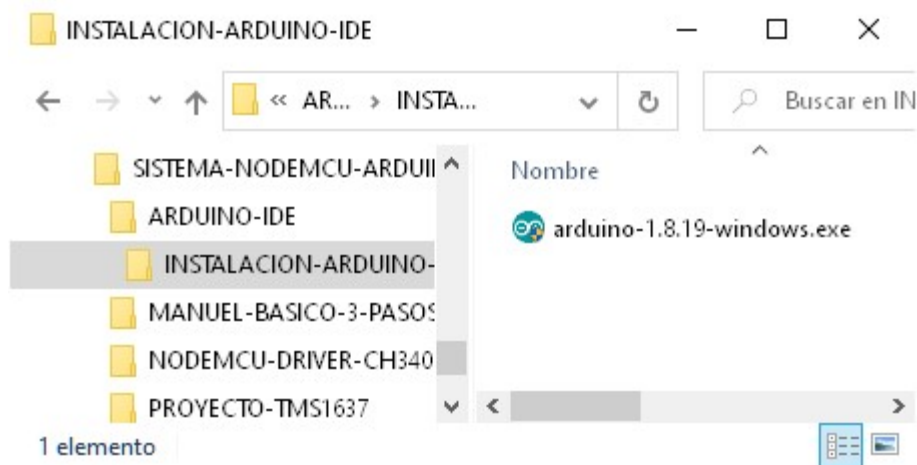


Figura 4

Ejecute el archivo de instalación. Al finalizar, usted tendrá la carpeta de instalación completa (mostrada en la imagen de abajo) en el path (figura 5):

C:\Program Files (x86)\Arduino

La carpeta default para almacenar sus nuevos proyectos estarán en el path:

C:\Users\JBM\Documents\Arduino

En donde “JBM” corresponde en este ejemplo al nombre del usuario.

Adicionalmente, habrá un ícono del IDE Arduino en el escritorio de su PC.

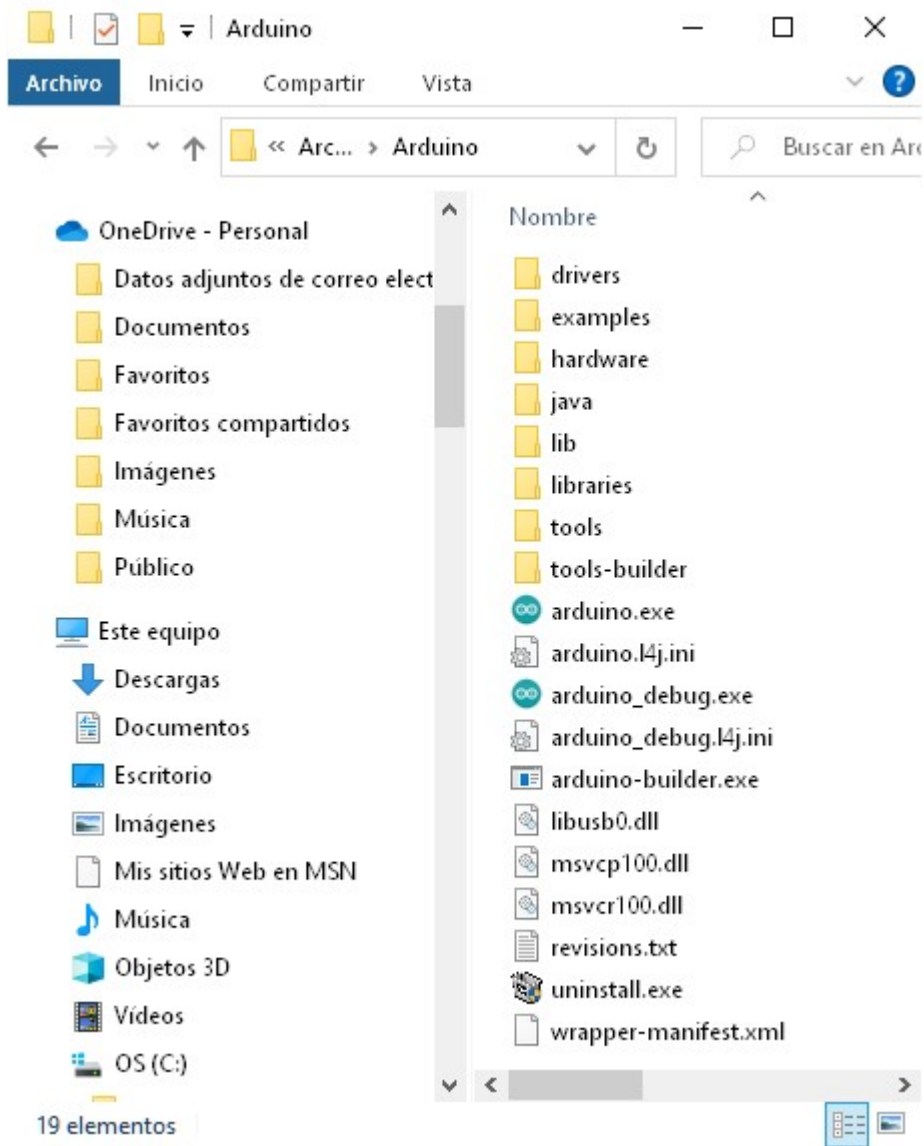


Figura 5

#### 4. Integrar (Add On) al IDE el sistema NodeMCU ESP8266, ESP-12E.

El IDE de Arduino permite integrar numerosas placas nuevas de desarrollo, en este caso es necesario añadir la que usaremos en este manual.

Abra el IDE y elija la opción “Archivo” y “Preferencias” (figura 6)

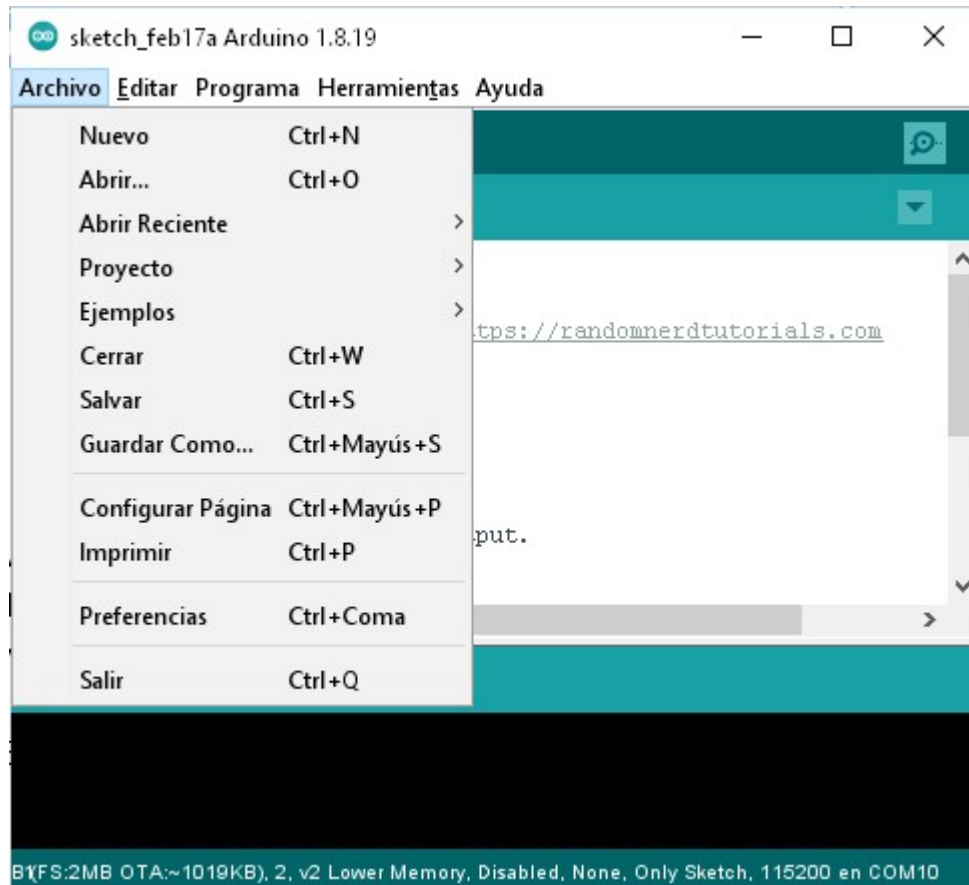


Figura 6

En la casilla “Gestor de URL adicionales de tarjetas”, copie y pegue la siguiente dirección URL, como se muestra en la imagen abajo (figura 7).



**[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)**

Finalice dando click en “OK”

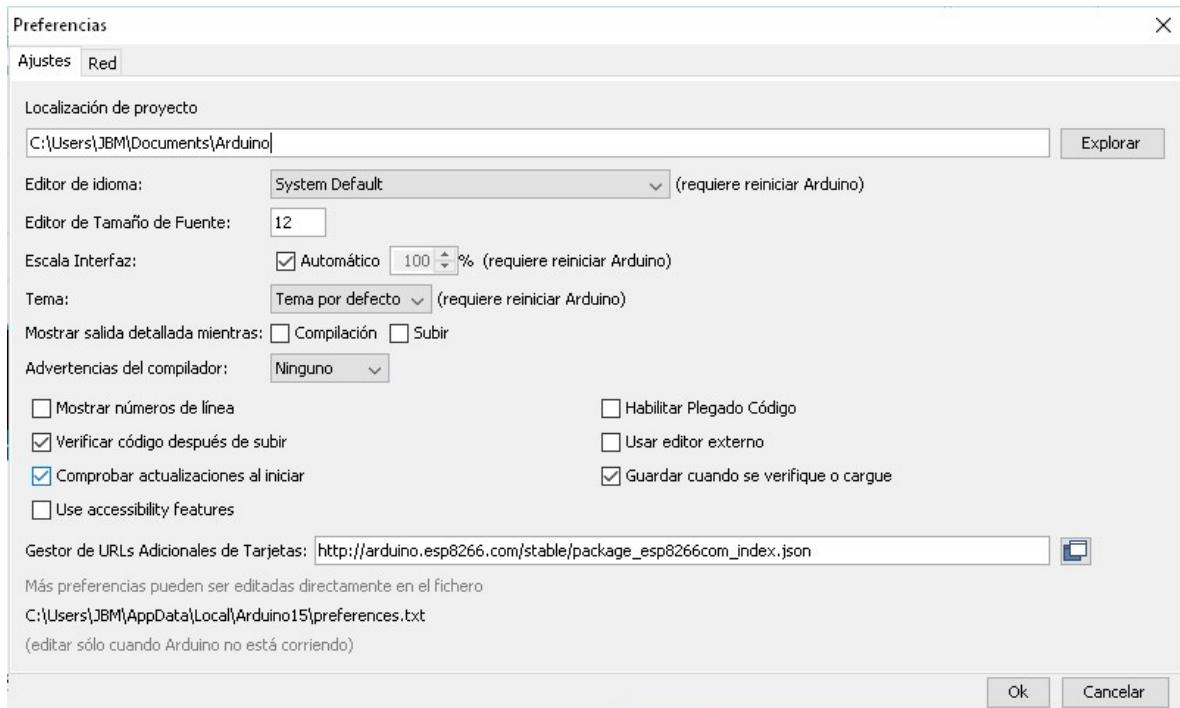


Figura 7

Elija ahora la opción “Herramientas”, ”Placa” (figura 8) y aparecerá una lista. Hasta arriba en la lista, elija “Gestor de Tarjetas” y aparecerá otra lista con nuevas placas. Elija ESP8266, como se muestra en la figura 9 y dé click en “Instalar”. Una vez que está instalado verifique que se muestre el mensaje “INSTALLED” (figura 10).

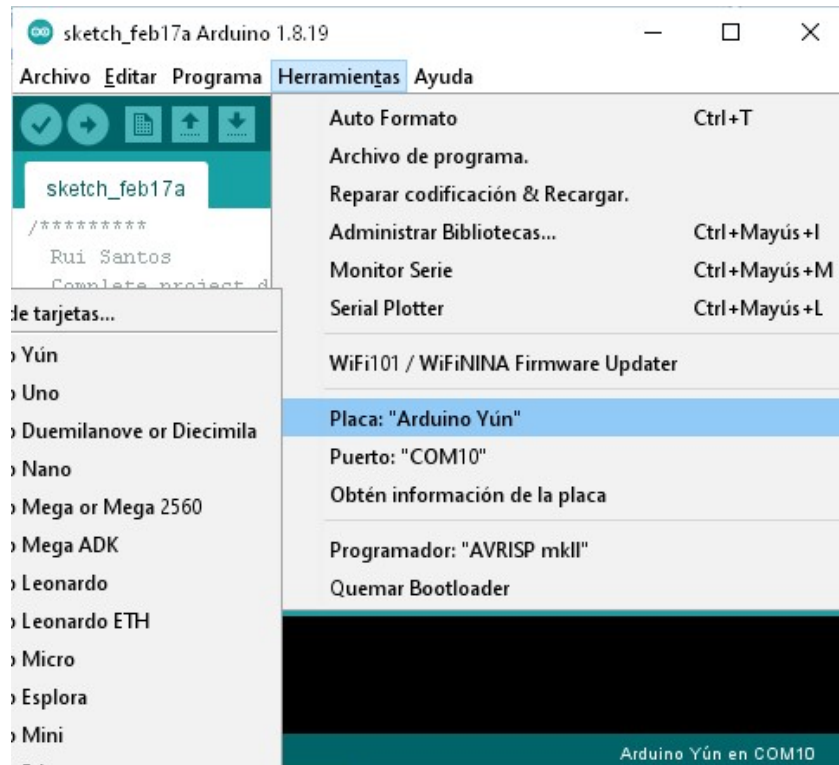


Figura 8



Figura 9

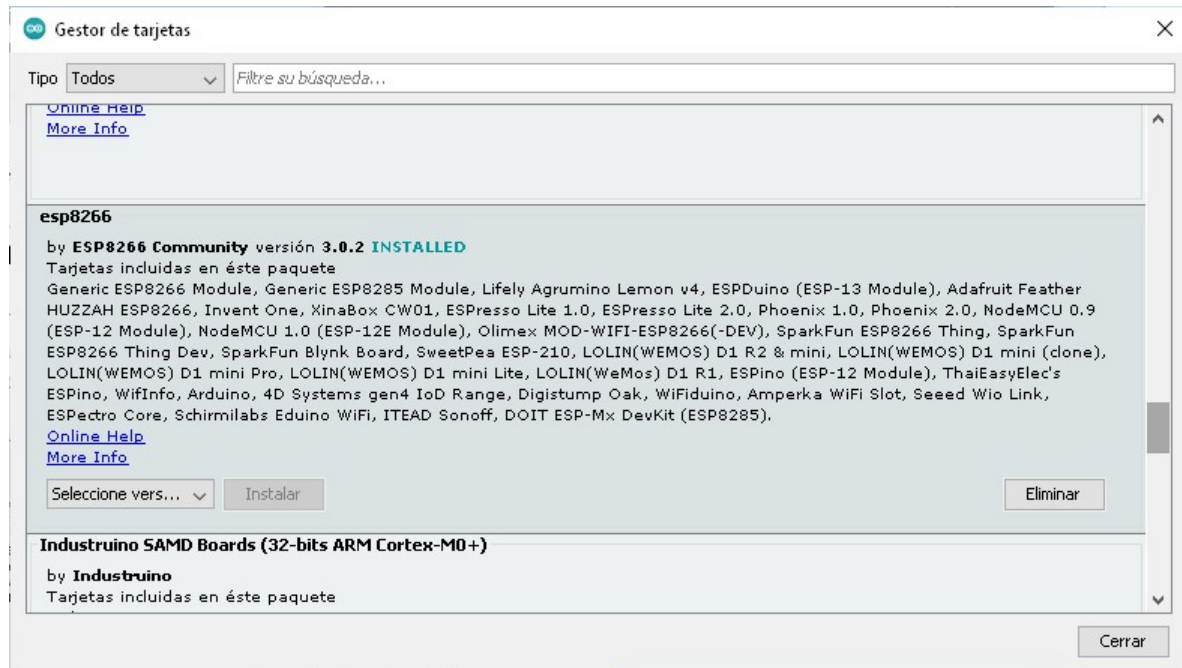


Figura 10

Una vez que el ESP8266 ha sido agregado al IDE, es necesario elegir el modelo del módulo, en este caso es NodeMCU 1.0, ESP8266, ESP-12E (ver la figura 11, que se muestra en su versión con letreros en inglés).

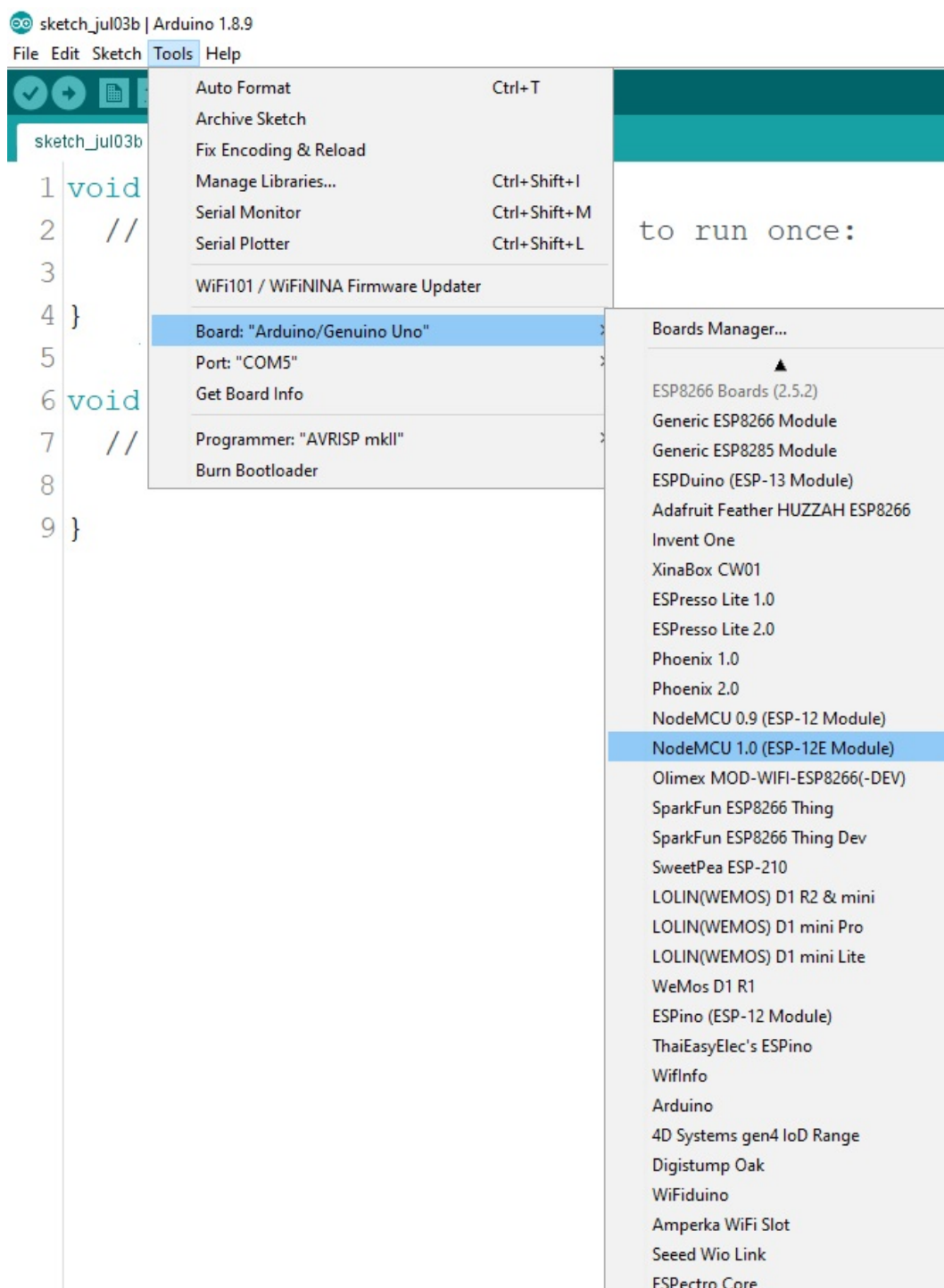


Figura 11

Hasta aquí termina la instalación en el IDE. Pase al siguiente capítulo para instalar el driver del circuito USB-Serial.



##### 5. Dar de alta el driver USB-Serial CH340G.

Inserte el cable Micro USB a su computadora (en cualquier puerto USB) y al conector del módulo ESP8266 (figura 12)

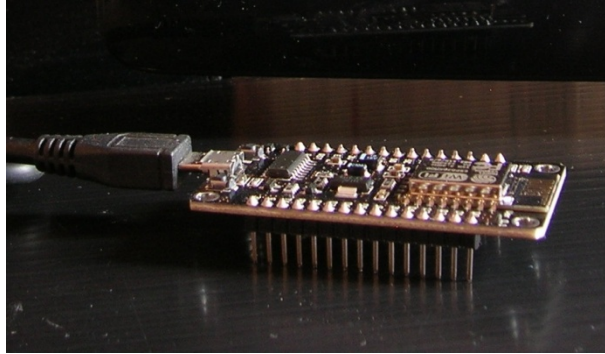


Figura 12

El módulo tiene integrado un circuito convertidor de USB a Serial. Es necesario cargar el driver CH340G en la computadora PC de tal manera que dicho circuito sea reconocido como un puerto serial COM por el administrador de dispositivos. Baje el driver aquí:

<https://puntoflotante.net/CH341SER.ZIP>

Extraiga los archivos del formato .zip y abra la carpeta del driver. Ejecute el archivo “SETUP.EXE” para instalar el driver.

Ahora abra el administrador de dispositivos y deberá mostrar la ventana con al puerto serial COM asignado. En este ejemplo se trata del COM10. En su computadora podría asignarse otro número del puerto (figura 13).

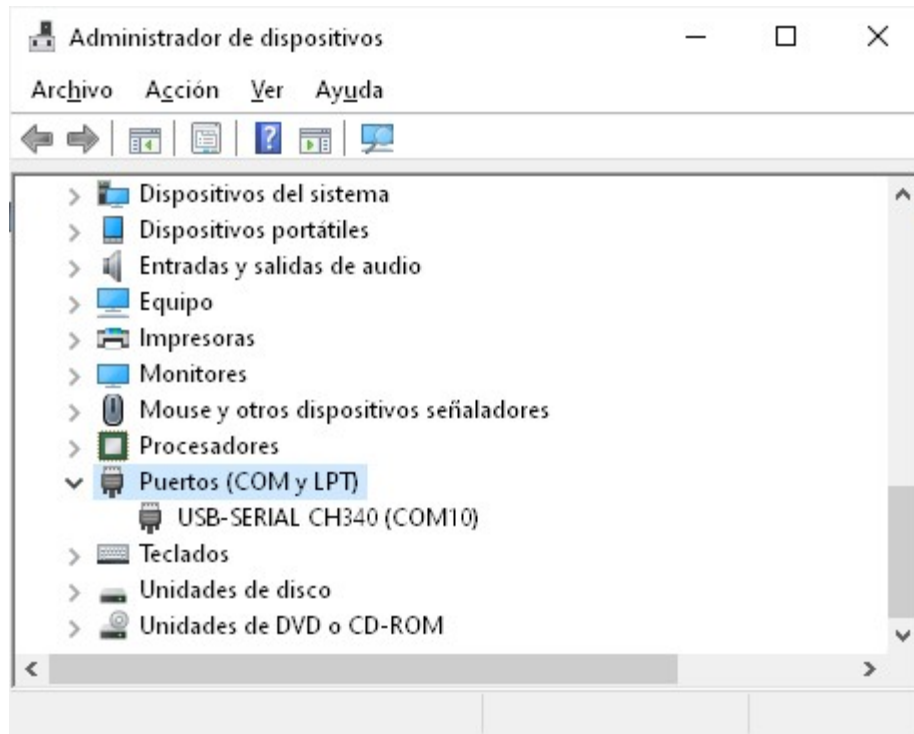


Figura 13

## 6. Carga de 2 programas de aplicación simples.

Finalmente, como prueba final de la instalación, realizaremos la carga de dos sencillos programas, utilizando el led azul y el botón llamado “FLASH”. Ambos dispositivos ya están integrados a los circuitos del módulo ESP8266, de tal manera que no es necesario realizar ningún alambrado.

Baje los 2 programas de prueba de este link y guarde el contenido extraído en una carpeta:

<https://puntoflotante.net/2-PROGRAMAS-DE-PRUEBA.zip>

PROGRAMA 1: Con el IDE de Arduino abierto, elija “Archivo”, “Abrir” y, desde la carpeta de los 2 proyectos abra: “BLINK-LED.ino” (figura 14).

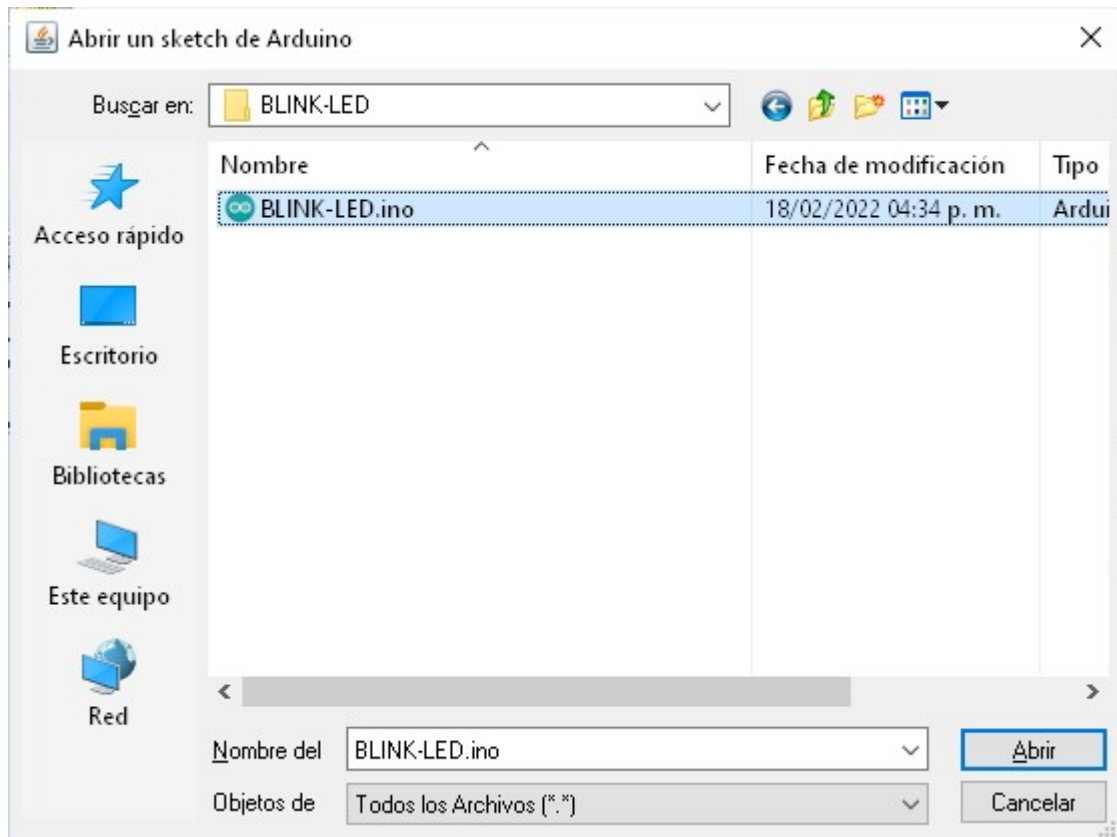
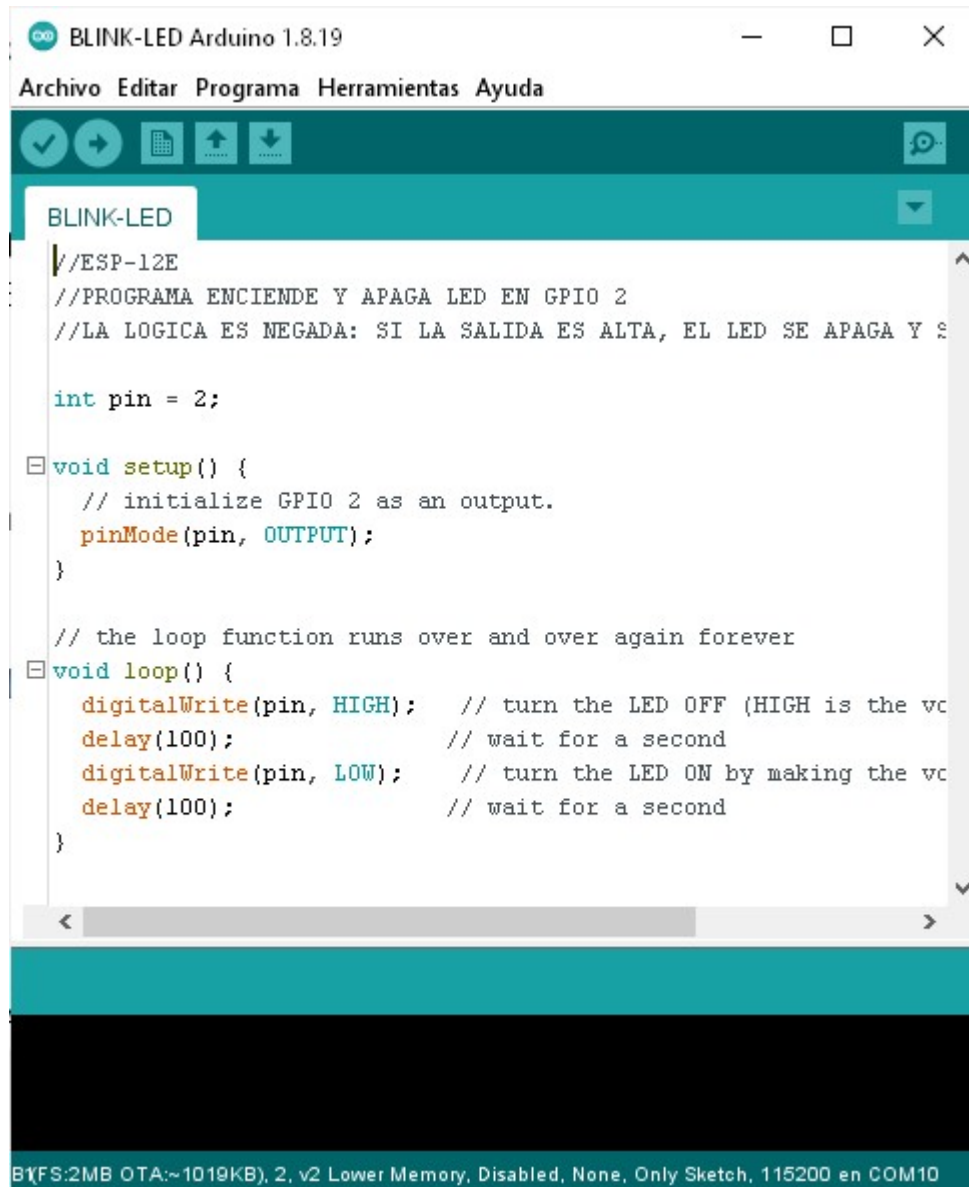


Figura 14

Se abrirá otra ventana en el IDE, como la mostrada en la figura 15, en donde puede observarse el archivo fuente.

Para cargar el programa ejecutable en el módulo, dé click en “Programa” y “Subir”. El IDE compila primeramente el archivo fuente y posteriormente envía el ejecutable al módulo, en donde corre automáticamente. Al final del proceso, usted observará el led azul parpadeando cada 100 ms.



```

BLINK-LED

//ESP-12E
//PROGRAMA ENCIENDE Y APAGA LED EN GPIO 2
//LA LOGICA ES NEGADA: SI LA SALIDA ES ALTA, EL LED SE APAGA Y SI LA SALIDA ES BAJA, EL LED SE ENCIENDE

int pin = 2;

void setup() {
  // initialize GPIO 2 as an output.
  pinMode(pin, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(pin, HIGH); // turn the LED OFF (HIGH is the voltage divider)
  delay(100); // wait for a second
  digitalWrite(pin, LOW); // turn the LED ON by making the voltage divider LOW
  delay(100); // wait for a second
}

```

B(FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 en COM10

Figura 15



PROGRAMA 2: Con el IDE de Arduino abierto, elija “Archivo”, “Abrir” y, desde la carpeta de los 2 proyectos abra: “BUTTON.ino” (figura 16).

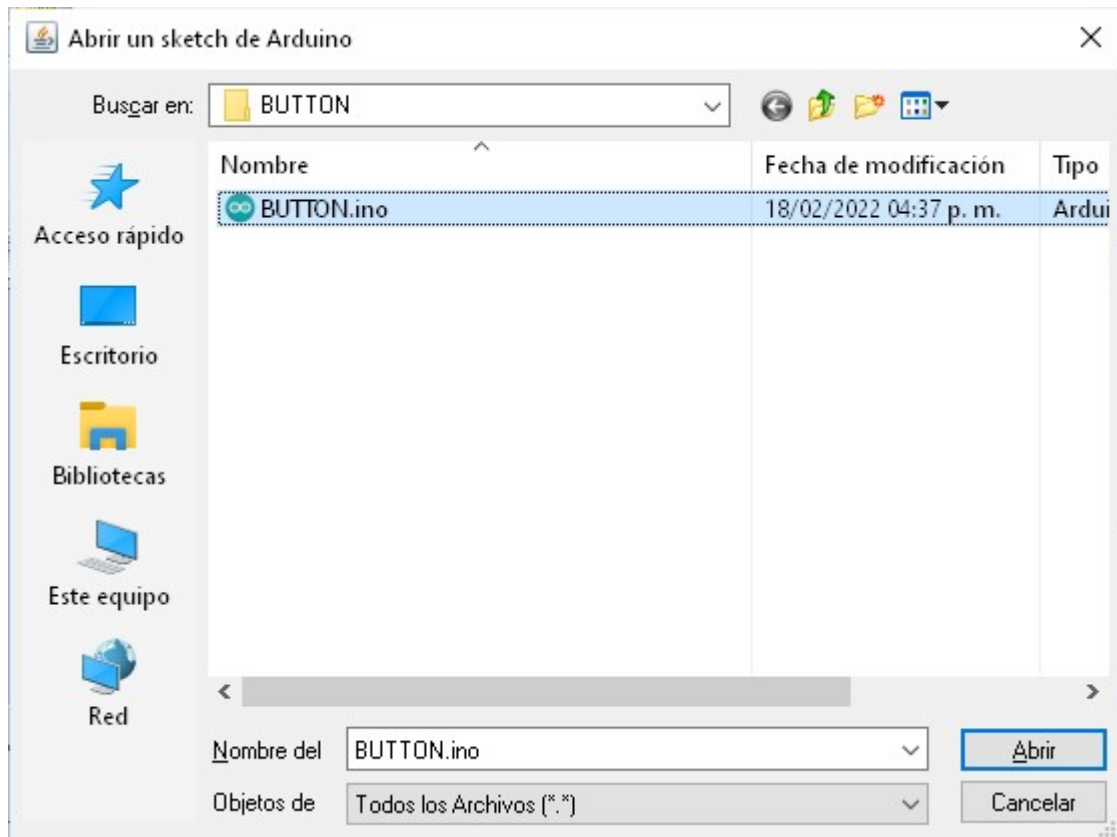


Figura 16

Se abrirá otra ventana en el IDE, como la mostrada en la figura 17, en donde puede observarse el archivo fuente.

Para cargar el programa ejecutable en el módulo ESP-12E, dé click en “Programa” y “Subir”. El IDE compila primeramente el archivo fuente y posteriormente envía el ejecutable al módulo, en donde corre automáticamente. Al final del proceso, al oprimir el botón “FLASH”, debe encenderse el led azul y al liberar el botón, debe apagarse el led.

BUTTON Arduino 1.8.19

Archivo Editar Programa Herramientas Ayuda

BUTTON

```

1 //ESP-12E
2 //LEE PUSH BUTTON "FLASH" EN GPIO 0.
3 //SI SE OPRIME EL BOTON, ENCIENDE LED EN GPIO 2
4 //SI LIBERA EL BOTON, EL LED SE APAGA
5 //LOGICA NEGADA EN EL LED
6
7 // constants won't change. They're used here to set pin numbers:
8 const int buttonPin = 0;    // the number of the pushbutton pin
9 const int ledPin = 2;      // the number of the LED pin
10
11 // variables will change:
12 int buttonState = 0;        // variable for reading the pushbutton status
13
14 void setup() {
15   // initialize the LED pin as an output:
16   pinMode(ledPin, OUTPUT);
17   // initialize the pushbutton pin as an input:
18   pinMode(buttonPin, INPUT);
19 }
20

```

Subido

leaving...

Hard resetting via RTS pin...

82KB IRAM (balanced), Use pgm\_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 en COM10

Figura 17