

## **Desarrollos de dispositivos IoT basado en el módulo ESP32**

Curso Noviembre 2021

Instructores:

M.I.M. Gabriel Jesús Pool Balam  
M.I.M. Julio Alejandro Lugo Jiménez

## Contenido

<b>Introducción.....</b>	<b>4</b>
<b>1. Instalación del IDE de Arduino .....</b>	<b>5</b>
1.1    Instalación del IDE del Arduino v1.8.16 .....	6
1.2    Instalación del plugin ESP32 para IDE del Arduino v1.8.16.....	14
1.3    Ejecutando el ejemplo “blink”.....	19
1.4    Usando la ayuda del IDE del Arduino .....	30
1.5    Usando el monitor del puerto serie del IDE del Arduino .....	35
1.6    Agregando bibliotecas al IDE del Arduino del catálogo de drivers.....	40
<b>2. Implementando códigos en IDE de Arduino.....</b>	<b>50</b>
2.1    El Hardware del módulo NodeMCU-32S .....	51
2.2    Análisis de ejemplos considerados relevantes para la implementación de un lector de temperatura.....	56
2.2.1    Blink.ino.....	57
2.2.2    BlinkWithOutDelay.ino .....	63
2.2.3    Máquina de estados.....	66
2.2.4    Usando la pantalla OLED SSD1306.....	70
2.2.5    Usando el sensor de temperatura DS18B20 de fabricado por dallas semiconductor.....	75
2.2.6    Usando el sensor de temperatura DHT22 (AM2301) fabricado por AMLOGIC	80
2.2.7    Mostrando los valores del sensor en la pantalla del SSD1306 .....	85
2.2.8    Usando el módulo Analog to Digital Converter (ADC) .....	89
<b>3. Diseño de WebServer para NodeMCU-32S.....</b>	<b>94</b>
3.1    El Servidor Web implementado en NodeMCU-32S .....	95
3.1.1    MDNS .....	95
3.1.2    Advanced Web Server .....	102
3.1.3    Paso de argumentos al servidor WEB .....	116
3.2    El Servidor Web con página dinámica en NodeMCU-32S.....	122
3.2.1    ESP32 Filesystem Uploader Plugin.....	122

3.2.2 SPIFFS WebServer.....	126
3.3 Servidor web implementado en Google Sheets .....	137
3.3.1 Simple Google Apps Script .....	137
3.3.2 Simple página web dinámica en Google Apps Script.....	158
3.3.3 Pagina web con gráfica y descarga de históricos en Google Apps Script .....	187
3.3.4 El Cliente para Google Apps Script implementado en NodeMCU-32S .....	217



Centro de Investigación Científica de Yucatán A. C.

Departamento de Instrumentación

[www.cicy.mx](http://www.cicy.mx)

## Introducción

Este manual se enfoca a reunir la información dispersa de internet en un solo lugar con el fin de poder explotar todo el potencial del módulo ESP32 en sus diferentes versiones. También se anexan funciones adicionales que se consideran importantes para el desarrollo de una solución. Para seguir los desarrollos de este manual, se requiere de una conexión a Internet, conocimientos básicos de programación y dominar el uso de una PC.

# Modulo I

## 1. Instalación del IDE de Arduino

**Objetivo General:** Descargará desde la página web, instalará y configurará el software del IDE del Arduino en su computadora, para desarrollar con el módulo ESP32.

## 1.1 Instalación del IDE del Arduino v1.8.16

**Objetivo específico:** Instalará desde el sitio web el IDE del Arduino en su computadora con las opciones del IDE que trae configuradas de manera predeterminada.

Para esta instalación se contempla que usted tiene instalado **Windows 10** en su computadora y con los **parches de Windows** actualizados al día.

Con el explorador de internet de su preferencia, escriba el siguiente link en la barra de búsqueda del navegador: <https://www.arduino.cc/en/Main/Software> presione la tecla **<enter>** y espere a que el navegador cargue la página web. Una vez cargada la página web, ubique la siguiente sección en la página web:

## Downloads



The screenshot shows the Arduino IDE 1.8.16 download page. On the left, there's a logo and the text "Arduino IDE 1.8.16". Below that is a description of what the IDE does and a link to the "Getting Started" page. On the right, there's a "DOWNLOAD OPTIONS" section with a red arrow pointing to the "Windows" section. The "Windows" section has two options: "Win 7 and newer" (link) and "ZIP file" (link). Below that is a "Windows app" section with a "Get" button. Further down are sections for "Linux" (32 and 64-bit), "Mac OS X", and "Release Notes Checksums (sha512)".

...déle un *click* con el botón izquierdo del mouse en el texto marcado por la flecha que se muestra en la figura anterior y espere a que la siguiente página web se cargue, a continuación, se muestra lo siguiente:

## Support the Arduino IDE

Since the release 1.x release in March 2015, the Arduino IDE has been downloaded **55,400,113** times — impressive! Help its development with a donation.

\$3

\$5

\$10

\$25

\$50

Other

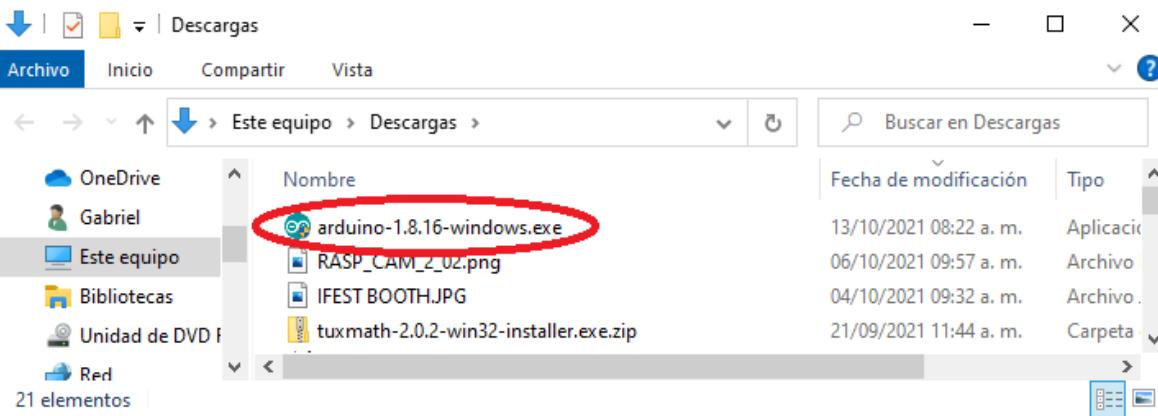
**JUST DOWNLOAD**

**CONTRIBUTE & DOWNLOAD**

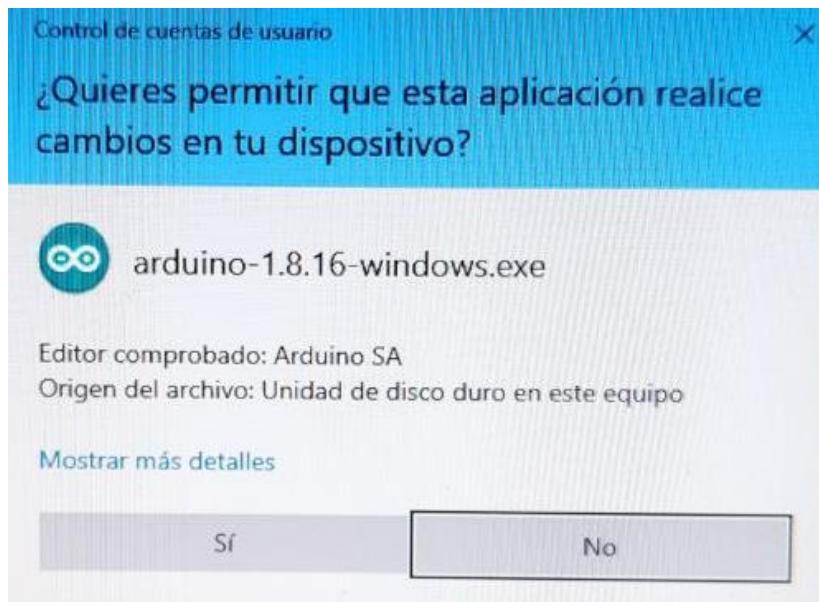
...dé un *click* con el botón izquierdo del mouse en el texto marcado como “**just download**” e inmediatamente saldrá una ventana emergente como ésta:



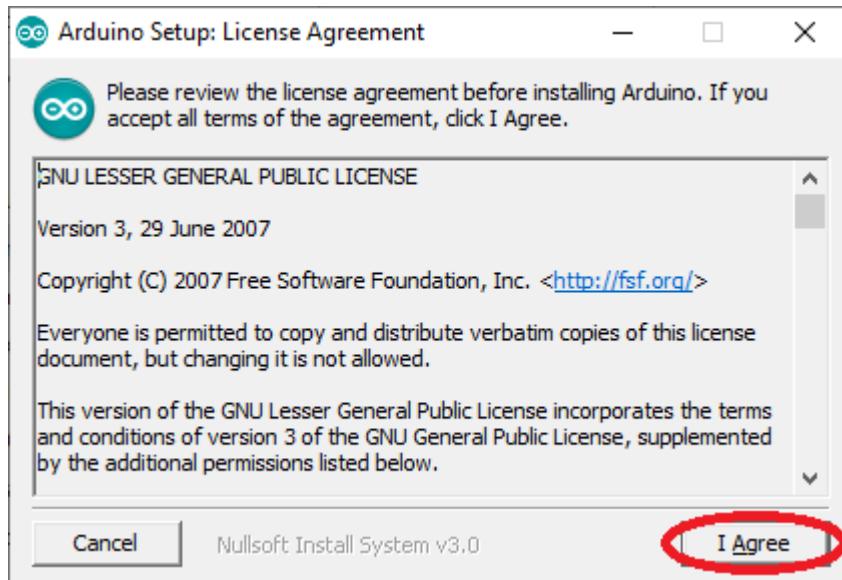
... presione el botón con el texto “**Guardar archivo**” y la descarga iniciará, cuando haya concluido, su navegador le avisará que la descarga ya ha terminado. Ubique el archivo descargado en la carpeta de descargas (típicamente los archivos se guardan en esta carpeta a menos que usted haya modificado la ruta de descarga). Se mostrará como sigue:



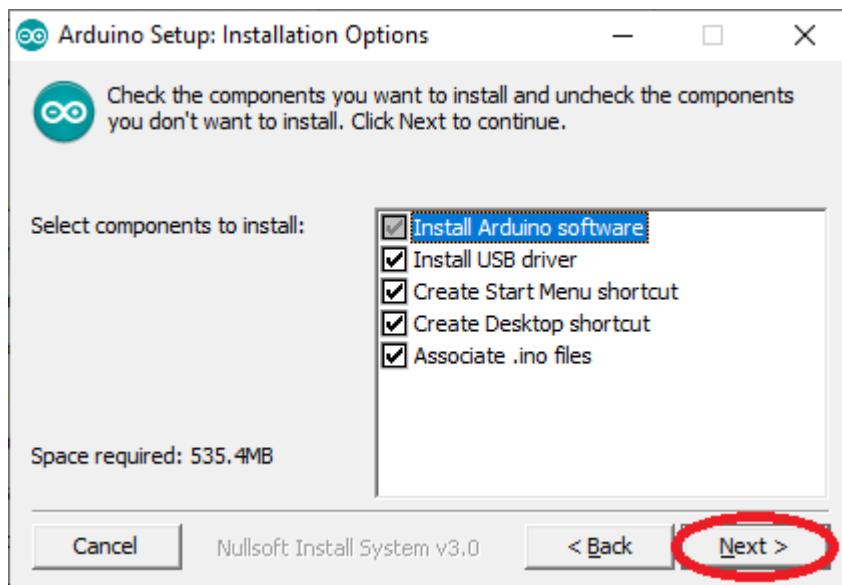
... presione doble *click* con el botón izquierdo del mouse al archivo recién descargado (llamado **Arduino-1.8.16-windows.exe**) e inmediatamente la pantalla se pondrá negra con una ventana como se muestra a continuación:



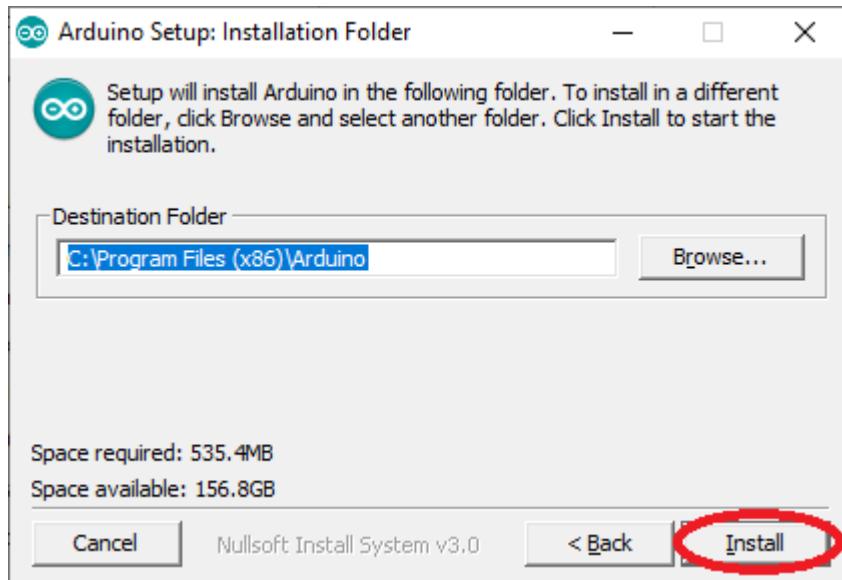
... presione el botón con el texto “**sí**” y la instalación se ejecutará de manera automática, se mostrará una ventana como se muestra a continuación:



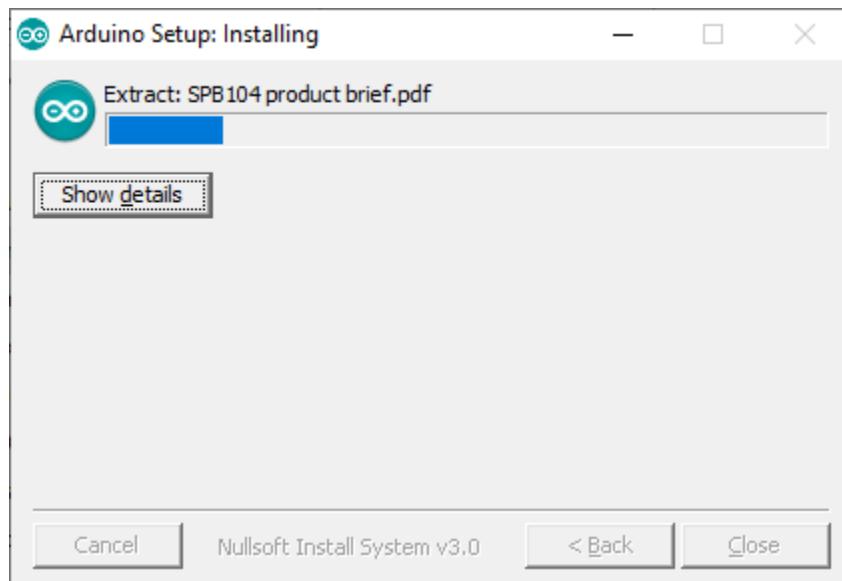
Presione el botón marcado con el texto “**I Agree**” y se mostrará lo siguiente:



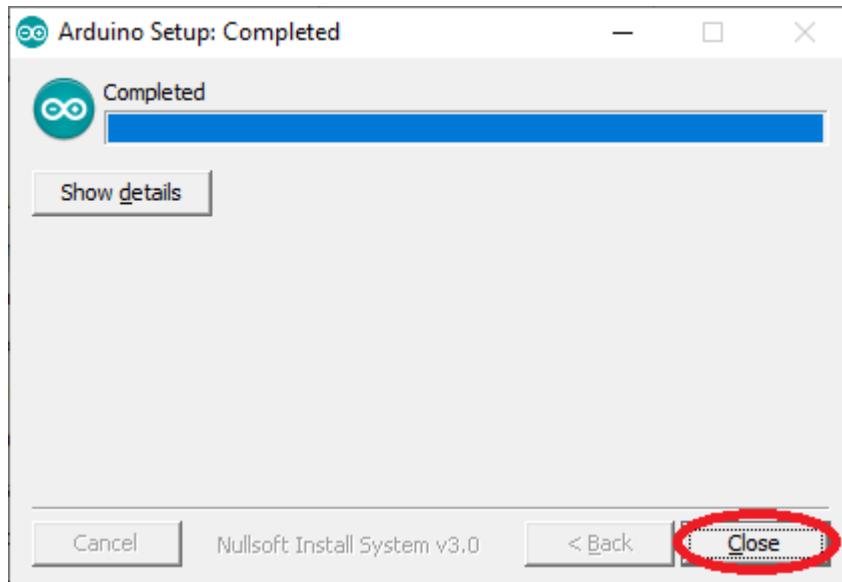
Presione el botón marcado con el texto “**Next**” y se mostrará lo siguiente:



Presione el botón marcado con el texto “**Install**” y se mostrará lo siguiente:



Espere a que la barra de progreso (de color azul) se llene y se mostrará lo siguiente:



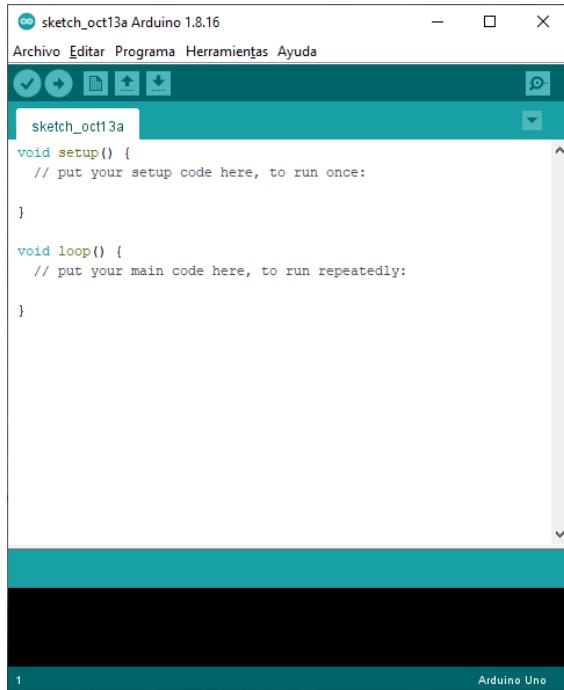
Presione el botón con el texto “**Close**” y la ventana anterior se cerrará. Para abrir el programa recién instalado basta con hacer doble *click* con el botón izquierdo del mouse al ícono ubicado en su escritorio:



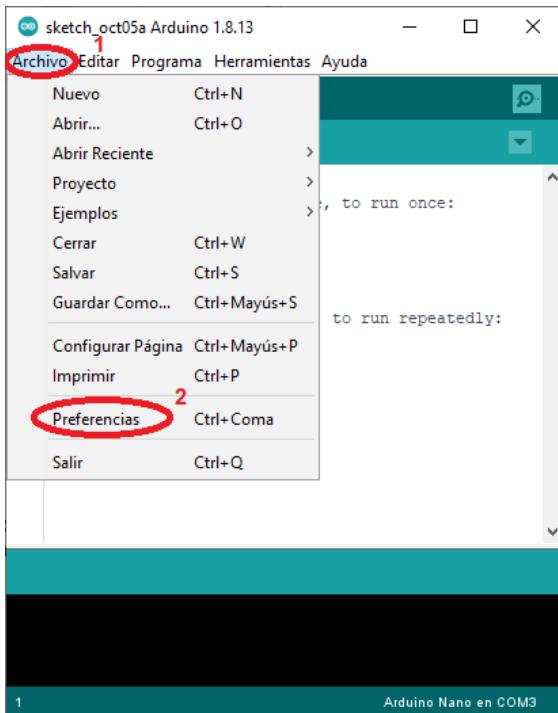
Al hacer doble *click* con el botón izquierdo del mouse en el ícono anterior, se abrirá como se muestra a continuación:



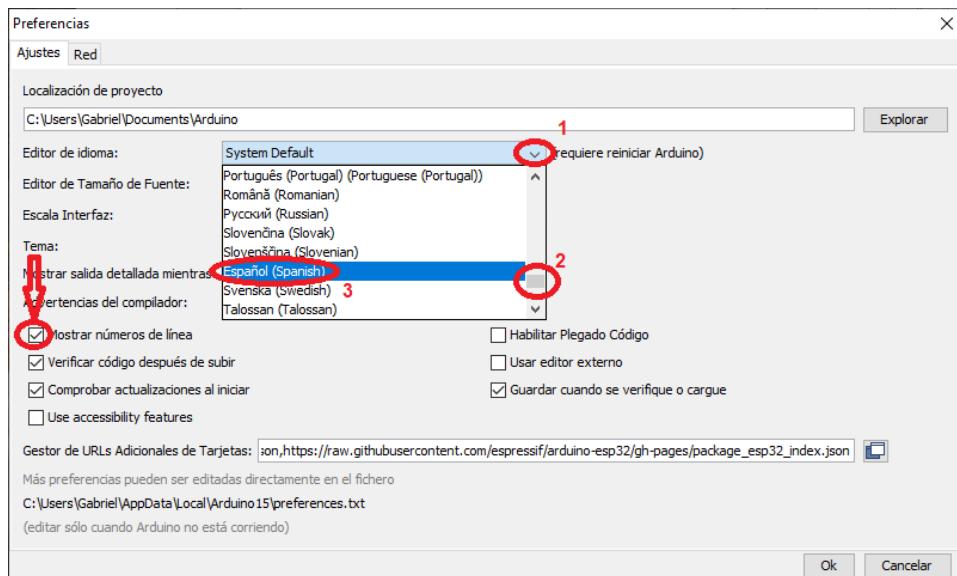
Seguidamente, se mostrará la ventana del programa Arduino:



Si por alguna razón, se muestra el menú del programa en otro idioma diferente al suyo, puede modificarlo como sigue, presione el botón izquierdo del mouse en la parte superior marcada como “**Archivo**” y seguidamente seleccione “**Preferencias**”:



Se abrirá una ventana como se muestra a continuación:



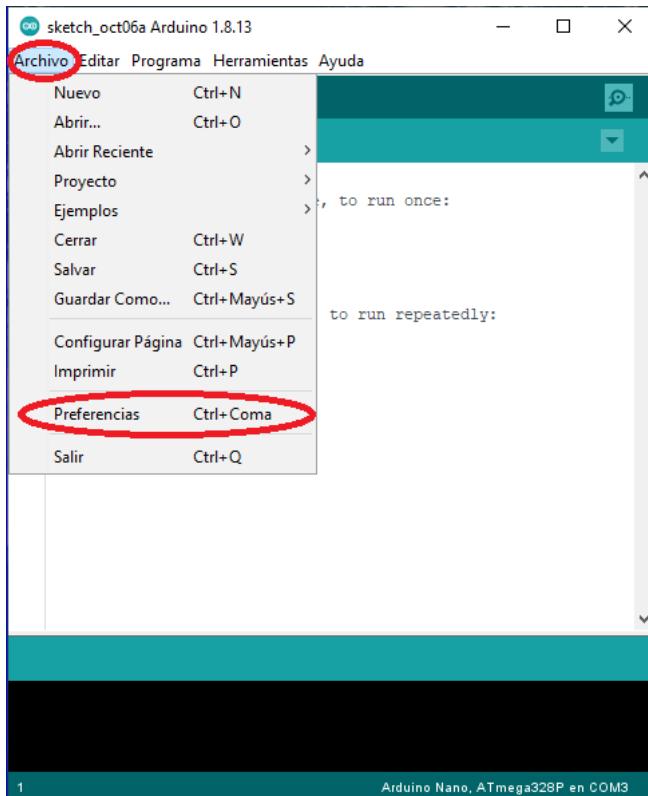
Ubique la flecha en el campo llamado “**Editor de Idioma**” y presione el botón izquierdo del mouse y se abrirá un menú, deslice el menú hacia abajo hasta encontrar el idioma de su preferencia, posteriormente, presione el botón izquierdo del mouse en el nombre del idioma que deseé elegir. Seguidamente, **active el casillero con la etiqueta “Mostrar números de línea”**. Por último, presione el

botón con el texto “Ok”. Para que los cambios surtan efecto, cierre el programa y vuélvalo a abrir.

## 1.2 Instalación del plugin ESP32 para IDE del Arduino v1.8.16

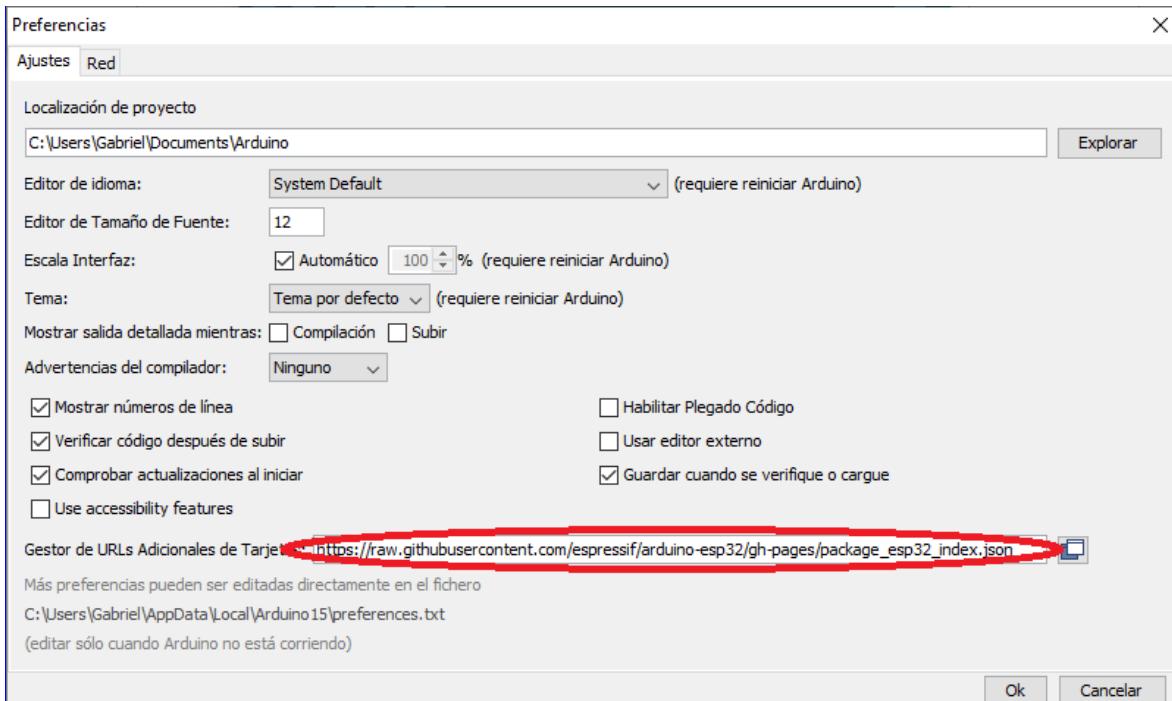
**Objetivo específico:** Configurar el IDE del Arduino para trabajar con diversos modelos de módulos basados en el chip ESP32

Una vez instalado el IDE del Arduino se configurará para habilitar las opciones de trabajo con los modelos de módulos EPS32. En el link: <https://github.com/espressif/arduino-esp32> se encuentran las instrucciones necesarias para cubrir este paso. Según lo anterior, abrir el IDE del Arduino, ubique la ventana de preferencias como sigue:

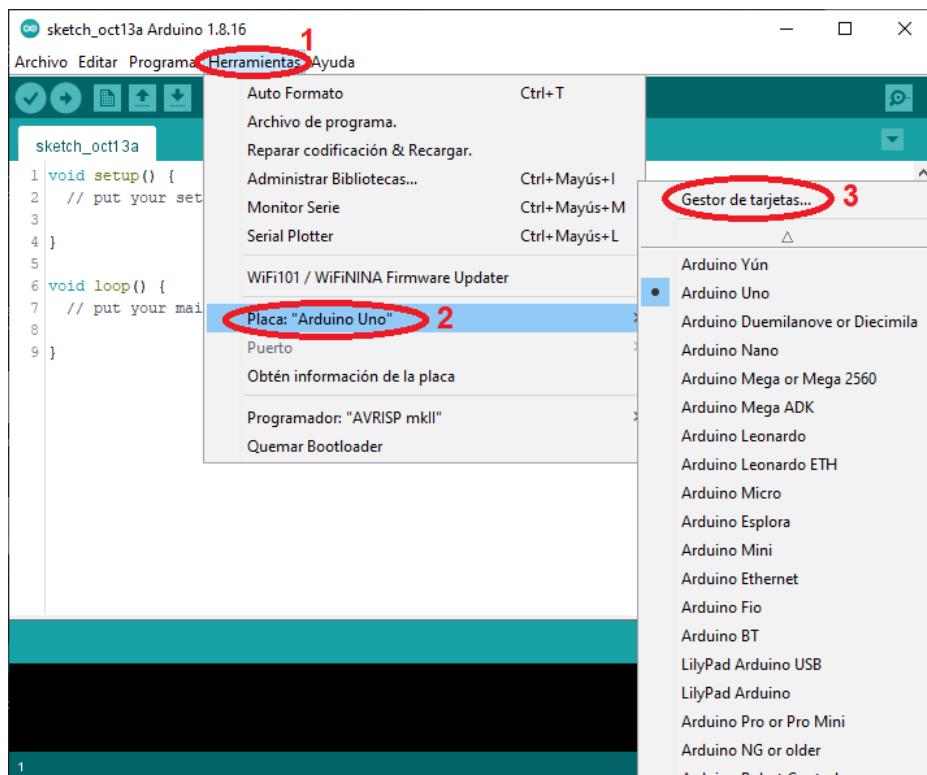


A continuación introduzca el siguiente link  
[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)

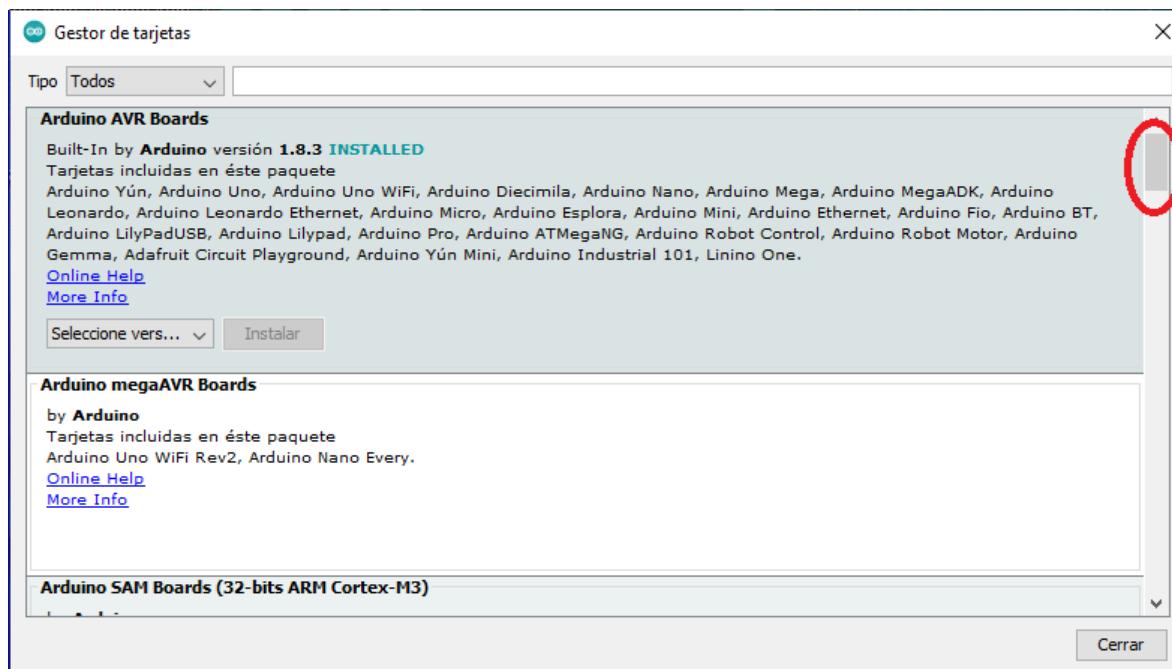
en el cuadro de texto llamado “Gestor de URLs adicionales de tarjetas” como se muestra a continuación:



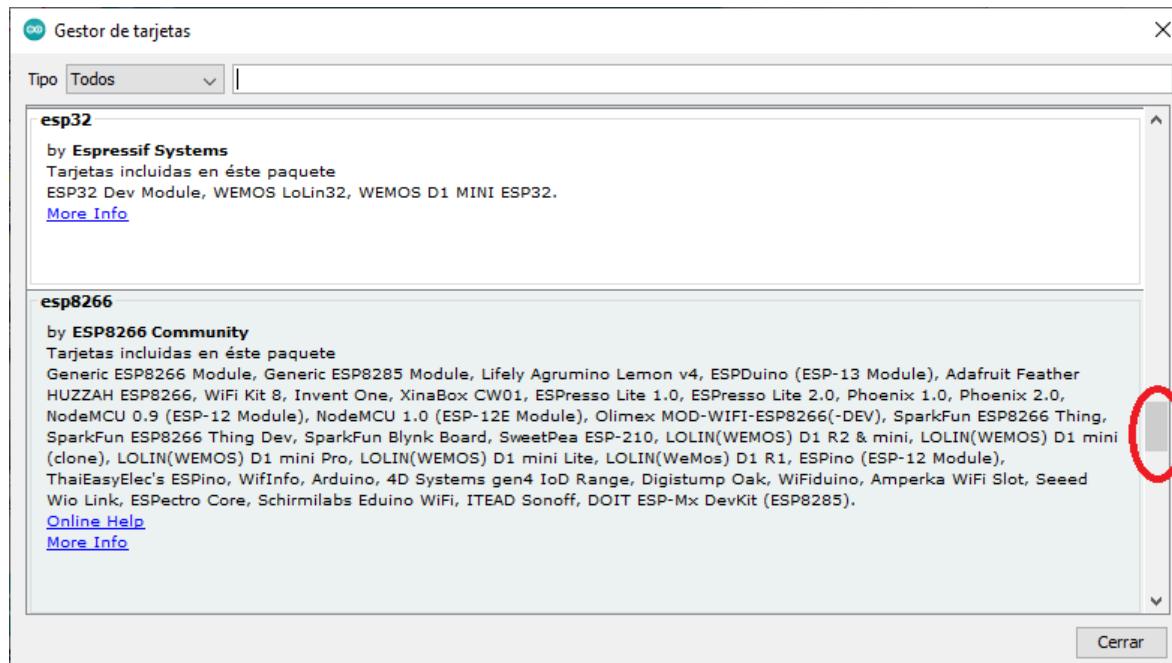
Si tenía alguna otra tarjeta instalada anteriormente y en el campo del “**Gestor de URLs Adicionales de tarjetas**” ya existe un *link*, sólo agregue una coma (,) y pegue el link anterior. Posteriormente, presione el botón “Ok” y la ventana se cerrará. Abra la ventana del gestor de tarjetas como sigue:



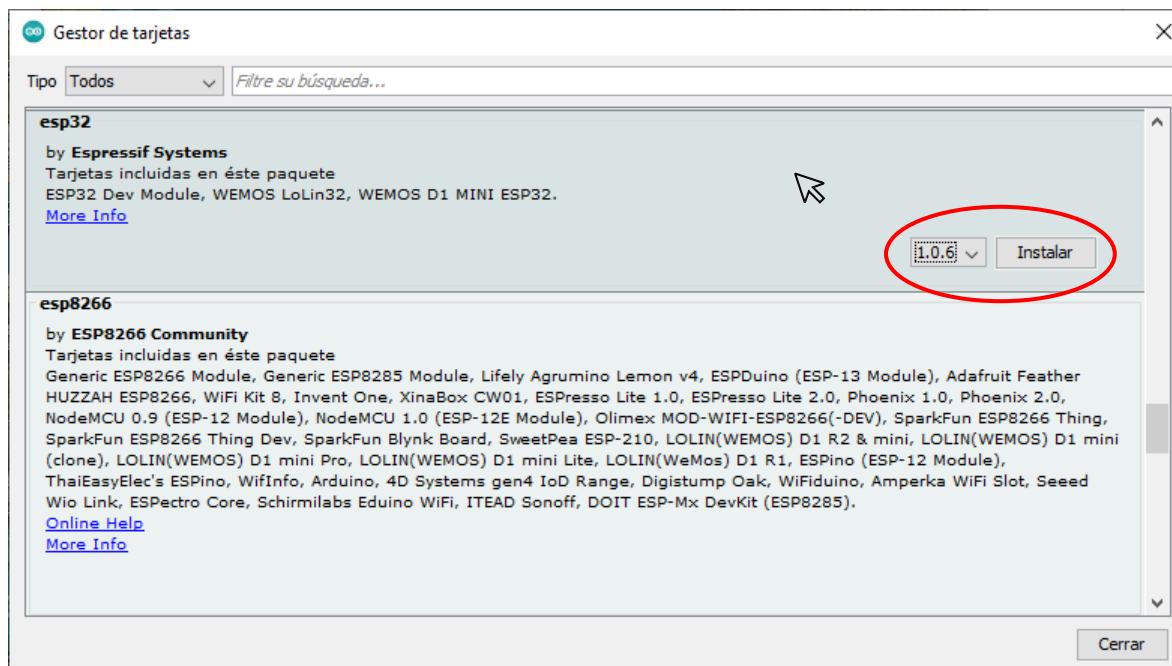
Al hacer *click* con el botón del mouse en la parte mostrada en la figura anterior, se abrirá la siguiente ventana:



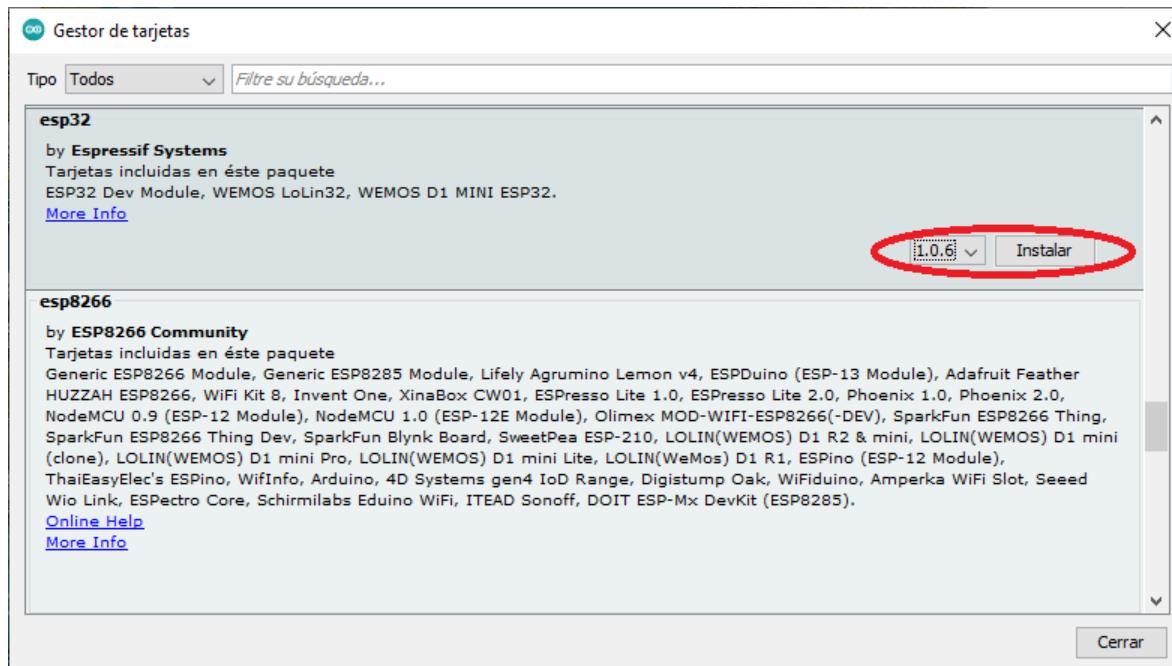
Deslice el control indicado en la figura anterior, así como se muestra a continuación:



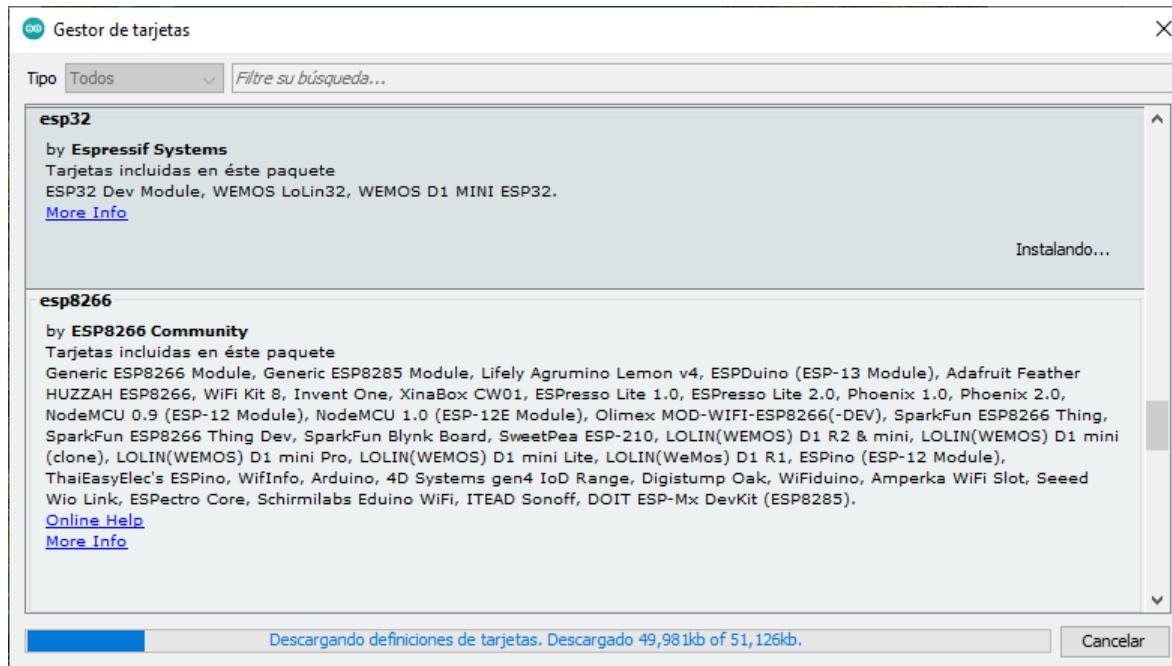
Ponga el puntero del mouse sobre el campo con el texto esp32, el área se pondrá en gris y se activarán dos botones como se muestra a continuación:



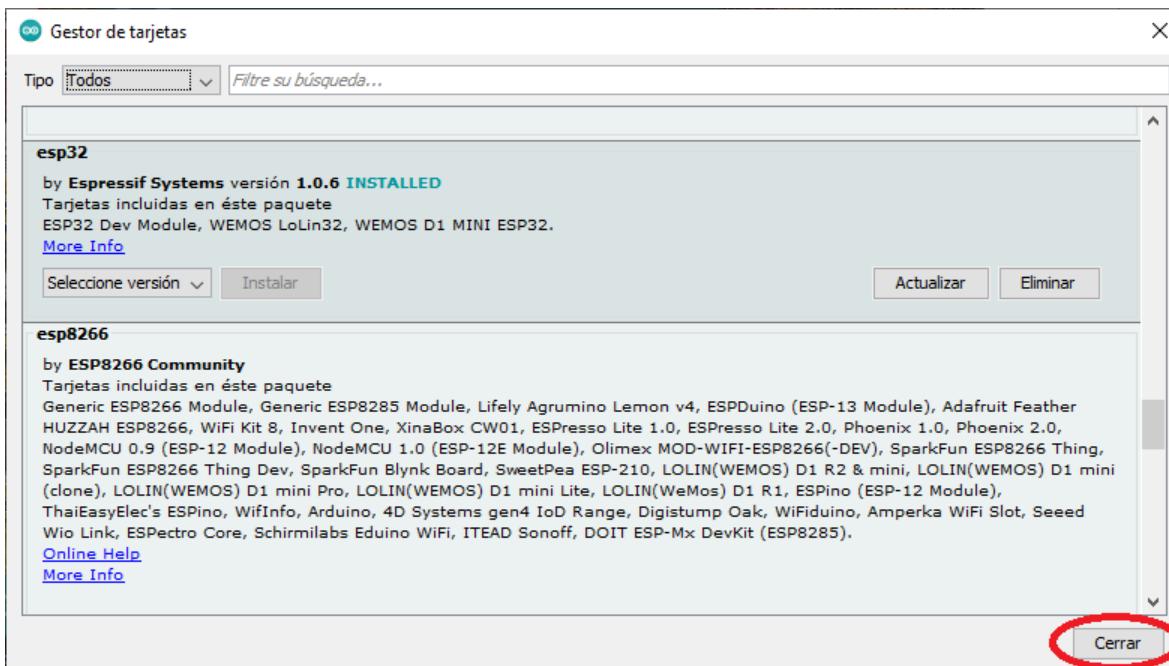
Seleccione la última versión (1.0.6) y presione el botón “Instalar” e iniciará la instalación de la versión seleccionada:



Espere a que la barra de progreso termine:



Cuando se termine de actualizar presione el botón de cerrar.

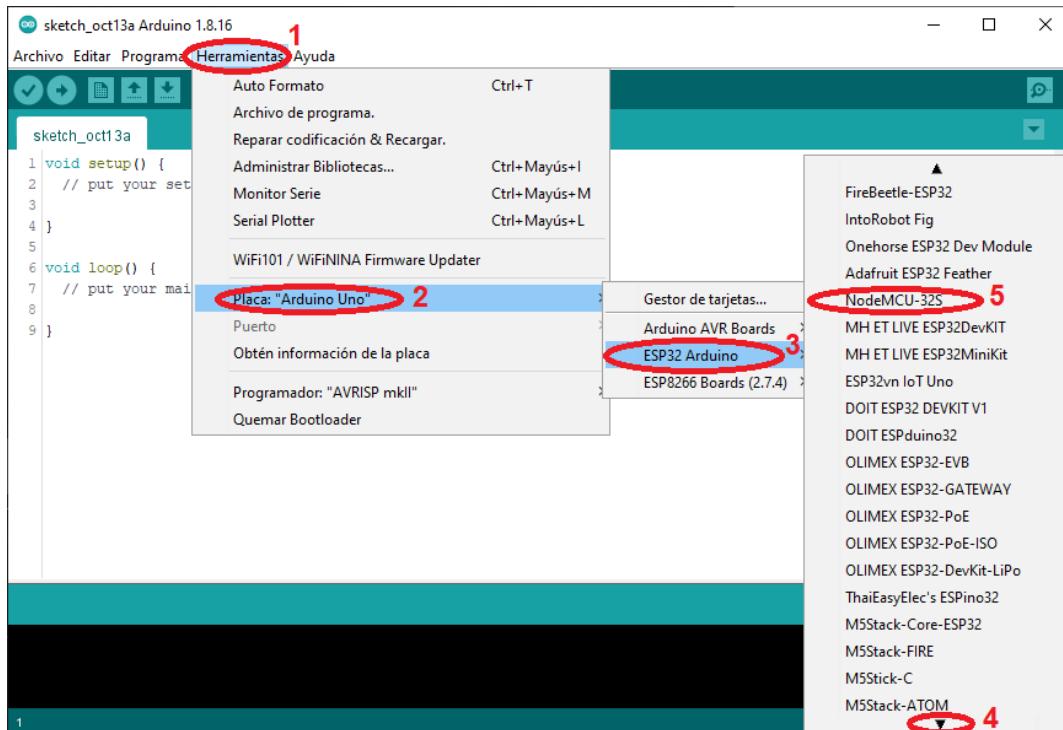


### 1.3 Ejecutando el ejemplo “blink”

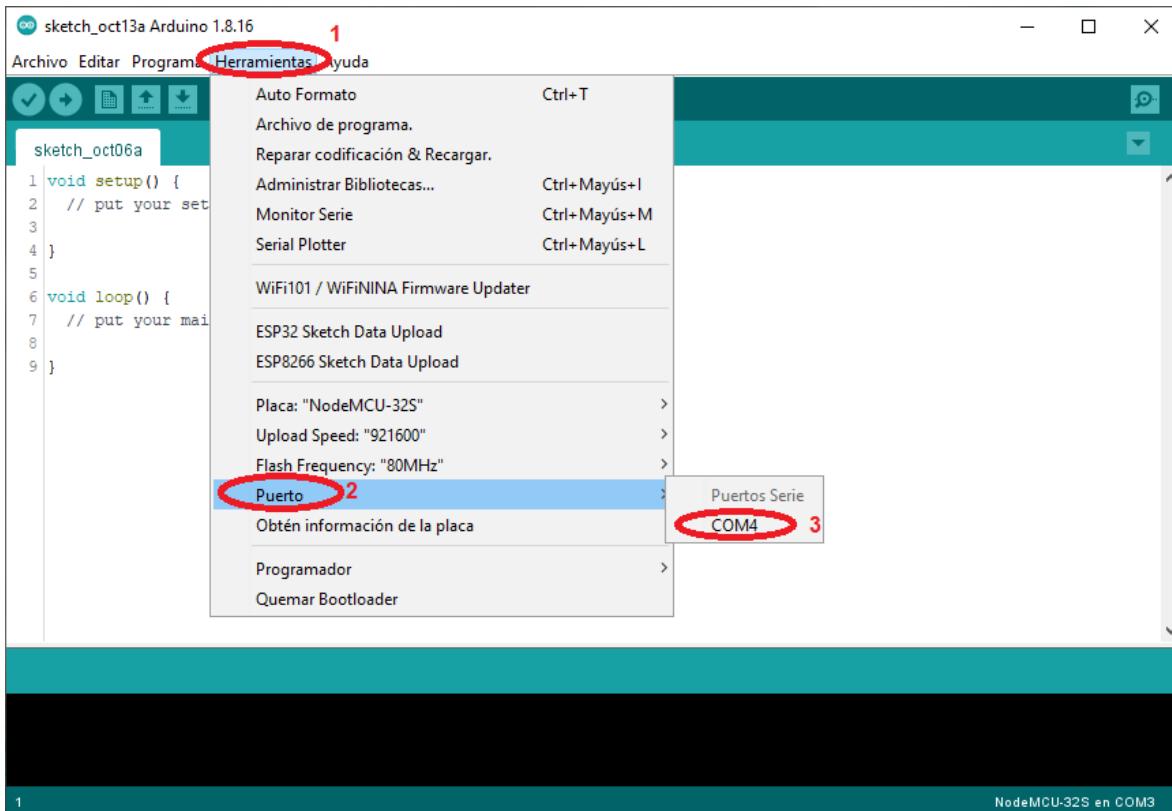
**Objetivo específico:** Elegir la tarjeta de desarrollo a utilizar en el IDE del Arduino, cargar, verificar (compilar) y descargar en la tarjeta de desarrollo un código de la lista de ejemplos que nos ofrece el desarrollador.

El **PRIMER PASO** es elegir la tarjeta de desarrollo a utilizar:

Haga **click** izquierdo en el menú **Herramientas**, después coloque el puntero del mouse en el submenú **Placa**, se abrirá un menú contextual a la derecha, coloque el puntero en **ESP32 Arduino** y después se abrirá otro menú contextual a la derecha con una lista de modelos disponibles, si la lista es demasiado larga para alcanzar en la pantalla, lleve el puntero al final de la lista y mantenga la posición del puntero sobre el triángulo que está debajo del último elemento mostrado, para que la lista se desplace hacia arriba mostrando los elementos ocultos, coloque el puntero del mouse en el modelo de su tarjeta de desarrollo y dele **click** izquierdo para seleccionarla.



Como **SEGUNDO PASO**, se requiere establecer el puerto donde se tiene conectado el dispositivo, **cconeete su tarjeta** de desarrollo y verifique el número de puerto que el sistema operativo le asigna:



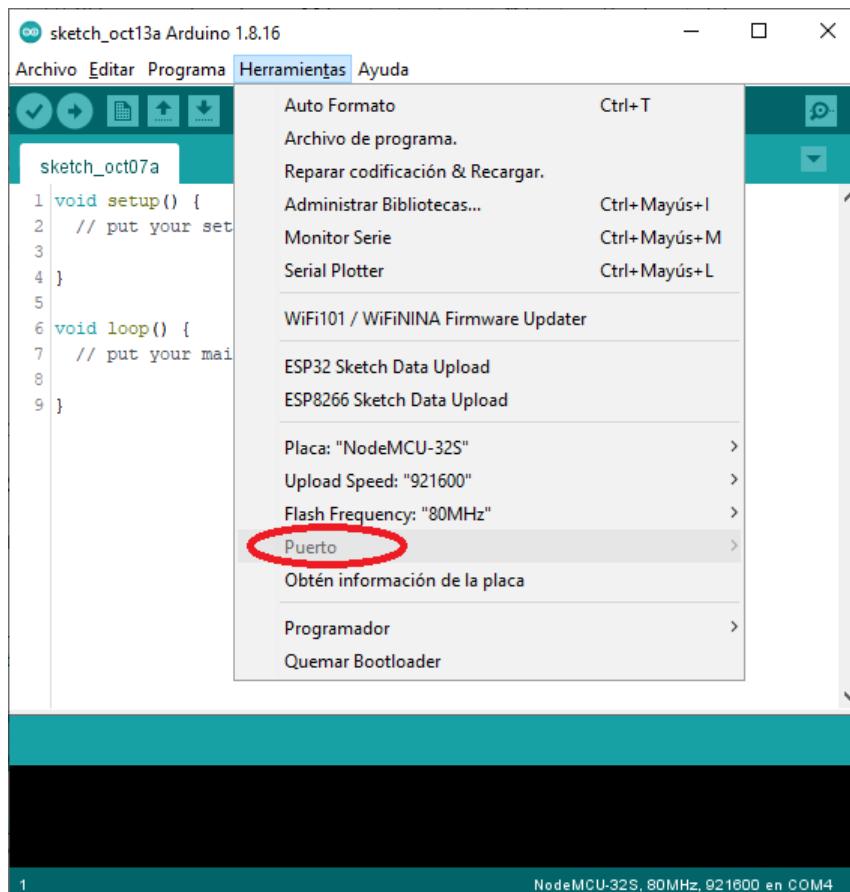
### **IMPORTANTE:**

Antes de conectar su dispositivo, verifique que no existan otros puertos ya reconocidos por el PC, si los hay, solamente tome en cuenta que, el puerto de su dispositivo será el puerto nuevo que el sistema operativo nos muestre al conectar la tarjeta de desarrollo.

No siempre se muestra el mismo número de puerto para todas las computadoras.

Para Windows 8 y 10, se requiere conectar el dispositivo y tener activado las actualizaciones de Windows. Apenas Windows detecte el nuevo dispositivo, éste lo INSTALARÁ AUTOMATICAMENTE. En éstas dos plataformas NO instale el driver manualmente (de la manera tradicional), ya que al momento de usar el dispositivo Windows sacará la clásica pantalla azul de error (BSOD).

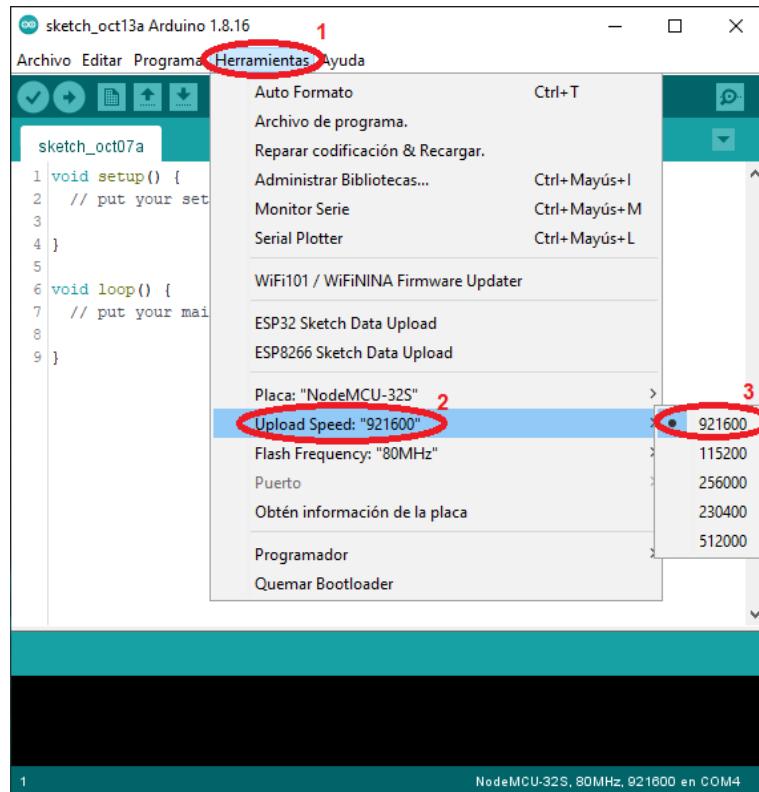
Si el puerto se muestra en gris como se indica en la siguiente figura:



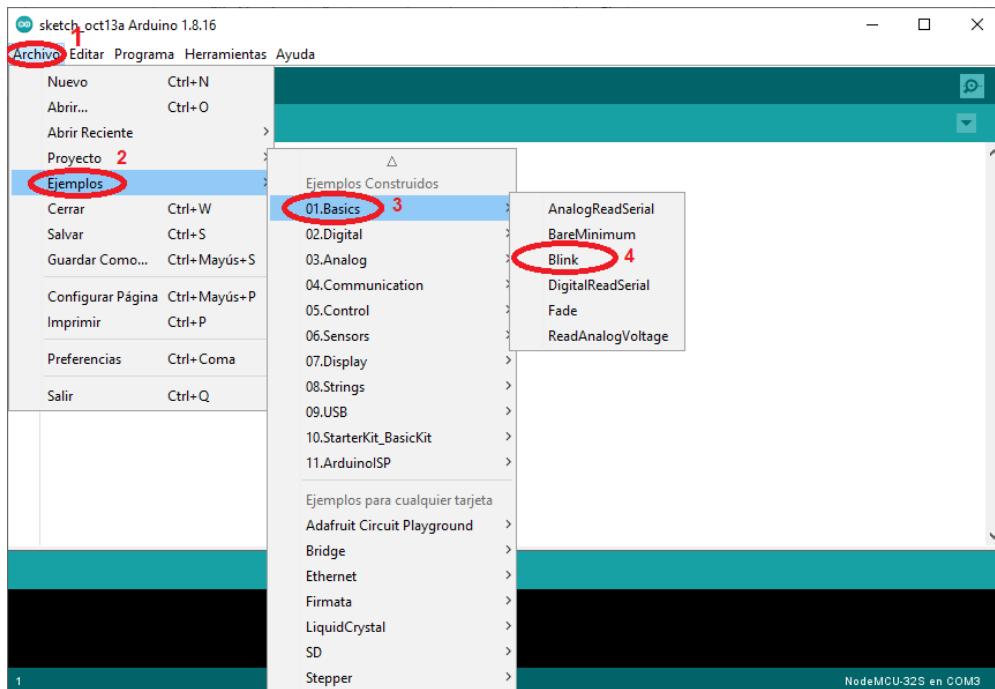
... indica que el módulo ESP32 no fue reconocido. Esto puede deberse a que:

1. La computadora no tiene acceso a Internet y no puede descargar los controladores para el dispositivo.
2. El cable USB no está firmemente conectado al módulo ESP32 ó al puerto USB de la computadora.
3. El cable USB podría estar dañado ó no es el apropiado (algunos cables USB sólo son para cargar dispositivos, no son para transferir datos).
4. El módulo ESP32 podría estar dañado (mantenga su módulo en su empaque metalizado cuando no se utilice, ya que es sensible a la electricidad estática).
5. El puerto USB de la computadora podría estar dañado.

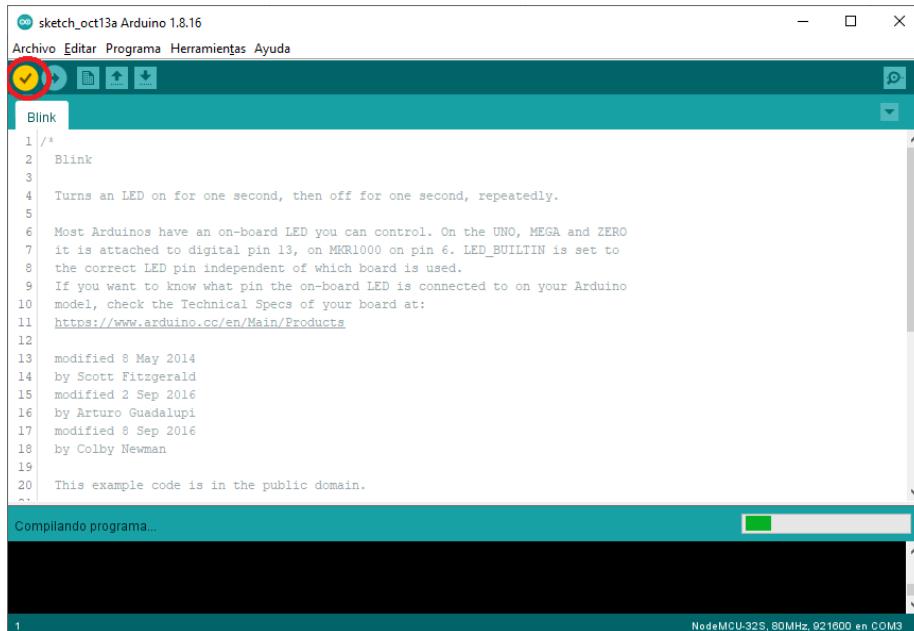
En el **TERCER PASO**, se requiere configurar la velocidad de descarga del código generado durante la compilación por el IDE del Arduino. Para eso, presionemos en la secuencia que coincide la siguiente figura:



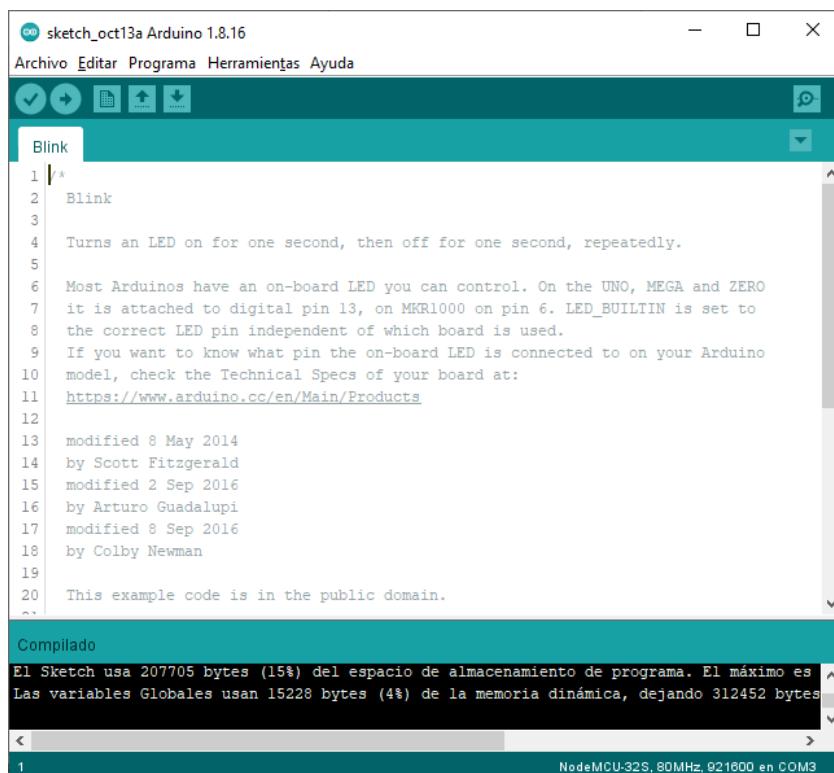
En el **CUARTO PASO**, se requiere escoger el ejemplo siguiente:



Como **QUINTO PASO** se verifica el programa presionando el botón  e indicará si hay errores:

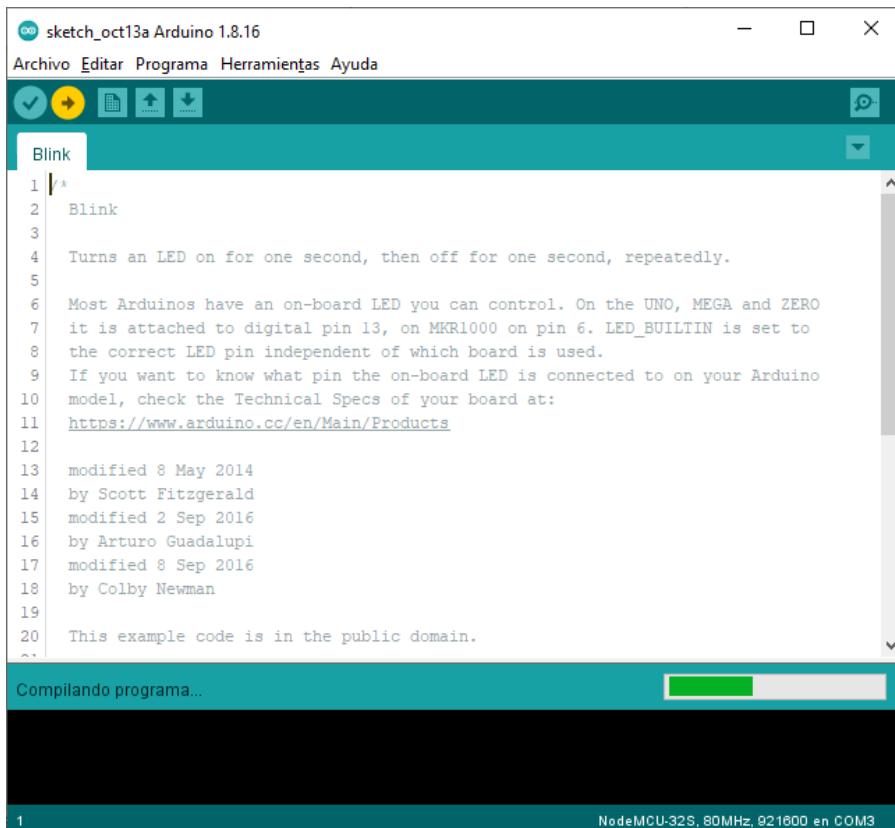


... espere que la barra de progreso termine y nos deberá de mostrar lo siguiente:



24/224

Como **SEXTO PASO**, se verifica y descarga el programa en su tarjeta de desarrollo presionando el botón  . Al presionar el botón se mostrará lo siguiente:

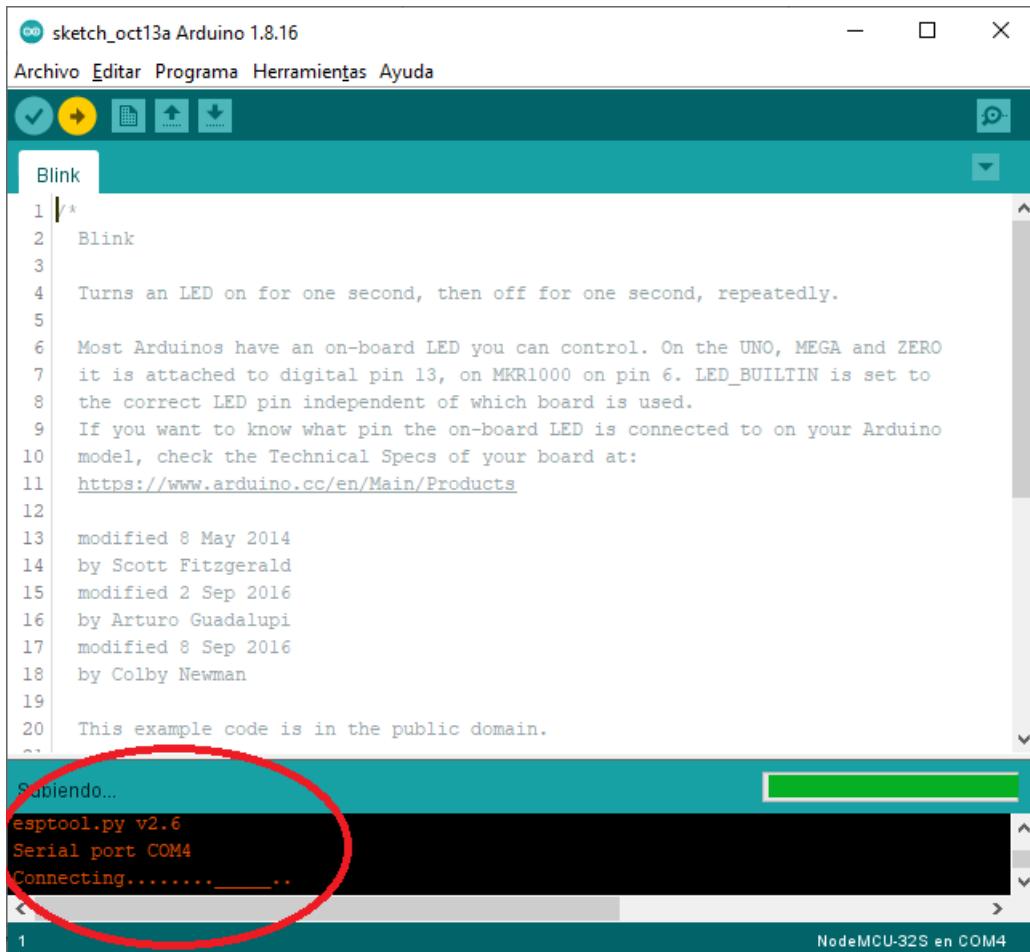


The screenshot shows the Arduino IDE interface. The title bar says "sketch\_oct13a Arduino 1.8.16". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". Below the menu is a toolbar with icons for file operations. The main window displays the "Blink" sketch code. The code is as follows:

```
1 // 
2 Blink
3
4 Turns an LED on for one second, then off for one second, repeatedly.
5
6 Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
7 it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
8 the correct LED pin independent of which board is used.
9 If you want to know what pin the on-board LED is connected to on your Arduino
10 model, check the Technical Specs of your board at:
11 https://www.arduino.cc/en/Main/Products
12
13 modified 8 May 2014
14 by Scott Fitzgerald
15 modified 2 Sep 2016
16 by Arturo Guadalupi
17 modified 8 Sep 2016
18 by Colby Newman
19
20 This example code is in the public domain.
```

Below the code editor, a status bar shows "Compilando programa..." with a progress bar. At the bottom right, it says "NodeMCU-32S, 80MHz, 921600 en COM3".

... espere a que la barra de progreso se llene, en algunos modelos de NodeMCU-32S no se requiere acción adicional para que el código empiece a descargarse, así que, no se mostrará la solicitud de conexión. De lo contrario, si se muestra lo siguiente:



```

sketch_oct13a Arduino 1.8.16
Archivo Editar Programa Herramientas Ayuda
Blink
/*
  Blink
  ...
  Turns an LED on for one second, then off for one second, repeatedly.
  ...
  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your Arduino
  model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products
  ...
  modified 8 May 2014
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman
  ...
  This example code is in the public domain.
  ...
Subiendo...
esptool.py v2.6
Serial port COM4
Connecting.....
```

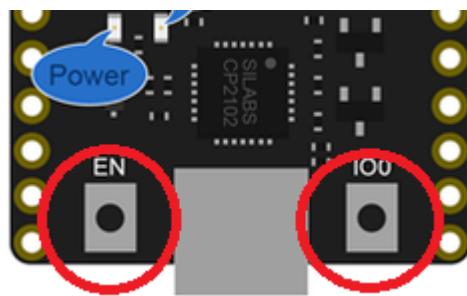
... deberá continuar con el **SEPTIMO PASO**, ya que, dependiendo del hardware adquirido, deberá probar **UNA** de las acciones siguientes:

- Presione momentáneamente el botón IO0 y el código deberá empezar a descargarse a su módulo NodeMCU-32S.

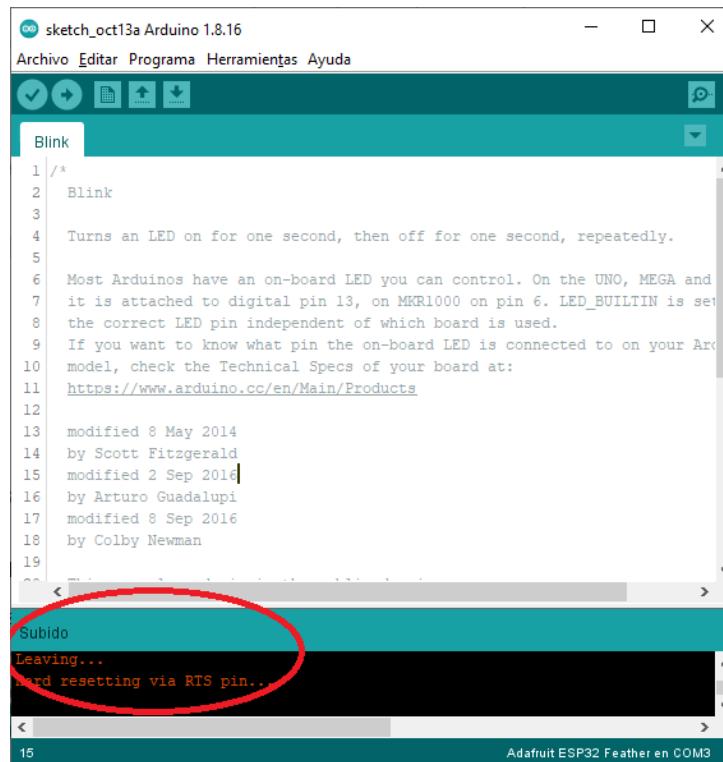


b) Presione y mantenga presionado el botón IO0, seguidamente presionar momentáneamente (presione y libere) el botón de EN ó reset y por último liberar el botón IO0. Siguiendo esto pasos el código deberá empezar a descargarse a su módulo NodeMCU-32S.

Ambos botones están ubicados a los costados del puerto microUSB de su módulo NodeMCU-32S y deberá empezar a descargar el código a su módulo NodeMCU32S.



En ese momento, el código empezará a transferirse a su tarjeta y al finalizar, el código se ejecutará de manera automática y se muestra un mensaje en la parte inferior izquierda del IDE como se muestra a continuación:



```

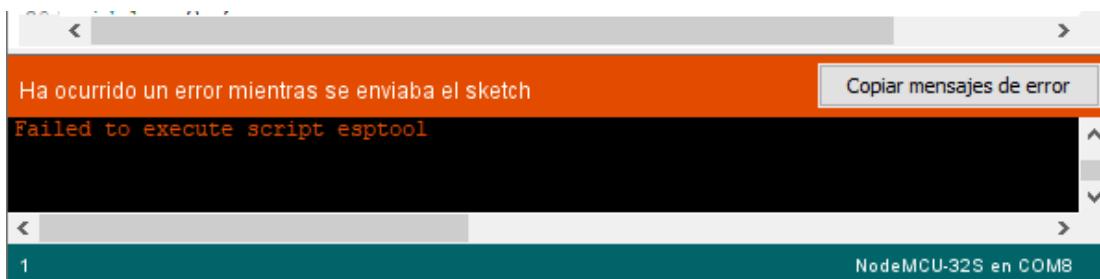
sketch_oct13a Arduino 1.8.16
Archivo Editar Programa Herramientas Ayuda
Blink
1 /*
2  * Blink
3
4  * Turns an LED on for one second, then off for one second, repeatedly.
5
6  * Most Arduinos have an on-board LED you can control. On the UNO, MEGA and
7  * it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set
8  * the correct LED pin independent of which board is used.
9  * If you want to know what pin the on-board LED is connected to on your Ar-
10 * model, check the Technical Specs of your board at:
11 https://www.arduino.cc/en/Main/Products
12
13 modified 8 May 2014
14 by Scott Fitzgerald
15 modified 2 Sep 2016
16 by Arturo Guadalupi
17 modified 8 Sep 2016
18 by Colby Newman
19
Subido
Leaving...
Hard resetting via RTS pin...

```

## Errores en la descarga del código

### Falla No. 1

Si en lugar de empezar a descargar el código hacia la tarjeta NodeMCU, le presenta este error:

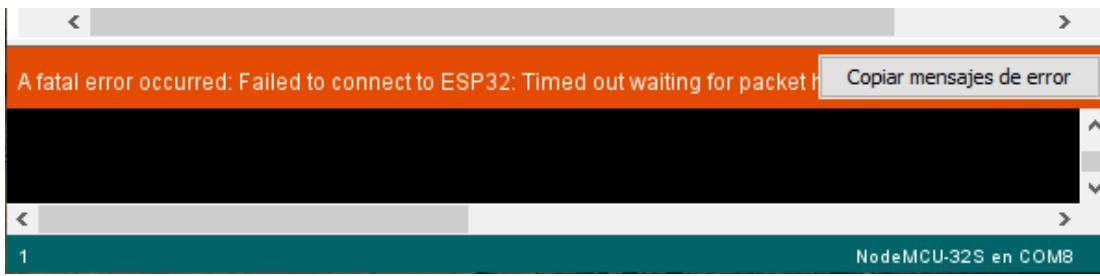


...verifique lo siguiente:

- El módulo NodeMCU **NO** se encuentra conectada a su computadora. Conecte el módulo a su PC siguiendo el procedimiento indicado en el segundo paso.
- El número de puerto COM no corresponde al módulo NodeMCU, este problema se presenta generalmente donde la computadora muestra dos o más puertos instalados. Para averiguar cuál es el puerto correcto, desconecte su módulo NodeMCU de la PC y observe cual es el número de puerto que desaparece, para eso siga la secuencia descrita en el segundo paso ya que, si deja el menú de "Herramientas" activo **NO SE MOSTRARÁ** cambio alguno si desconecta o conecta su módulo NodeMCU.
- La computadora **NO** reconoce ningún puerto COM instalado. En este caso, verifique procedimiento descrito en el segundo paso.

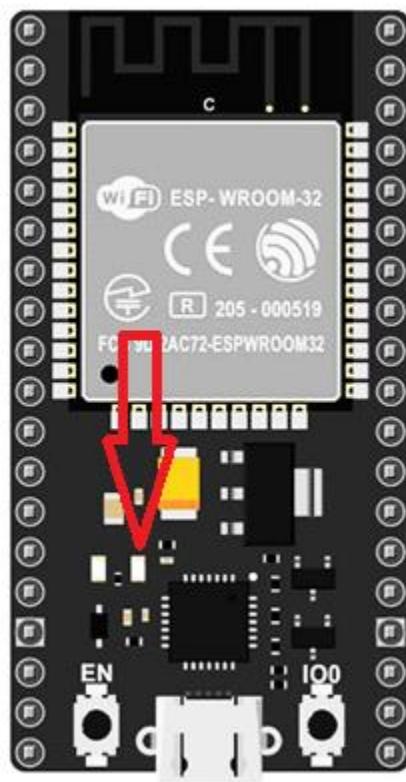
### Falla No. 2

Otro error típico es el siguiente:



Esto indica que no siguió el procedimiento descrito en el séptimo paso. Siga el procedimiento descrito en el séptimo paso.

Al terminar la descarga del código anterior en el módulo NodeMCU, observe con atención su módulo. Si todo salió bien, debería de ver un led parpadeando continuamente.

**IMPORTANTE:**

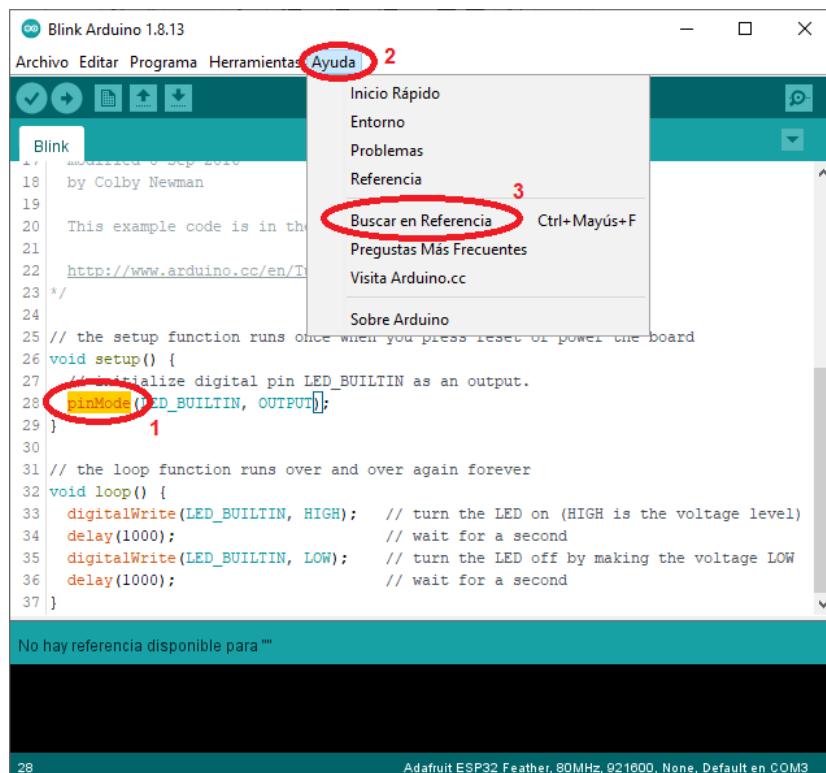
1. Algunas tarjetas ESP32 **NO** traen incluidos un led en el puerto GPIO, por lo que deberá ser agregada externamente en el pin GPIO2 (En el caso de la tarjeta NodeMCU-32S)
2. Si está utilizando otro modelo de hardware (diferente al NodeMCU-32S), el led incorporado puede venir implementado en otro pin (en el caso de Adafruit ESP32 feather viene implementado en el pin 13), sólo basta seleccionar su hardware correspondiente en el primer paso.
3. La definición de pines se encuentra en:  
`{X}:\Users\{nombreDeUsuario}\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.6\variants\nodemcu-32s`

## 1.4 Usando la ayuda del IDE del Arduino

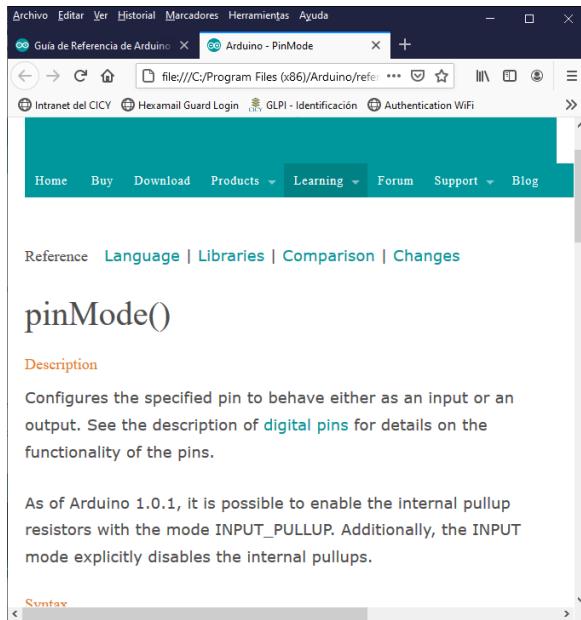
**Objetivo específico:** Aprenderá a usar la ayuda que ofrece el IDE del Arduino y usar el traductor que incorporan los navegadores Mozilla Firefox y Chrome para traducir la ayuda del inglés al español.

Para aprender a usar las sentencias es importante conocer que el IDE del Arduino nos proporciona una ayuda en el tema:

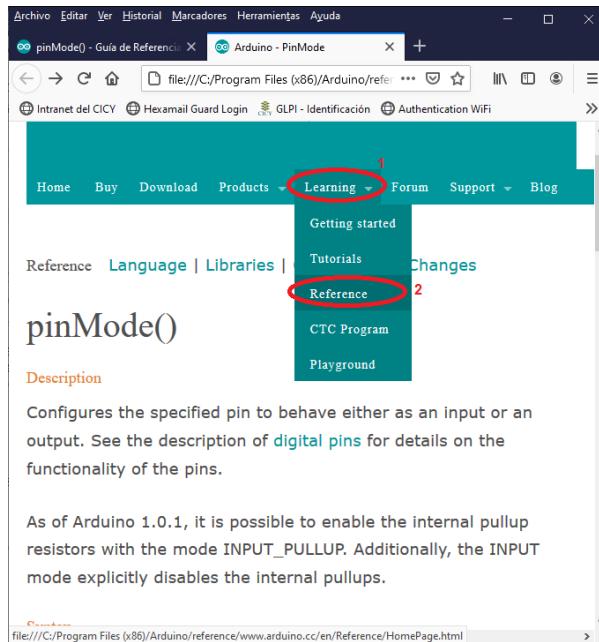
Para este punto se requiere tener abierto el ejemplo Blink.ino" (ver punto anterior), primero seleccione la sentencia que desea conocer, dé un click en la pestaña ayuda y dé un click en "Buscar en referencia", así como se muestra:



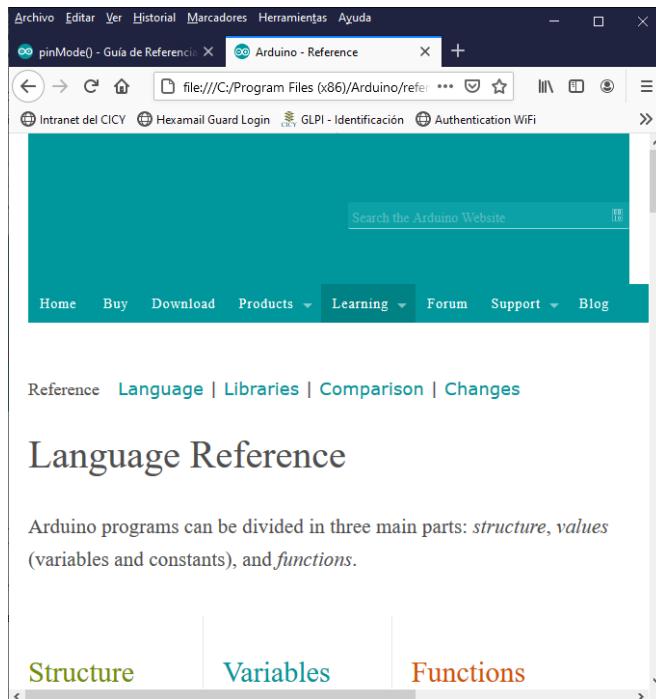
Se abrirá el navegador de internet y mostrará lo siguiente:



Desafortunadamente para algunos, la ayuda está en inglés y afortunadamente, para usar esta ayuda no se requiere internet. Si desea conocer todas las sentencias que el IDE del Arduino contiene solo presione la pestaña “Learning” y posteriormente “reference”, así como se indica:



... y se mostrará toda la ayuda disponible:

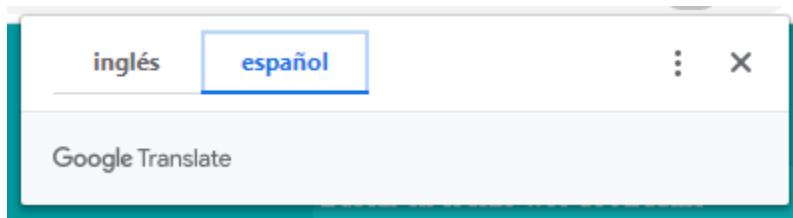


Si dispone de conectividad de internet, la ayuda la encontrará en español:

<https://www.arduino.cc/reference/es/>

... pero la descripción de las funciones estará en inglés. 😞

Una ayuda muy interesante para los aprendices es utilizar el navegador Chrome y utilizar el plugin de “Translate” (traducción):





Igualmente, al navegador Mozilla Firefox, puede descargar el plugin para la traducción:



Es importante estar pendiente de que en la traducción también se alteran las sentencias del código:

Ejemplo

```
int ledPin = 13;           // LED conectado al pin digital 13

void setup()
{
    pinMode(ledPin, OUTPUT); // establece el pin digital como salida
}

void loop()
{
    digitalWrite(ledPin, HIGH); // pone el LED en
    delay(1000);             // espera una segunda
    digitalWrite(ledPin, LOW); // establece el apagado del LED
    delay(1000);             // espera un segundo
}
```

La ayuda es magnífica, sólo ignore la traducción de las sentencias (instrucciones). La traducción correcta sería:

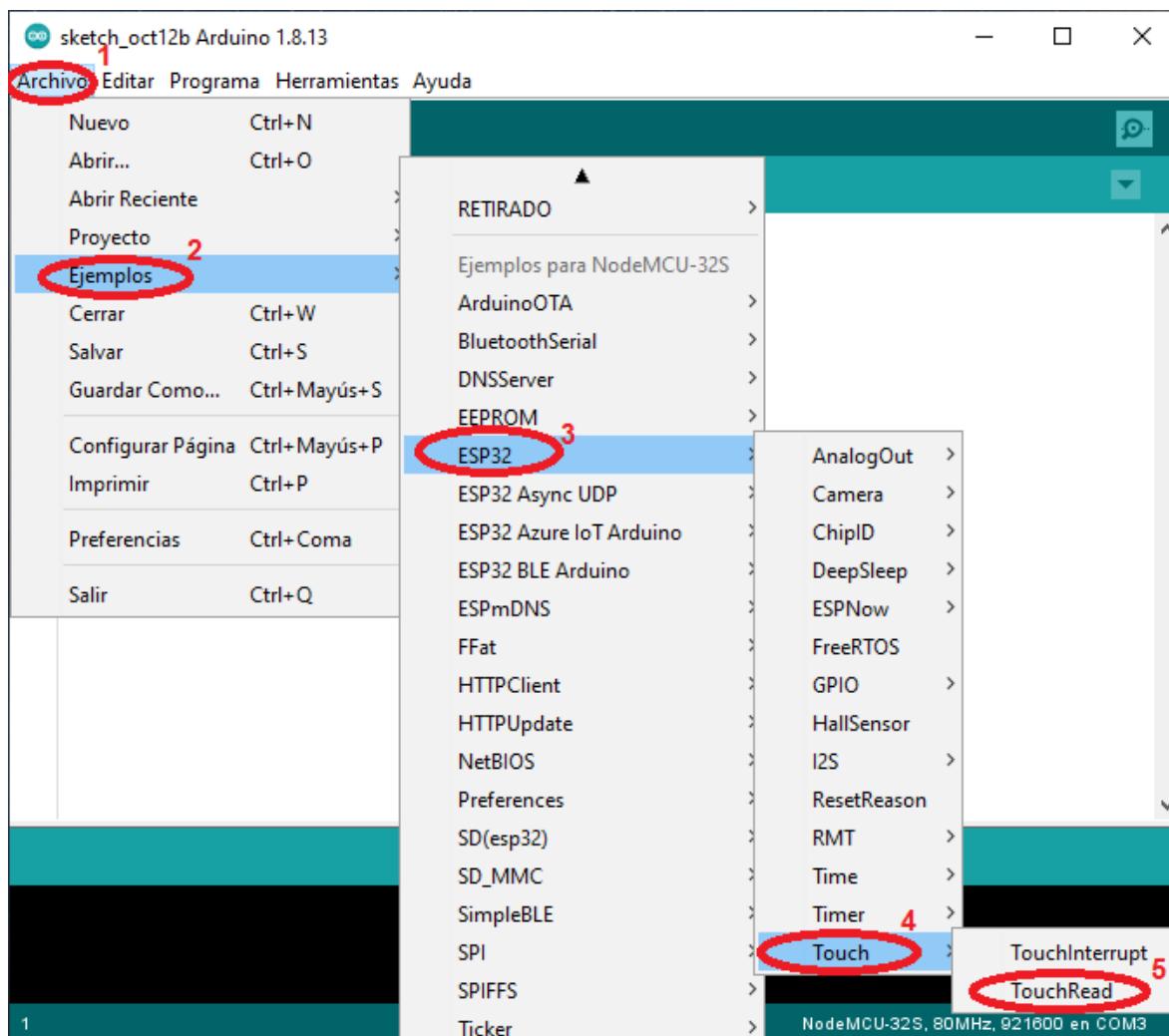
```
// la función setup() se ejecuta una vez al reiniciar o encender la placa
void setup() {
    // inicializa el pin digital llamado LED_BUILTIN como salida.
    pinMode(LED_BUILTIN, OUTPUT);
}

// la función loop() se ejecuta una y otra vez para siempre
void loop() {
    digitalWrite(LED_BUILTIN, HIGH);    // enciende el LED (HIGH es el nivel de
    delay(1000);                      // voltaje)
    digitalWrite(LED_BUILTIN, LOW);     // apaga el led haciendo el voltaje bajo
    delay(1000);                      // espera un segundo
}
```

## 1.5 Usando el monitor del puerto serie del IDE del Arduino

**Objetivo específico:** Utilizará el monitor del puerto serie para mostrar datos de variables.

Se abrirá el ejemplo TouchRead siguiendo la secuencia que se muestra a continuación:



No olvide que:



1. Con el botón  se verifica e inicia la solicitud de conexión con el módulo para la descarga del código a la tarjeta de desarrollo.
2. En algunos modelos de NodeMCU-32S no se requiere acción adicional para que el código empiece a descargarse, así que, no se mostrará la solicitud de conexión. El código se descargará automáticamente.

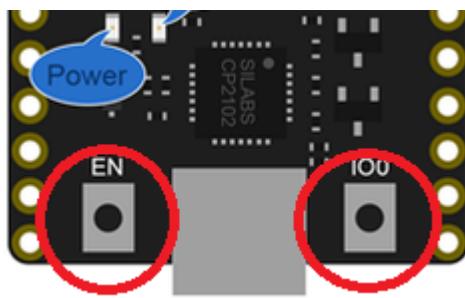
3. En otros modelos de hardware, mostrará una solicitud de conexión. En la parte inferior del IDE se mostrará la leyenda “conectando” así como se muestra en la siguiente figura:



... dependiendo del hardware adquirido, puede probar **UNA** de las acciones siguientes:

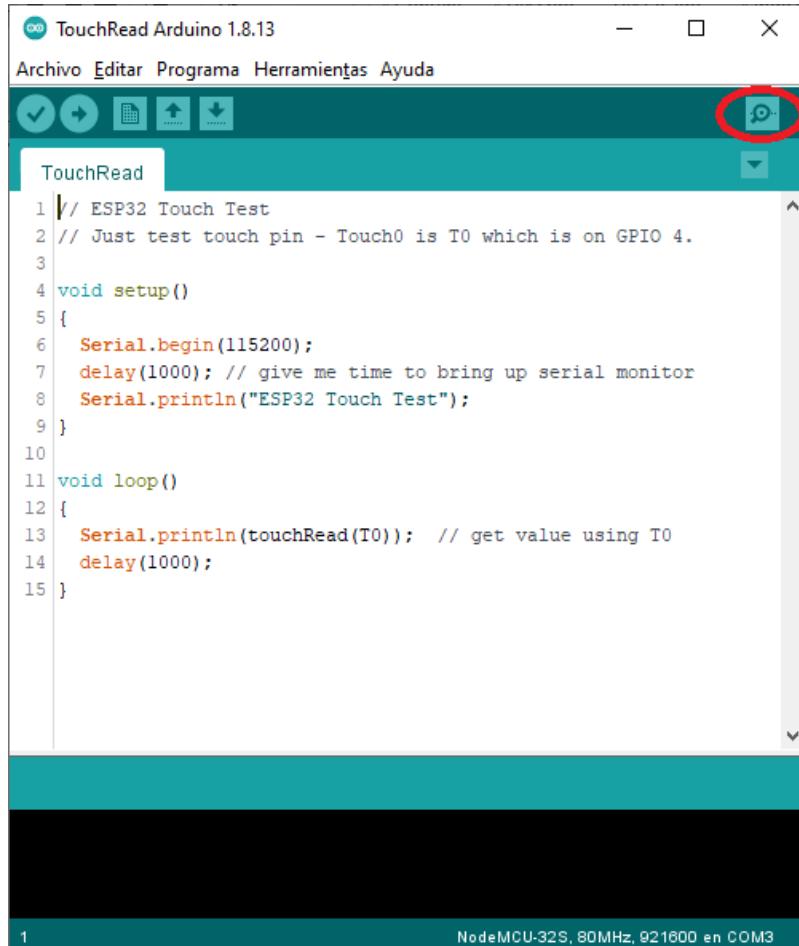
- Presione momentáneamente el botón IO0 y el código deberá empezar a descargarse a su módulo NodeMCU-32S.
- Presione y mantenga presionado el botón IO0, seguidamente presionar momentáneamente el botón de EN ó reset y por último liberar el botón IO0. Siguiendo éstos pasos el código deberá empezar a descargarse a su módulo NodeMCU-32S.

Ambos botones están ubicados a los costados del puerto microUSB de su módulo NodeMCU-32S y deberá empezar a descargar el código a su módulo NodeMCU32S.



4. Si la barra de notificación (intermedia entre la zona del código y la zona negra) del IDE cambia a naranja indica que ocurrió un error en la descarga, para corregirlo siga el procedimiento descrito al final del punto 1.3 “Errores en la descarga del código”.

Cuando el código se haya descargado, ubíquese en la parte superior derecha del IDE del Arduino y dé un click al botón del mouse, así como se muestra a continuación:



The screenshot shows the Arduino IDE interface with the title bar "TouchRead Arduino 1.8.13". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". Below the menu is a toolbar with icons for file operations. The main window displays a sketch named "TouchRead" with the following code:

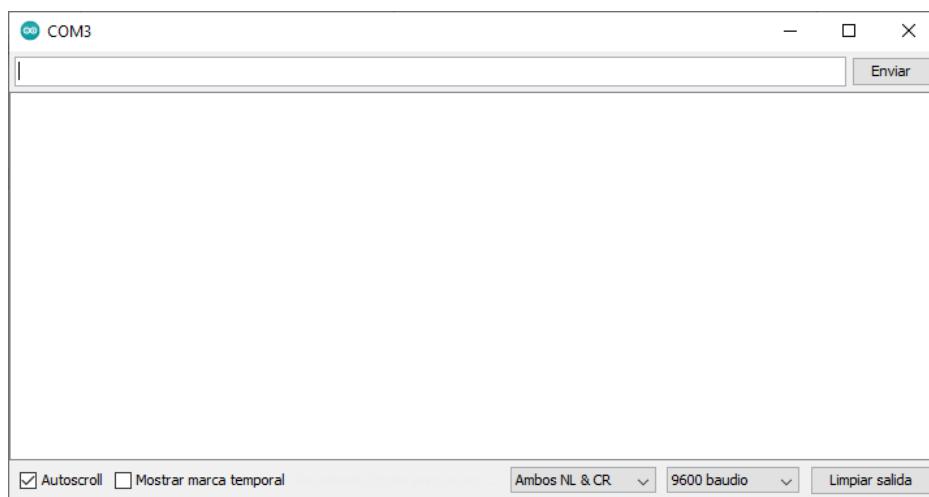
```
// ESP32 Touch Test
// Just test touch pin - Touch0 is T0 which is on GPIO 4.

void setup()
{
    Serial.begin(115200);
    delay(1000); // give me time to bring up serial monitor
    Serial.println("ESP32 Touch Test");
}

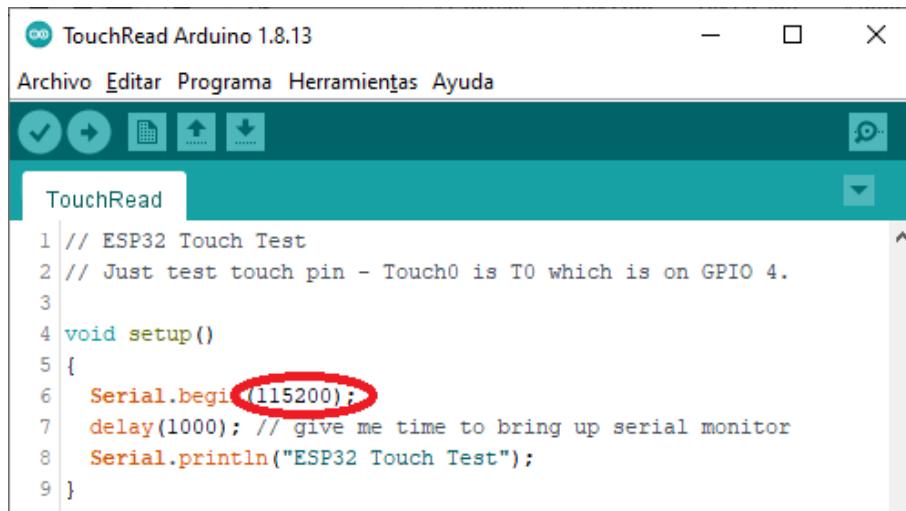
void loop()
{
    Serial.println(touchRead(T0)); // get value using T0
    delay(1000);
}
```

At the bottom of the IDE, it says "NodeMCU-32S, 80MHz, 921600 en COM3".

... e inmediatamente se abrirá una ventana como ésta:

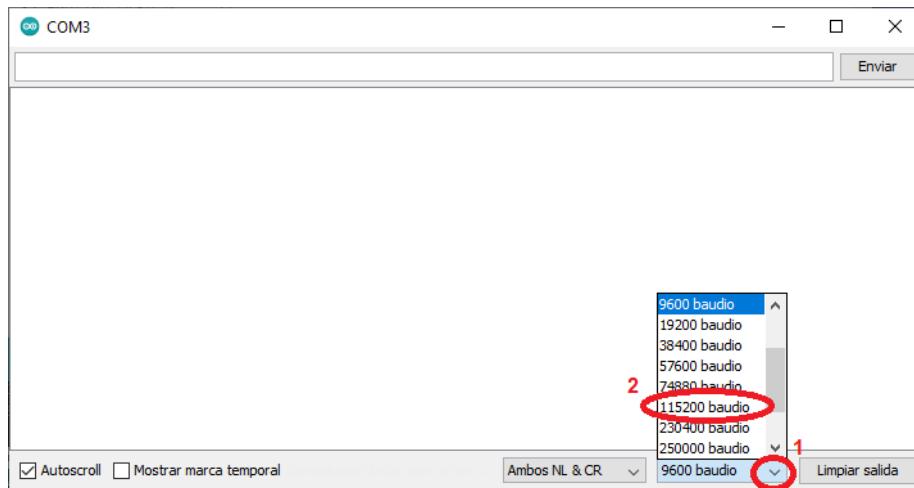


Regrese a la ventana del IDE del Arduino, observe y memorice el valor que se declara en la línea 6 del código:

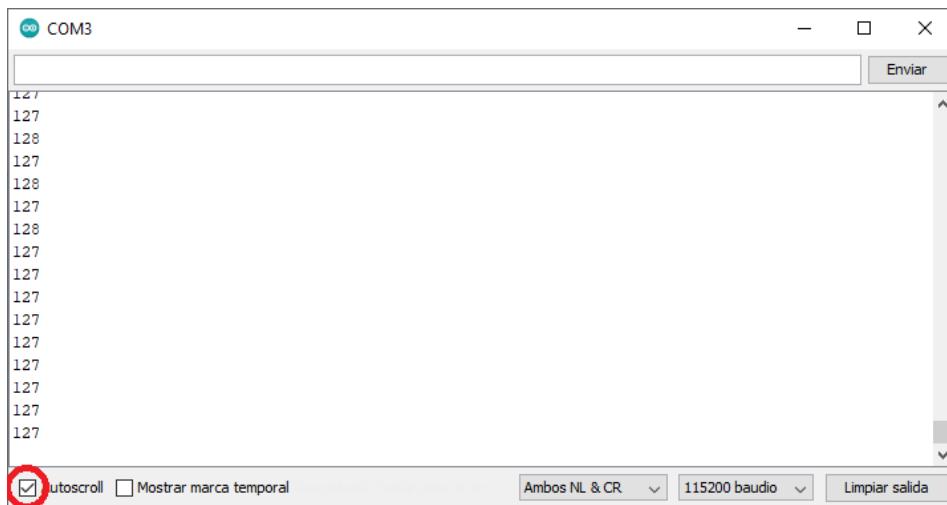


```
// ESP32 Touch Test
// Just test touch pin - Touch0 is T0 which is on GPIO 4.
void setup()
{
    Serial.begin(115200);
    delay(1000); // give me time to bring up serial monitor
    Serial.println("ESP32 Touch Test");
}
```

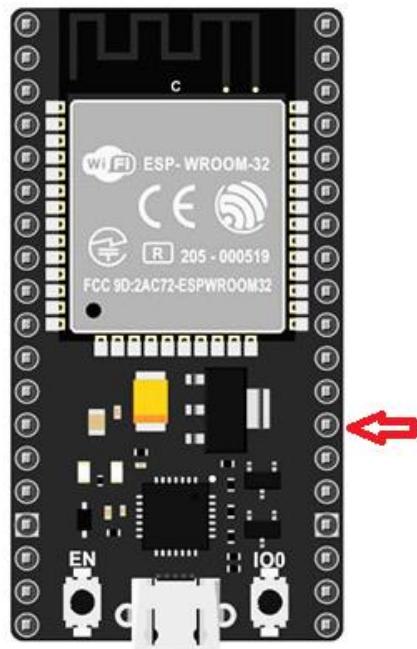
Posteriormente, regrese al IDE del Arduino y seleccione el valor declarado en la línea 6 del código, así como se muestra a continuación:



... y observe lo que se muestra en el monitor del puerto serie:



Asegúrese de haber activado el checkbox con la etiqueta “autoscroll”, esto con el objetivo de visualizar el último valor enviado por la tarjeta de desarrollo. Si todo salió bien, Usted deberá de tocar el pin con el dedo (así como se muestra en la figura) y al mismo tiempo, observar las lecturas en el monitor del puerto serie.

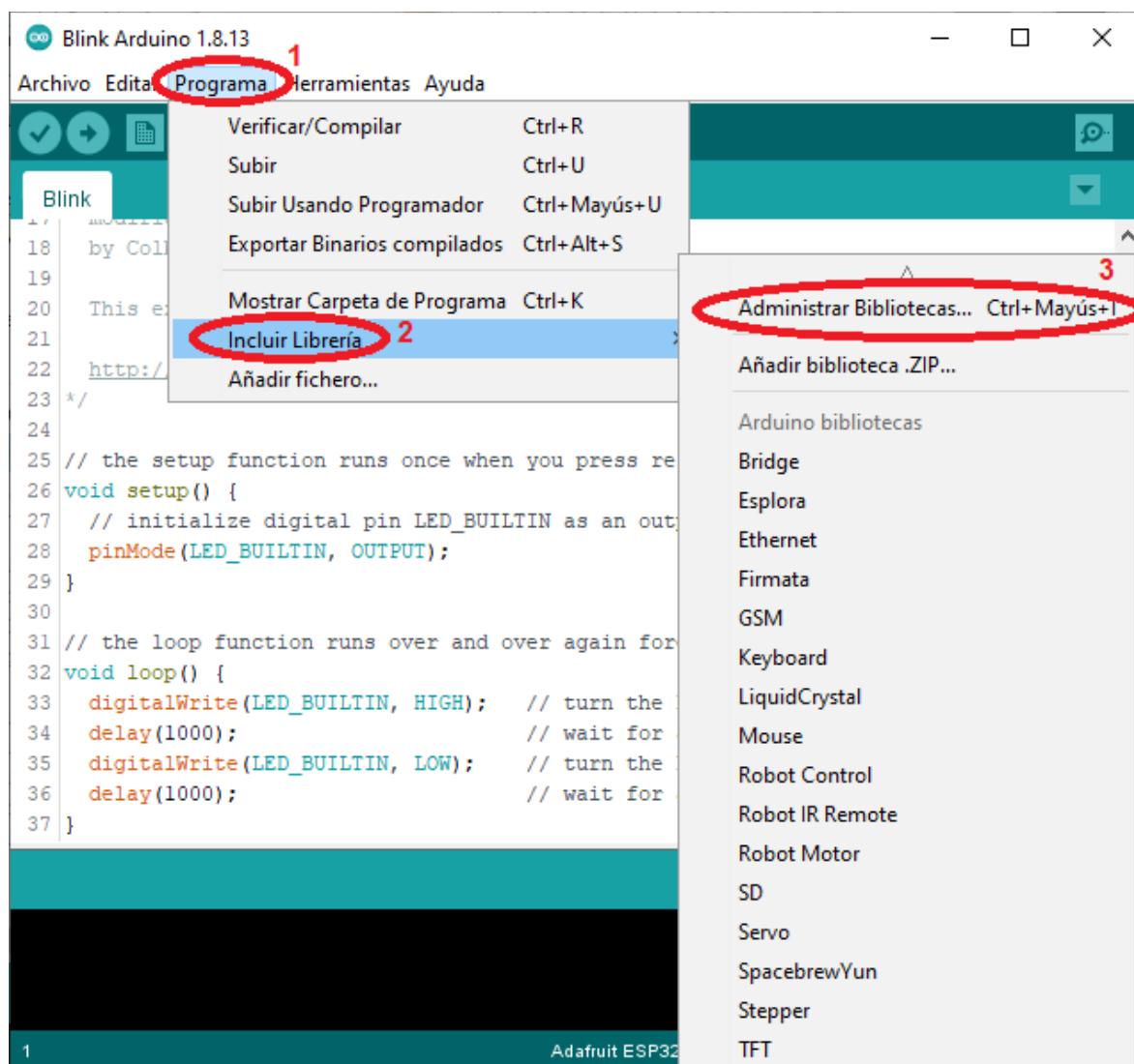


## 1.6 Agregando bibliotecas al IDE del Arduino del catálogo de drivers

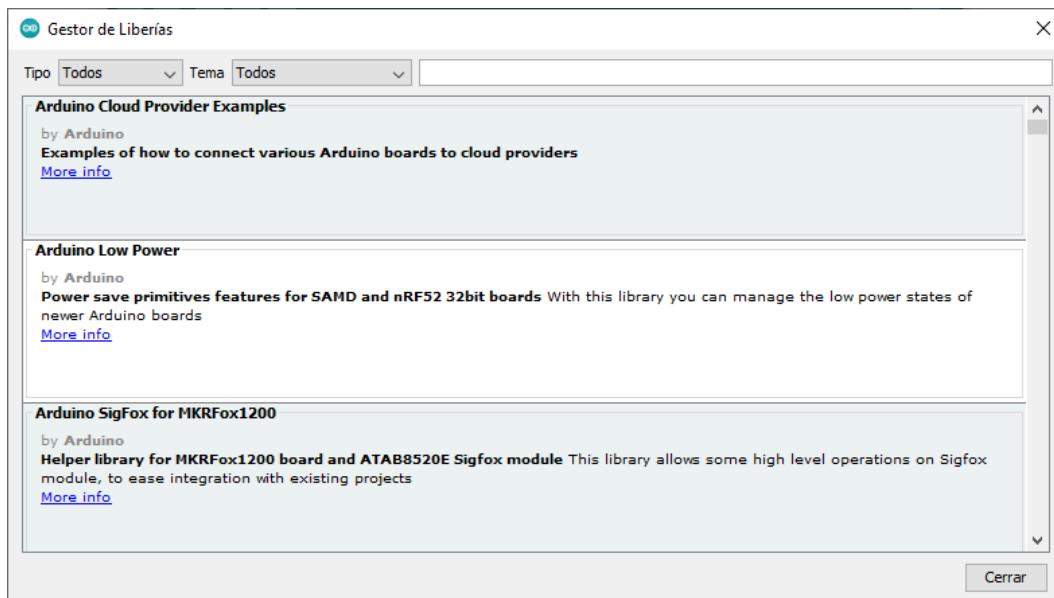
**Objetivo específico:** Aprenderá la manera de instalar las bibliotecas para el hardware que pretenda utilizar en su proyecto, en este caso: pantalla OLED Mod SSD1306, la biblioteca gfxlibrary (con sus dependencias), el sensor DS18B20 y el sensor DHT22 (AM2301).

Supongamos que se desea trabajar con la pantalla SSD1306, para ello, se requiere descargar la biblioteca del catálogo.

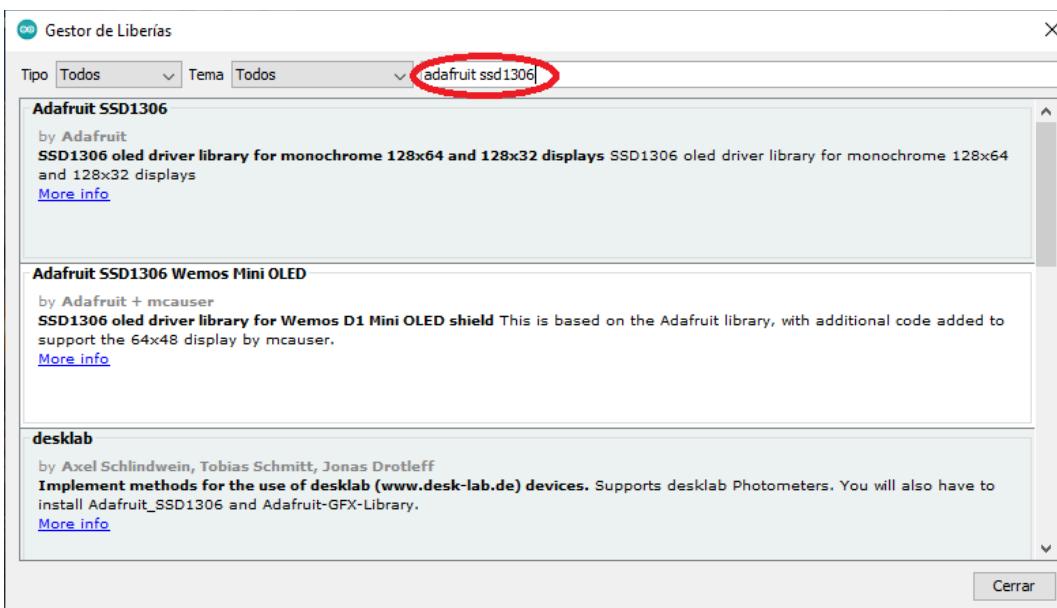
Se inicia presionando de manera secuencial las opciones que se muestran a continuación:



Y se abrirá una ventana como se muestra a continuación:

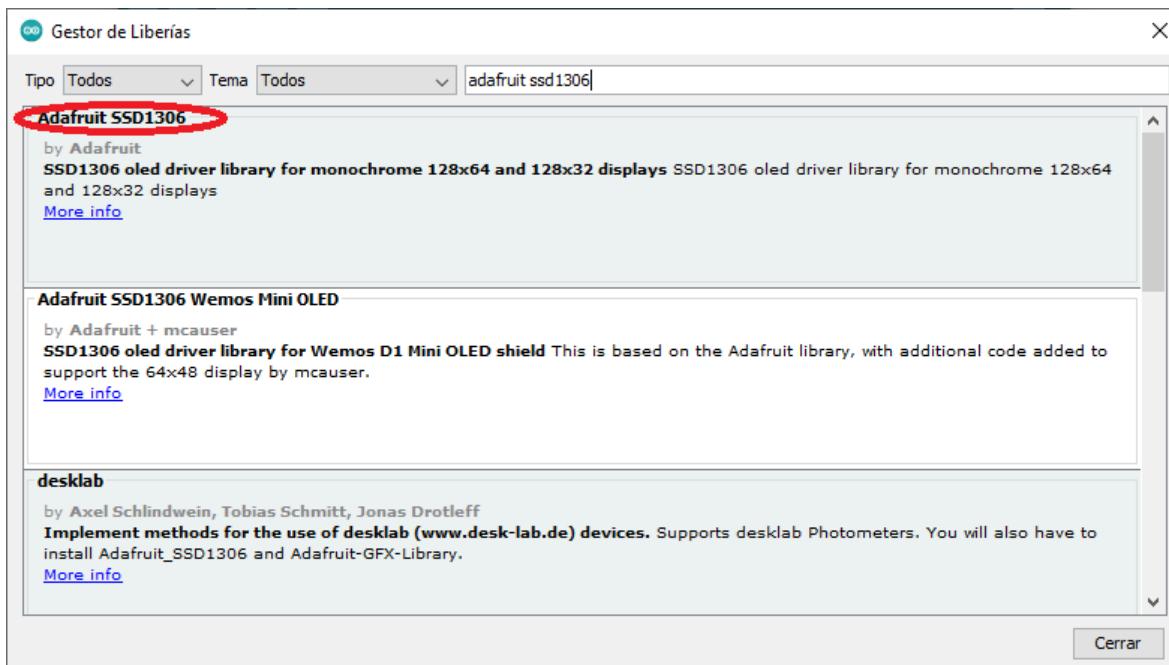


Ubique el campo de texto con la etiqueta “Tema” y escriba el nombre de la biblioteca que usted desea, en este caso escriba “adafruit ssd1306”:

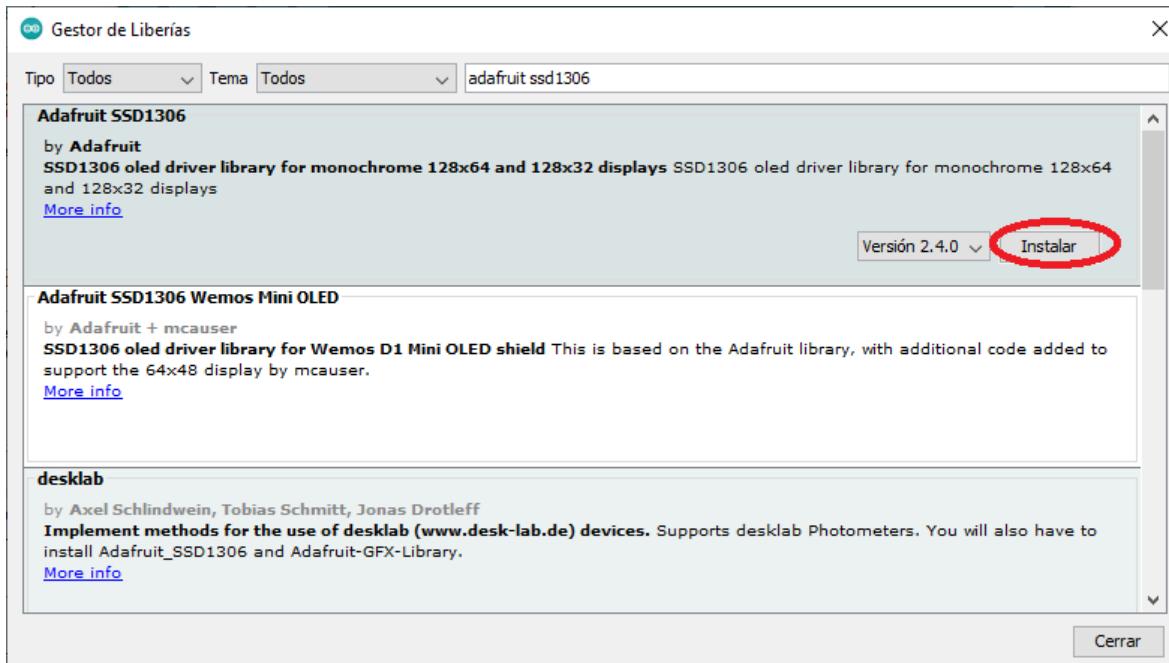


El gestor de bibliotecas automáticamente lo buscará en la base de datos y posteriormente mostrará las versiones disponibles, en este caso la biblioteca

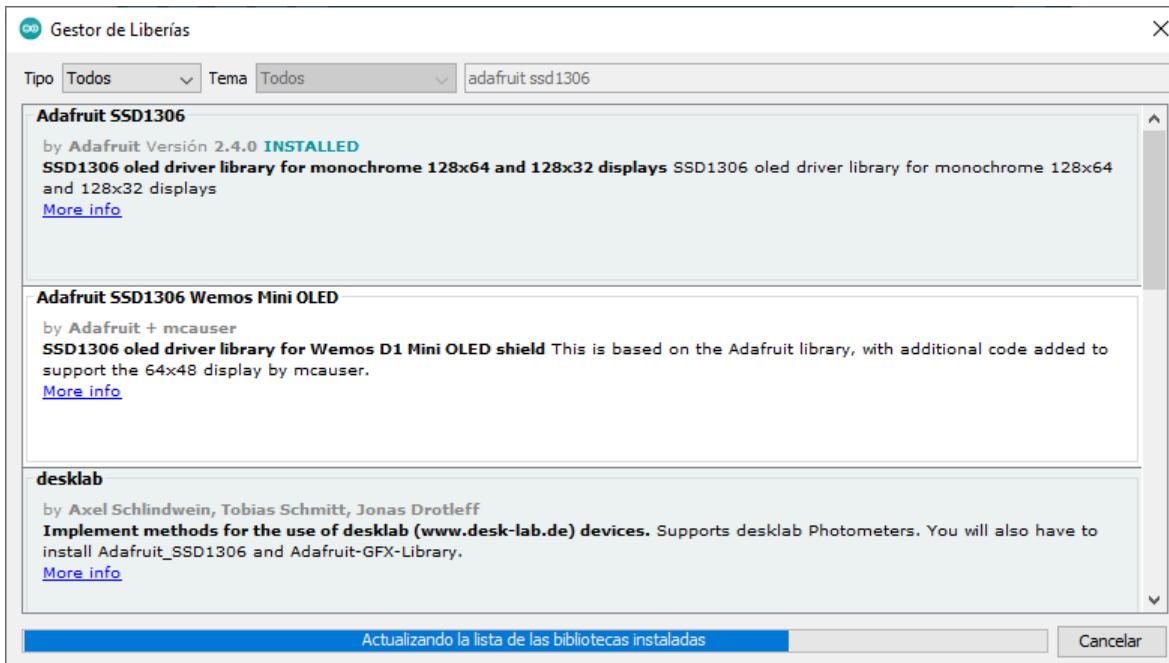
deseada, será la que se muestra primero:



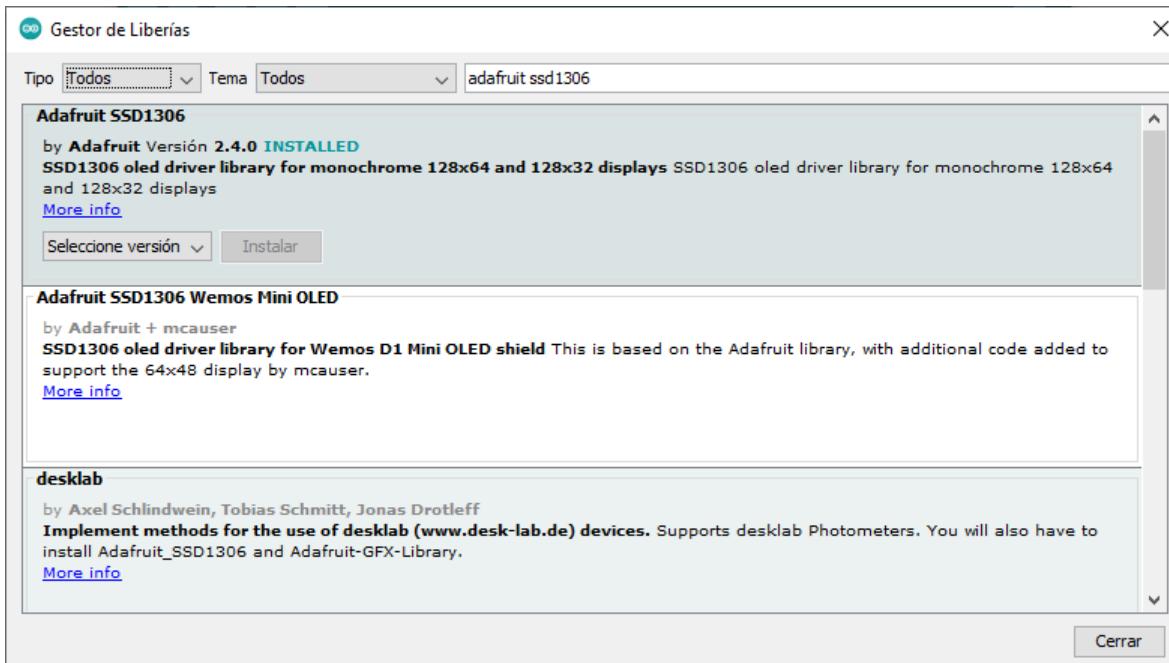
Ponga el cursor del mouse en el campo con el texto Adafruit SSD1306, se mostrarán la versión y un botón llamado “instalar”, presione el botón “Instalar”.



Espere a que la barra de progreso se complete:

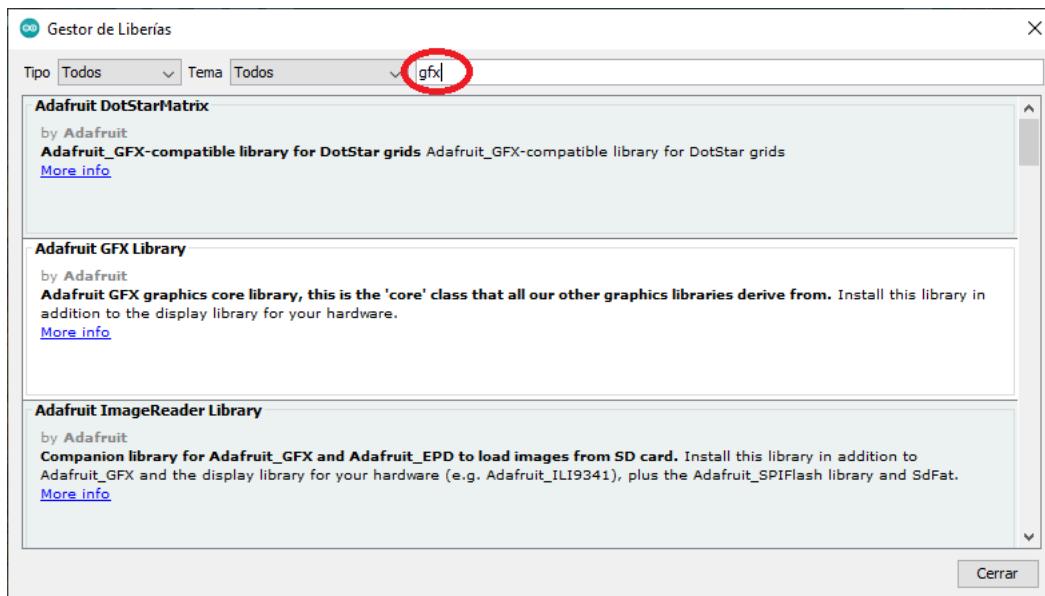


Cuando en la esquina inferior derecha se muestre el botón “cerrar”, indicará que la biblioteca ha sido instalada, sin embargo, aún no presione el botón:

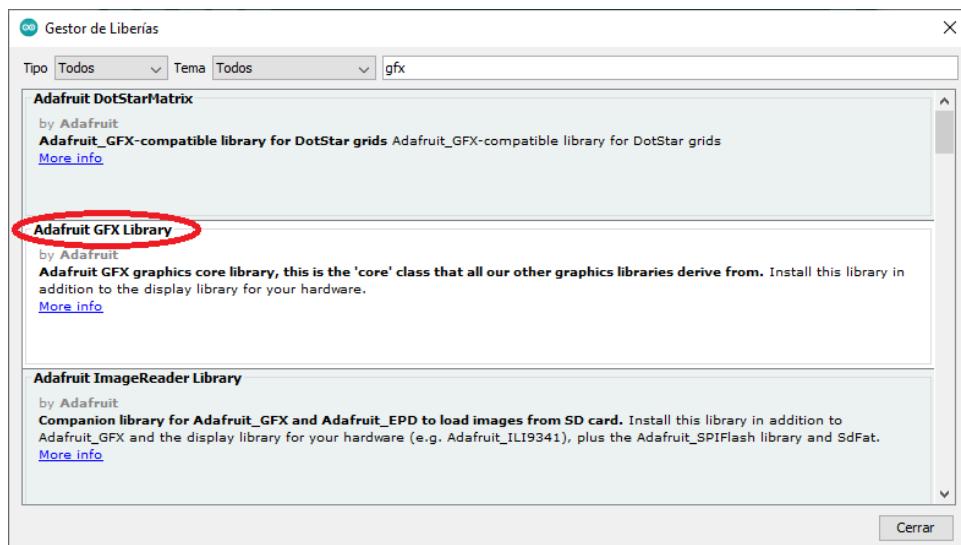


Esta biblioteca depende de otra llamada GFX Library y se instala de la misma

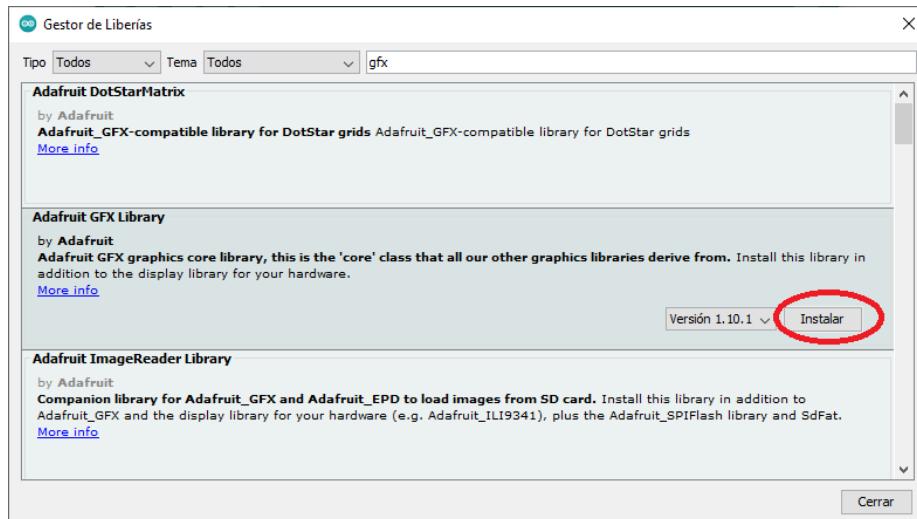
manera, borre lo anterior y teclee “gfx” en el campo de tema, así como se muestra:



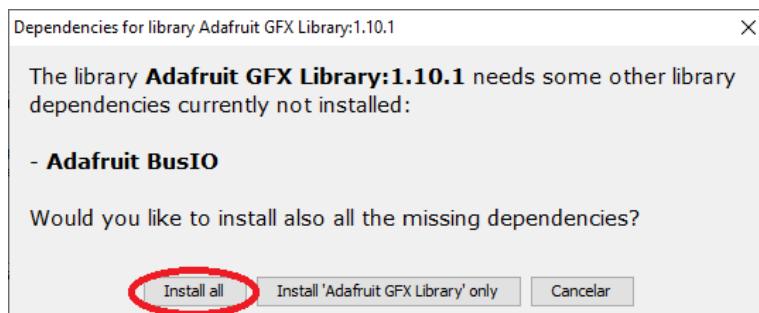
... la biblioteca deseada se muestra inmediatamente:



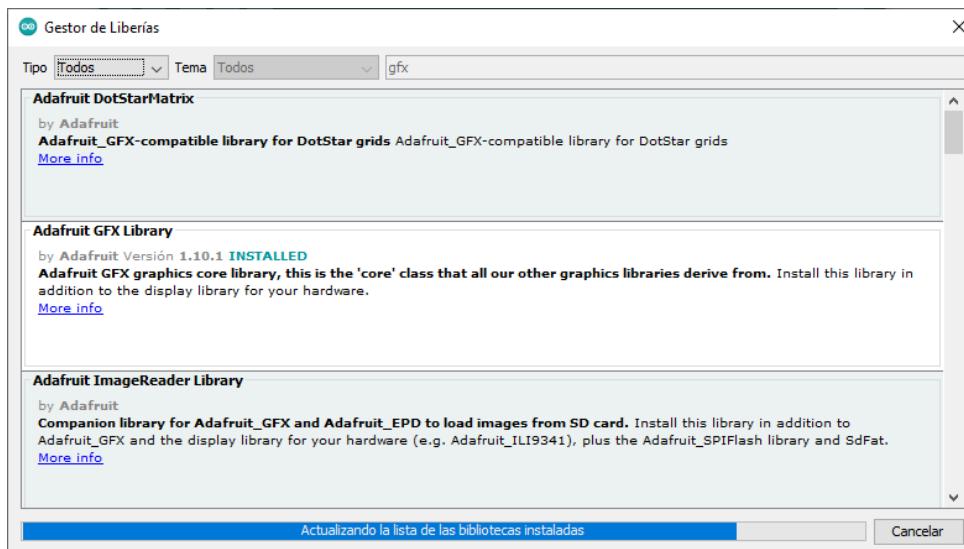
Ponga el mouse sobre el campo mostrado en la figura anterior, el cual tiene el texto “Adafruit GFX Library” y se mostrarán la versión y el botón “instalar”:



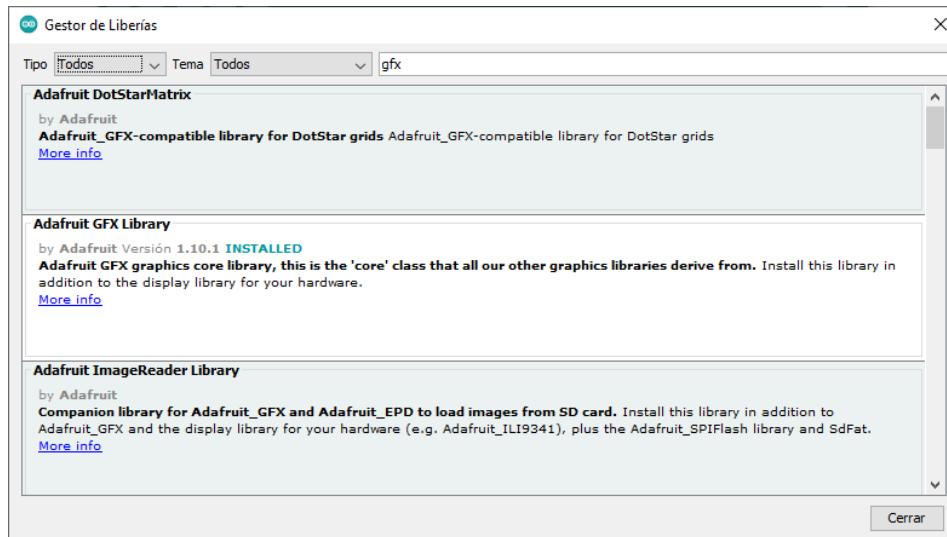
Al presionar instalar, le mostrará la siguiente ventana:



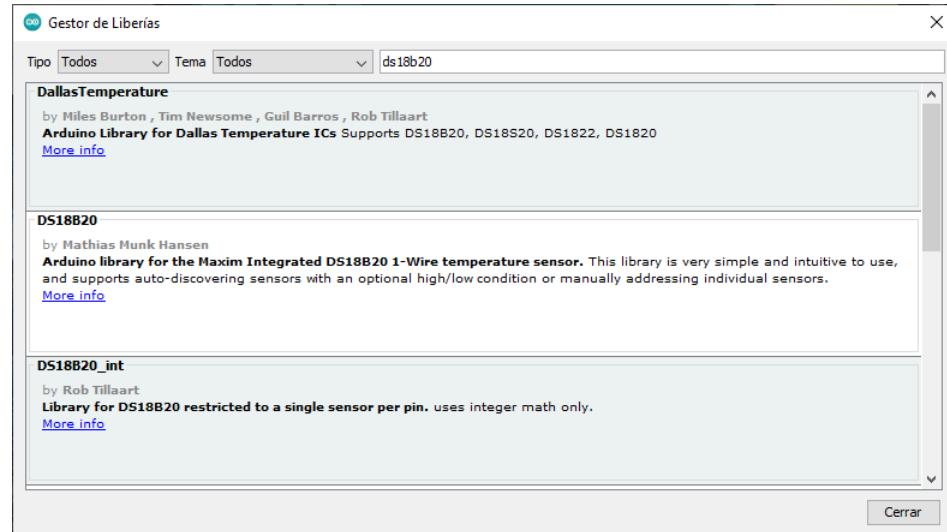
... presión el botón “Install all” y espere a que la instalación finalice:



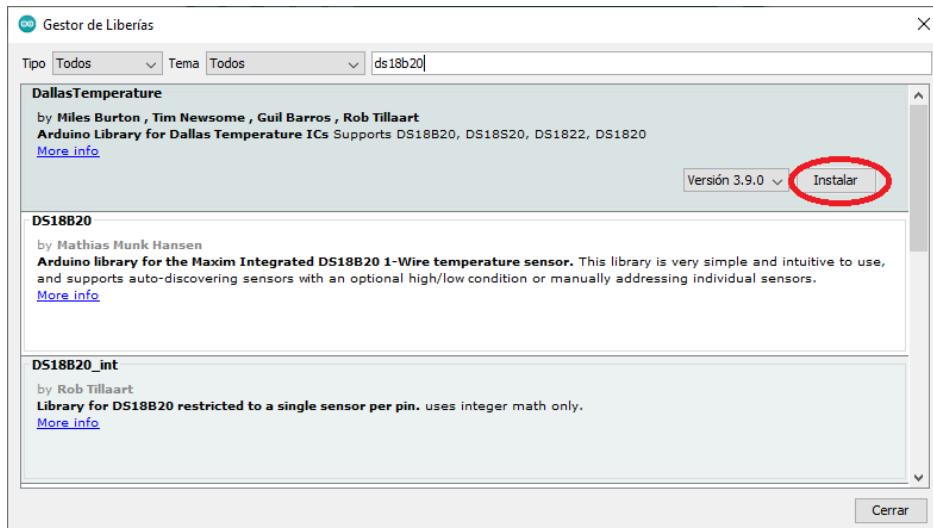
Cuando la instalación finalice, aún no presione el botón cerrar:



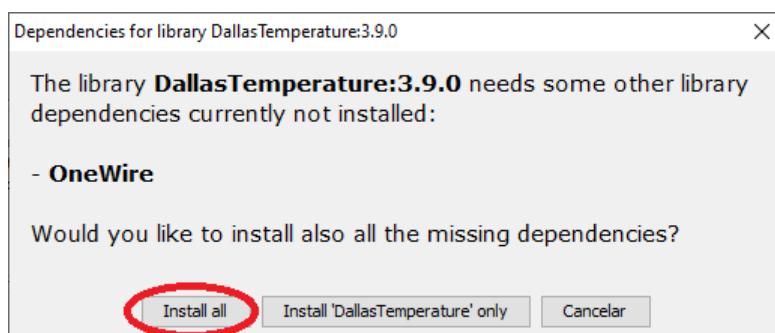
Aún queda pendiente la instalación de la biblioteca DS18B20 que se utilizará en el proyecto, así que, borre lo anterior y escriba “ds18b20” en el campo marcado como “Tema”, así como se muestra:



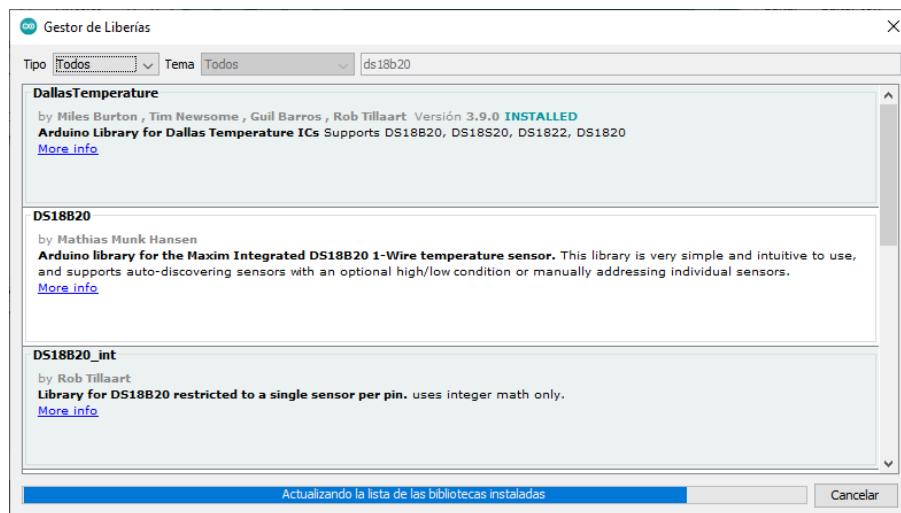
Ponga el cursor de mouse en el área del campo llamado “Dallas temperature” y se activarán los campos con la versión y el botón de instalar:



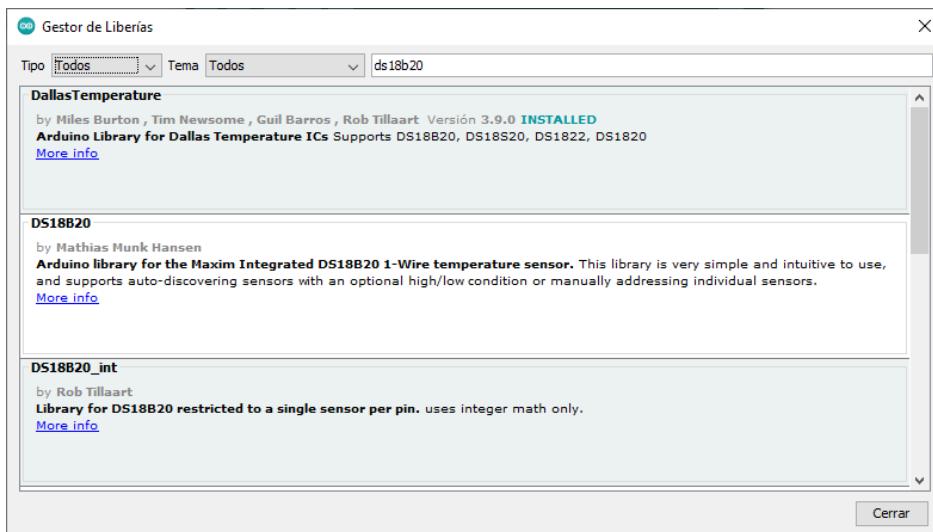
Presione el botón con el texto “instalar” y saldrá una ventana como se muestra a continuación:



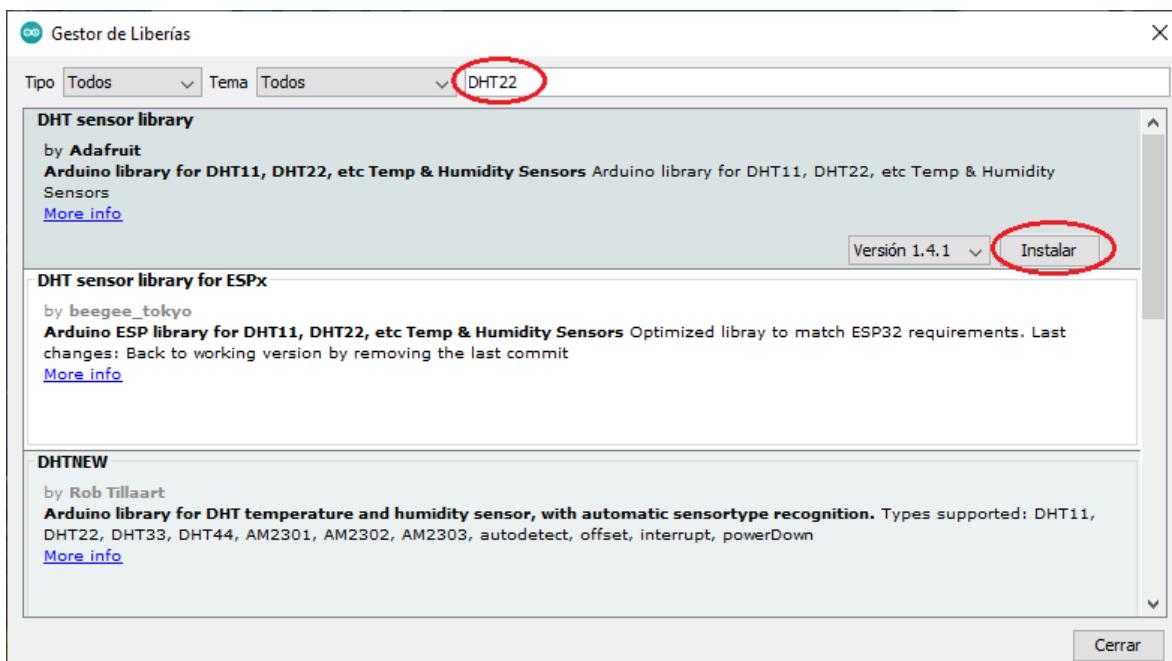
... presione el botón “Install all” y saldrá la siguiente ventana:



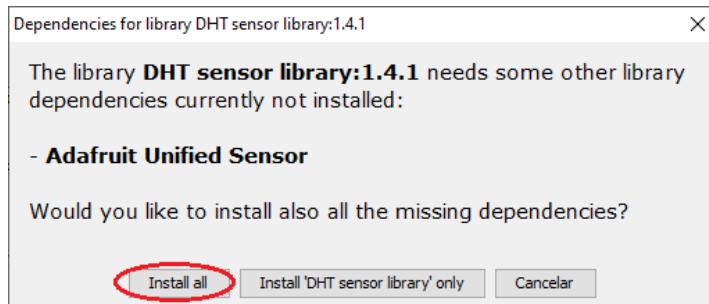
... espere a que la instalación termine y se mostrará una ventana como ésta:



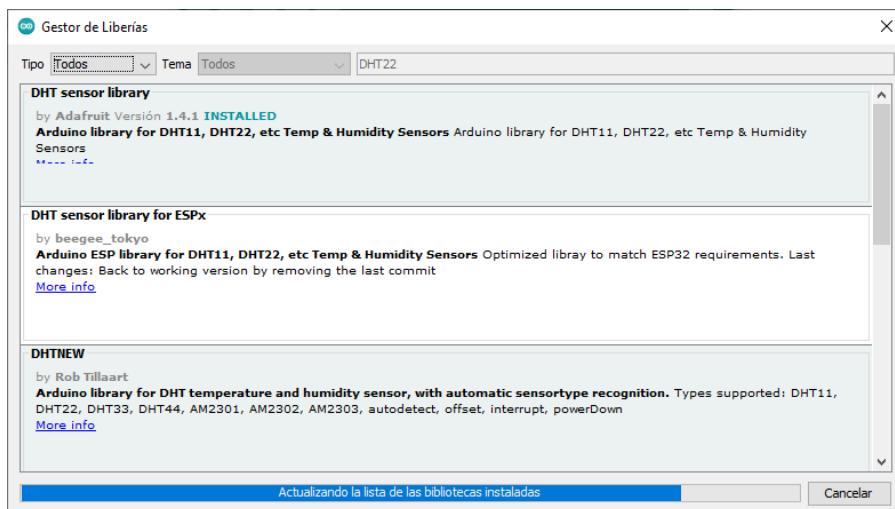
Es relevante mencionar que existe otro sensor, el cual, es ampliamente utilizado y de igual manera, se instalará su biblioteca correspondiente. El sensor mencionado anteriormente es el DHT22 (AM2301) y se inicia la instalación de la manera siguiente: borre lo escrito anteriormente y teclee DHT22 en el campo que se muestra a continuación:



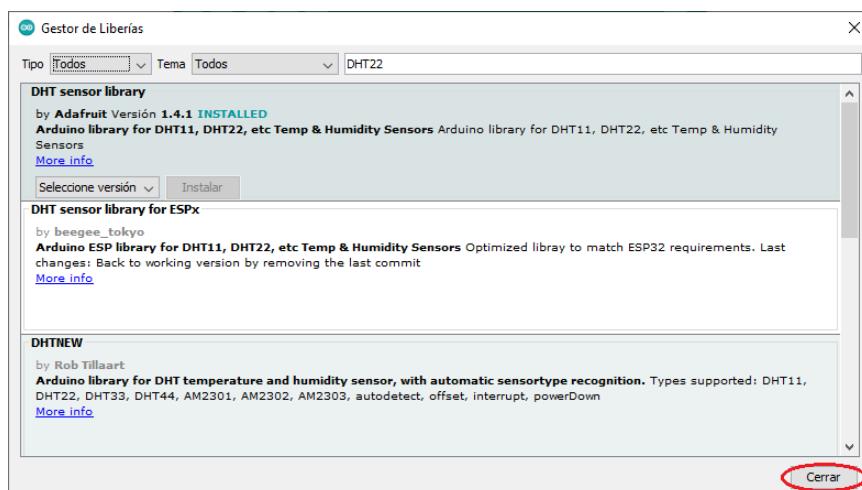
... e inmediatamente se mostrará en primer lugar la biblioteca del sensor DHT, seleccione la última versión y presione el botón "instalar" y a continuación le saldrá una ventana como se muestra:



... presione el botón “Install all” y espere a que la instalación concluya:



Cuando la barra azul se complete y se active el botón inferior derecho, notará que la leyenda del botón habrá cambiado a “cerrar”:



... presione el botón cerrar y se habrá concluido con la instalación de las bibliotecas.

## Modulo II

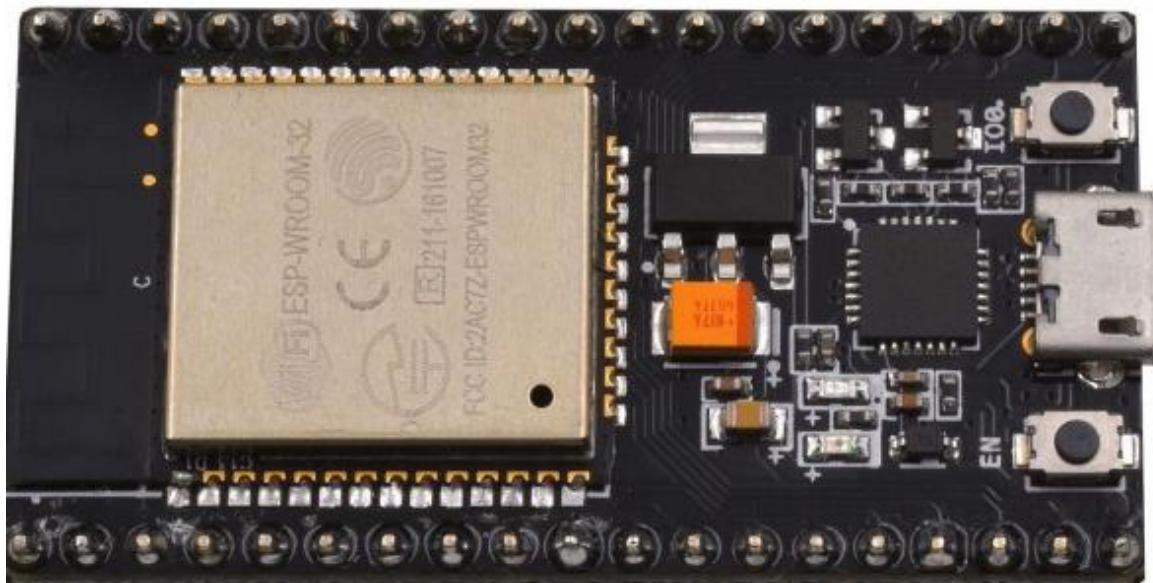
### 2. Implementando códigos en IDE de Arduino

**Objetivo General:** Conocerá la relación con el software del IDE del Arduino y el hardware del módulo NodeMCU-32S. Desarrollará e implementará el código de un medidor de temperatura.

## 2.1 El Hardware del módulo NodeMCU-32S

**Objetivo específico:** Conocer la distribución y funciones de los pines del módulo NodeMCU-32S.

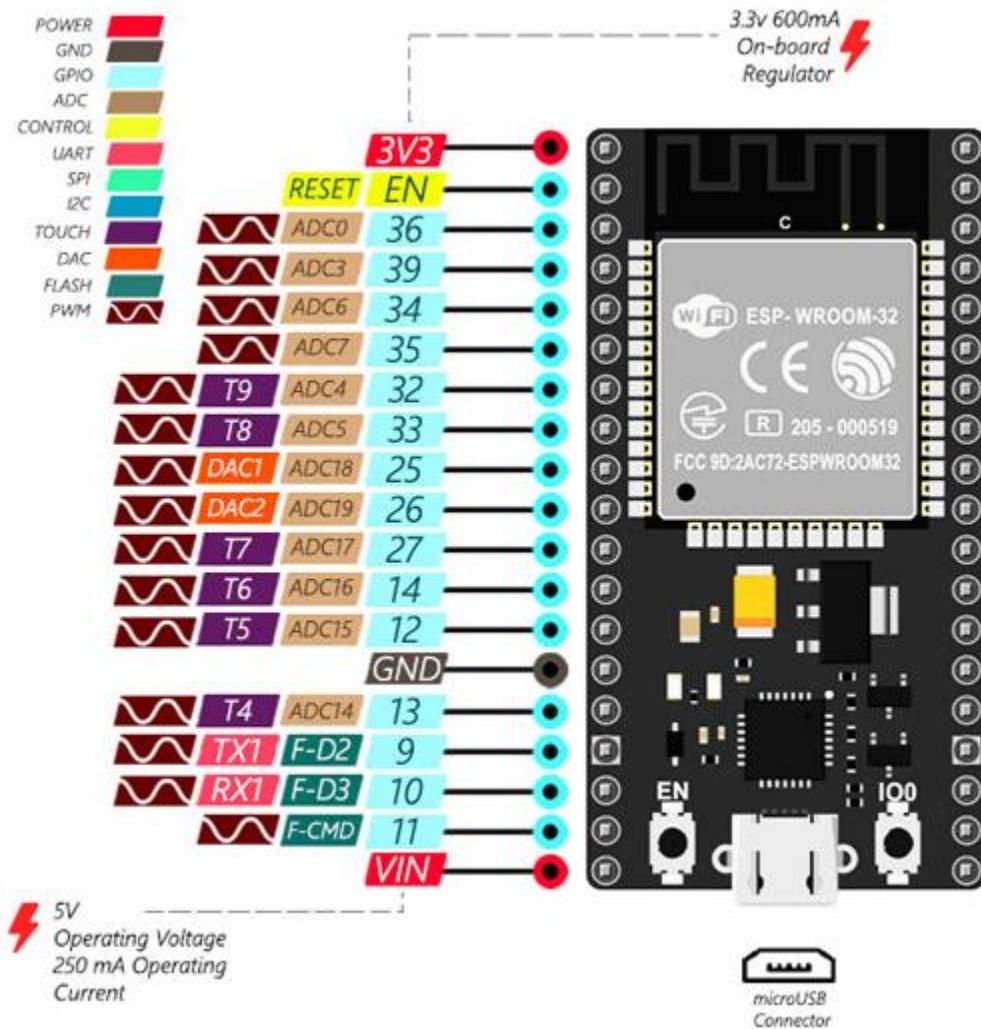
El módulo NodeMCU-32S tiene como núcleo un chip ESP32 y sólo requiere un voltaje de 3.3V. Dicho chip se compone de dos núcleos y pueden ser controlados individualmente. Se puede programar con el IDE del Arduino para la fácil implementación de dispositivos IoT. Adicionalmente, el módulo NodeMCU-32S contiene un chip CP2102, el cual, es un convertidor USB a puerto serie (RS232 TTL) fabricado por Silicon Labs y una entrada microUSB el cual, programar el módulo Huzzah y una comunicación con protocolo USB-RS232 hacia la computadora.



### Características:

- IEEE 802.11 b/g/n Wi-Fi 2.4Ghz
- Clock frequency adjustment range from 80Mhz to 240 Mhz
- Built-in 2-channel 12-bit high-precision ADC with up to 18 channels
- Soporta modos STA/AP/STA+AP
- Support UART/GPIO/ADC/DAC/SDIO/SD card/PWM/I2C/I2S interface
- Deep Sleep ultra bajo consumo < 5uA
- Memoria Flash 4 MB
- Memoria RAM para el usuario < 327KB

Pinout:

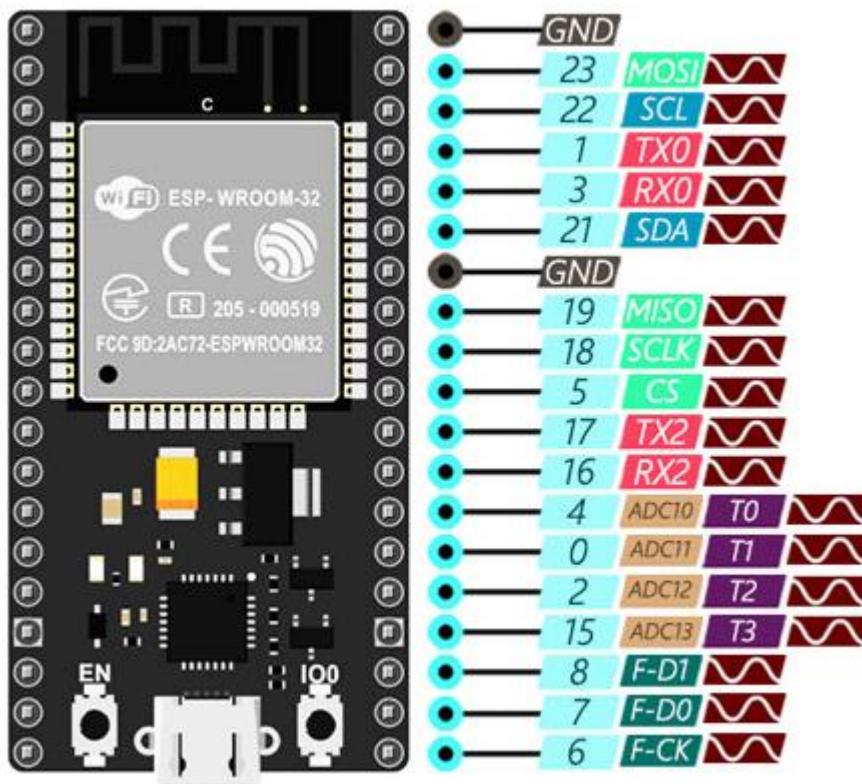


GPIO0 se encuentra conectado al pushbutton (ubicado en la esquina inferior derecha) marcado con el texto IO0.

GPIO2 se encuentra conectado al led verde para indicador para propósito general.

Los pines indicados con el cuadro color azul claro (light blue) son los que corresponden al número asignado en el IDE del Arduino.

Pinout:



Los pines indicados con el cuadro color azul claro ( ) son los que corresponden al número asignado en el IDE del Arduino.

Algunas definiciones hechas en el IDE del Arduino haciendo referencia a los pines del NodeMCU-32S para tomar en cuenta en la programación:

- LED\_BUILTIN corresponde al pin 2 (led verde en la tarjeta)
- A0 corresponde al pin 36 (en la figura se muestra los pines A's como ADC's, en el IDE del Arduino no reconoce el texto "ADC0")
- T0 corresponde al pin 4
- KEY\_BUILTIN corresponde al pushbutton marcado como IO0 (ver esquina inferior derecha)

Para más definiciones ver:

C:\Users\MiUsuario\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.4\variants\nodemcu-32s\pins\_Arduino.h

No olvidar que **MiUsuario** es el nombre de usuario que se le asignó a la computadora.

Los pines marcados en verde están perfectos para ser utilizados, los marcados en ámbar deben manejarse con cuidado ya que, pueden tener un comportamiento inesperado al iniciar y los marcados en rojo no son recomendable para su uso.

<b>GPIO</b>	<b>Input</b>	<b>Output</b>	<b>Notes</b>
<b>0</b>	PULLED UP	OK	Se conecta el <b>pushbutton</b> de la tarjeta Genera una señal PWM al iniciar Entrará en bootloader en estado LOW después del RESET
<b>1</b>	TX PIN	OK	Genera una señal serial al iniciar
<b>2</b>	OK	OK	Se conecta el <b>LED</b> de la tarjeta Requiere estar en FLOATING para bootloader
<b>3</b>	OK	RX PIN	Genera una señal HIGH al iniciar
<b>4</b>	OK	OK	
<b>5</b>	OK	OK	Debe de estar en estado HIGH al iniciar Genera una señal PWM al iniciar VSPI CS
<b>6</b>	X	X	Se encuentra conectado a la SPI flash interna
<b>7</b>	X	X	Se encuentra conectado a la SPI flash interna
<b>8</b>	X	X	Se encuentra conectado a la SPI flash interna
<b>9</b>	X	X	Se encuentra conectado a la SPI flash interna
<b>10</b>	X	X	Se encuentra conectado a la SPI flash interna
<b>11</b>	X	X	Se encuentra conectado a la SPI flash interna
<b>12</b>	OK	OK	Debe de estar en estado LOW al iniciar HSPI MISO
<b>13</b>	OK	OK	HSPI MOSI
<b>14</b>	OK	OK	Genera una señal PWM al iniciar HSPI CLK
<b>15</b>	OK	OK	Debe de estar en estado HIGH al iniciar Genera una señal PWM al iniciar HSPI CS
<b>16</b>	OK	OK	
<b>17</b>	OK	OK	
<b>18</b>	OK	OK	VSPI CLK

<b>19</b>	OK	OK	VSPI MISO
<b>21</b>	OK	OK	
<b>22</b>	OK	OK	
<b>23</b>	OK	OK	VSPI MOSI
<b>25</b>	OK	OK	
<b>26</b>	OK	OK	
<b>27</b>	OK	OK	
<b>32</b>	OK	OK	
<b>33</b>	OK	OK	
<b>34</b>	OK		Sólo INPUT
<b>35</b>	OK		Sólo INPUT
<b>36</b>	OK		Sólo INPUT
<b>39</b>	OK		Sólo INPUT

## 2.2 Análisis de ejemplos considerados relevantes para la implementación de un lector de temperatura.

### Consideración importante

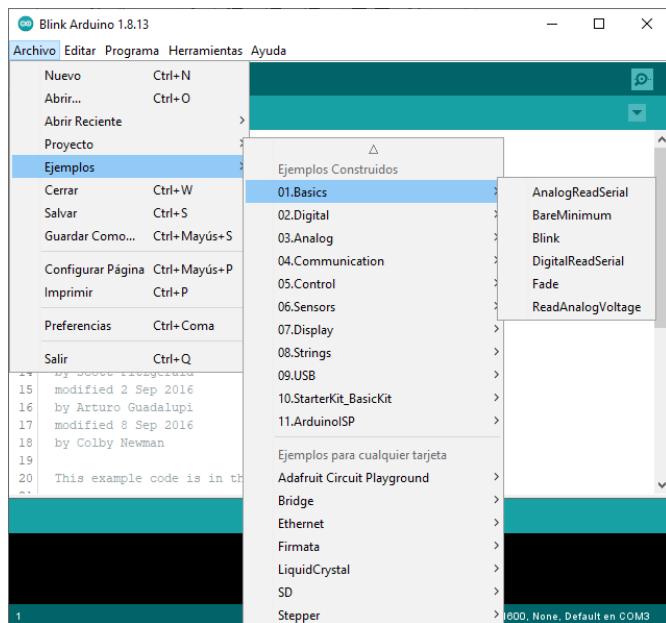
La estructura de programación del Arduino es muy sencilla, se debe considerar lo siguiente, el método `setup()` solamente se ejecuta una vez y sirve para configurar los periféricos:

```
void setup() {  
    // put your setup code here, to run once:  
  
}
```

El método `loop()` se ejecuta de manera infinita, es decir, cuando llegue a la última línea de código, el microcontrolador ejecutará nuevamente la primera:

```
void loop() {  
    // put your main code here, to run repeatedly:  
  
}
```

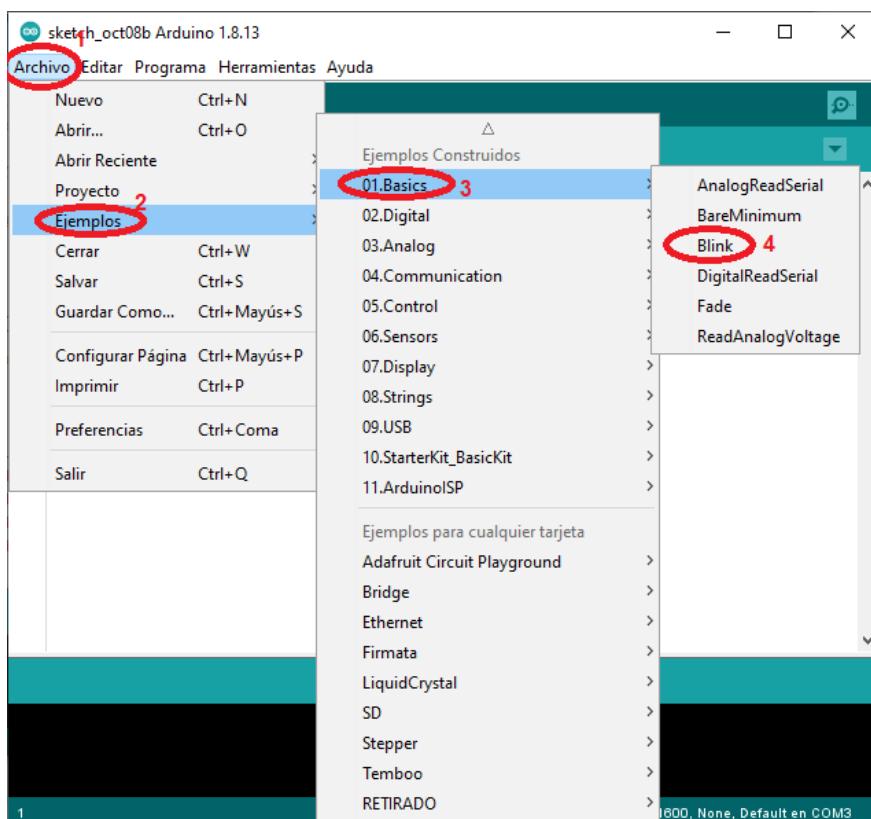
Todos los ejemplos que a continuación se mencionan se encuentran en la sección de ejemplos:



## 2.2.1 Blink.ino

**Objetivo específico:** Aprenderá a usar los pines de salida desde el IDE del Arduino.

Se inicia abriendo el ejemplo blink.ino:



Analizaremos el código de éste ejemplo:

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);
}
```

1. La primera y tercera línea son comentario (observe que están en inglés)
2. La segunda línea es el encabezado del método `setup()` que es el encargado de configurar nuestro periféricos y el microcontrolador SOLO lo ejecuta UNA vez.
3. La cuarta línea es una sentencia (comando ó instrucción) que se encarga de configurar los pines GPIO del microcontrolador.

4. La quinta línea le indica al compilador (convertidor de texto a lenguaje máquina) que ahí terminan las instrucciones del método setup().

Posteriormente analizamos la función loop()

```
// la función loop() se ejecuta una y otra vez para siempre
void loop() {
    digitalWrite(LED_BUILTIN, HIGH);      // enciende el LED (HIGH es el nivel de
                                         // voltaje)
    delay(1000);                      // espera un segundo
    digitalWrite(LED_BUILTIN, LOW);     // apaga el led haciendo el voltaje bajo
    delay(1000);                      // espera un segundo
}
```

Si usted recuerda, la definición “LED\_BUILTIN” (con letras azules), está hecha de la siguiente manera (OJO: No es visible desde el IDE del arduino):

```
int LED_BUILTIN = 2;
```

El programador puede cambiar por otra asignación, por ejemplo:

```
int LEDR = 2;
```

... y reescribir el código:

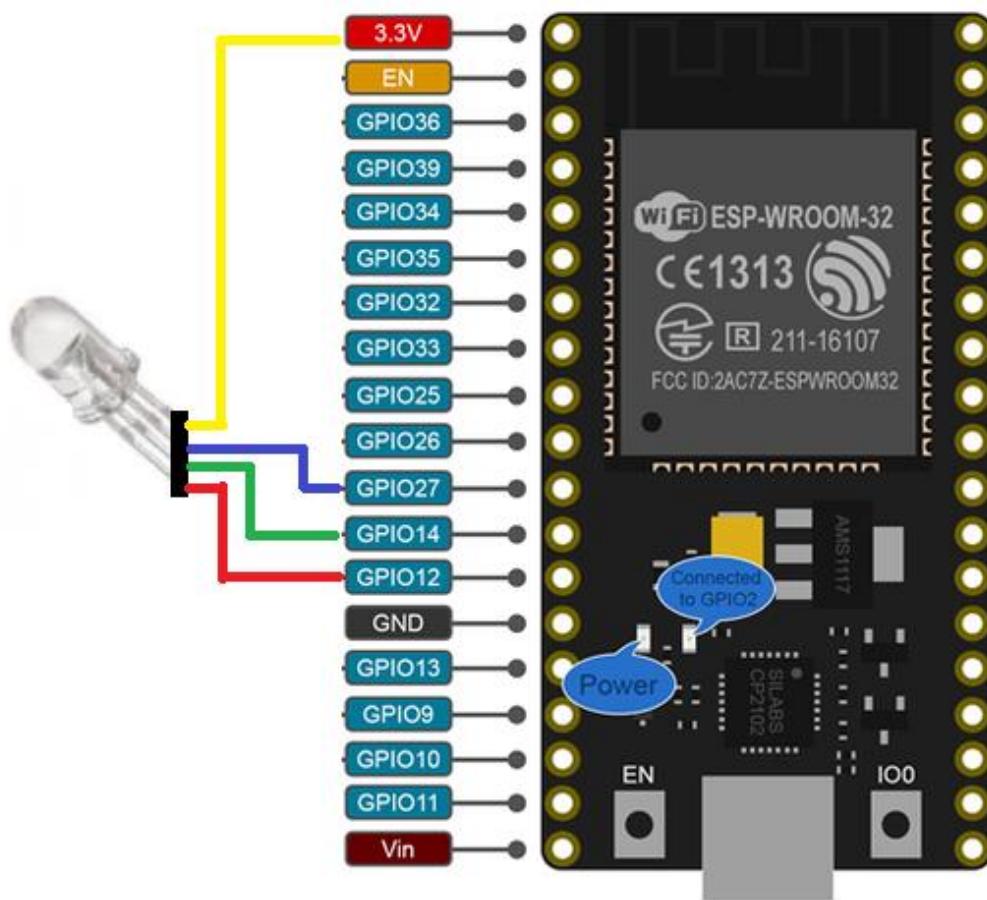
```
int LEDR = 2;
```

```
// la función setup() se ejecuta una vez al reiniciar o encender la placa
void setup() {
    // inicializa el pin digital llamado LED_BUILTIN como salida.
    pinMode(LEDR, OUTPUT);
}

// la función loop() se ejecuta una y otra vez para siempre
void loop() {
    digitalWrite(LEDR, LOW);    // enciende el LED (HIGH es el nivel de voltaje)
    delay(1000);              // espera un segundo
    digitalWrite(LEDR, HIGH);   // apaga el led haciendo el voltaje bajo
    delay(1000);              // espera un segundo
}
```

Nótese que el color de la nueva definición cambió a negro. Pero el código funciona exactamente igual al código anterior.

Ahora, con esa nueva definición... ¿Qué números le puedo asignar?, ¿Cuántos led puedo hacer parpadear al mismo tiempo?, Ahora conecte un led tricolor y haga las siguientes conexiones:



**IMPORTANTE: NO CONECTE EL LED DIRECTAMENTE AL MODULO**, utilice sólo el LED proporcionado en el curso, ya que contiene 3 resistencias limitadoras de corriente: LedR (180 ohms), LedG y LedB (100 ohms)

Cambie el valor de la variable por 12, compile y descargue su código a la tarjeta de desarrollo, posteriormente cámbiela por los valores 14 y 27. En cada caso observe los efectos:

```
int LEDR = 12;
```

... OJO, No olvide que:

1. Con el botón  se verifica e inicia la solicitud de conexión con el módulo para la descarga del código a la tarjeta de desarrollo.
  2. En algunos modelos de NodeMCU-32S no se requiere acción adicional para que el código empiece a descargarse, así que, no se mostrará la solicitud de conexión. El código se descargará automáticamente.

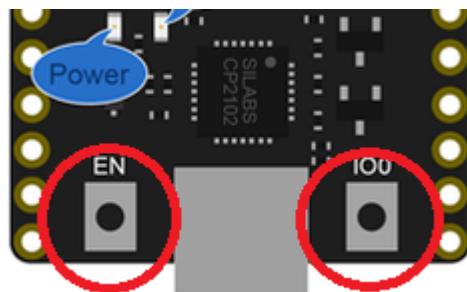
3. En otros modelos de hardware, mostrará una solicitud de conexión. En la parte inferior del IDE se mostrará la leyenda “conectando” así como se muestra en la siguiente figura:



... dependiendo del hardware adquirido, puede probar **UNA** de las acciones siguientes:

- Presione momentáneamente el botón IO0 y el código deberá empezar a descargarse a su módulo NodeMCU-32S.
- Presione y mantenga presionado el botón IO0, seguidamente presionar momentáneamente el botón de EN ó reset y por último liberar el botón IO0. Siguiendo éstos pasos el código deberá empezar a descargarse a su módulo NodeMCU-32S.

Ambos botones están ubicados a los costados del puerto microUSB de su módulo NodeMCU-32S y deberá empezar a descargar el código a su módulo NodeMCU32S.



4. Si la barra de notificación (intermedia entre la zona del código y la zona negra) del IDE cambia a naranja indica que ocurrió un error en la descarga, para corregirlo siga el procedimiento descrito al final del punto 1.3 “Errores en la descarga del código”.

¿Qué sucede si desea que parpadeen dos led's? Agregue al código anterior lo marcado en verde:

```
int LEDR = 12;  
int LEDG = 14;
```

```
// la función setup() se ejecuta una vez al reiniciar o encender la placa
void setup() {
    // inicializa el pin digital llamado LEDR como salida.
    pinMode(LEDR, OUTPUT);

    // inicializa el pin digital llamado LEDG como salida.
    pinMode(LEDG, OUTPUT);
}

// la función loop() se ejecuta una y otra vez para siempre
void loop() {
    digitalWrite(LEDR, LOW);      // enciende el LED (HIGH es el nivel de voltaje)
    delay(1000);                // espera un segundo
    digitalWrite(LEDR, HIGH);     // apaga el led haciendo el voltaje bajo
    delay(1000);                // espera un segundo

    // agregamos ésta sección de código para que parpadee ahora el led verde
    digitalWrite(LEDG, LOW);    // enciende el LED (HIGH es el nivel de voltaje)
    delay(1000);                // espera un segundo
    digitalWrite(LEDG, HIGH);   // apaga el led haciendo el voltaje bajo
    delay(1000);                // espera un segundo
}
```

¿y en el caso de tres led's? Agregue al código lo marcado en azul:

```
int LEDR = 12;
int LEDG = 14;
int LEDB = 27;

// la función setup() se ejecuta una vez al reiniciar o encender la placa
void setup() {
    // inicializa el pin digital llamado LEDR como salida.
    pinMode(LEDR, OUTPUT);

    // inicializa el pin digital llamado LEDG como salida.
    pinMode(LEDG, OUTPUT);

    // inicializa el pin digital llamado LEDB como salida,
    pinMode(LEDB, OUTPUT);

}

// la función loop() se ejecuta una y otra vez para siempre
void loop() {
    digitalWrite(LEDR, LOW);      // enciende el LED (HIGH es el nivel de voltaje)
    delay(1000);                // espera un segundo
    digitalWrite(LEDR, HIGH);     // apaga el led haciendo el voltaje bajo
    delay(1000);                // espera un segundo
```

```
// agregamos ésta sección de código para que parpadee ahora el led verde
digitalWrite(LEDG, LOW); // enciende el LED (HIGH es el nivel de voltaje)
delay(1000); // espera un segundo
digitalWrite(LEDG, HIGH); // apaga el led haciendo el voltaje bajo
delay(1000); // espera un segundo

// agregamos ésta sección de código para que parpadee ahora el led azul
digitalWrite(LEDB, LOW); // enciende el LED (HIGH es el nivel de voltaje)
delay(1000); // espera un segundo
digitalWrite(LEDB, HIGH); // apaga el led haciendo el voltaje bajo
delay(1000); // espera un segundo
}
```

### Actividades:

Ahora implemente un código que simule un semáforo real:

1. El led Verde encendido por 3 segundos, luego que parpadee cada medio segundo tres veces.
2. Luego el led Azul encienda durante 1 segundo.
3. Por último que el led Rojo se quede encendido 3 segundos.

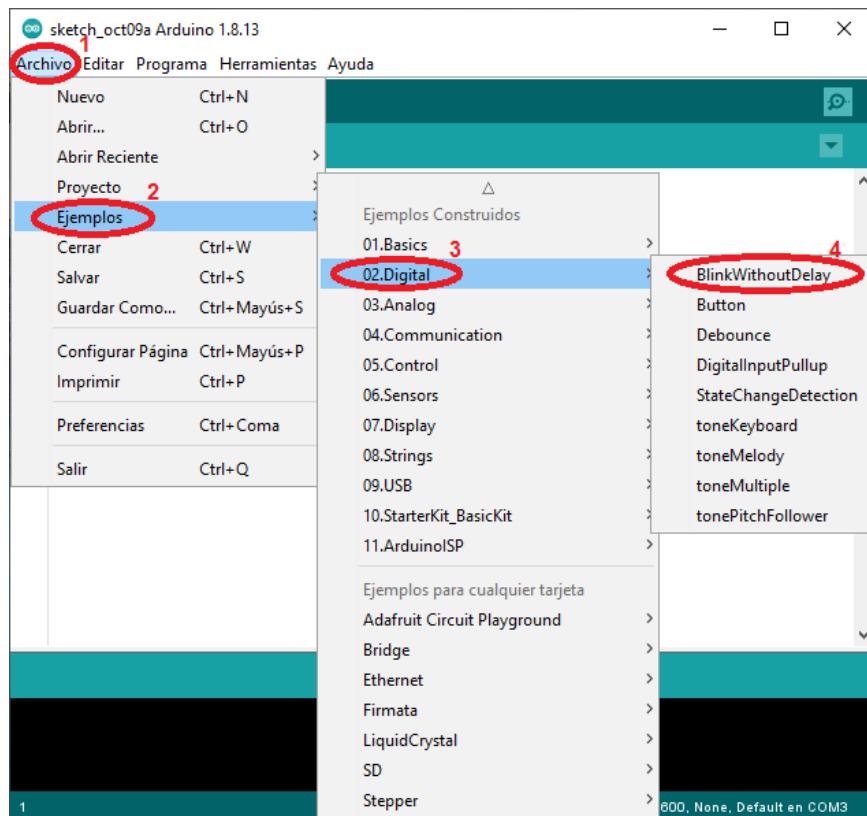
Considere que `delay(1000);` el valor de 1000 corresponde a 1 segundo.

## 2.2.2 BlinkWithOutDelay.ino

**Objetivo específico:** Aprenderá a programar prescindiendo de la sentencia `delay()`. Adicionalmente, conocerá la función `millis()`.

La sentencia `delay()` es muy útil cuando se desea que el procesador espere por algún tiempo, pero encadena al microcontrolador a realizar una sola función conocida como NOP (No Operation) y no se puede estar pendiente de otros eventos. Dicha instrucción (`delay`) NO puede usarse junto con el stack de WiFi ni con otras funciones que dependen de eventos programados.

Se analizará el siguiente ejemplo, cuya conexión del led está basado en el punto 2.2.1 de éste manual:



Compile y descargue el código a su módulo NodeMCU.

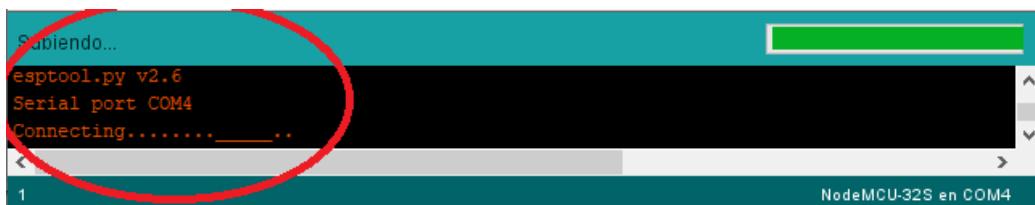
... **OJO**, No olvide que:



1. Con el botón  se verifica e inicia la solicitud de conexión con el módulo para la descarga del código a la tarjeta de desarrollo.

2. En algunos modelos de NodeMCU-32S no se requiere acción adicional para que el código empiece a descargarse, así que, no se mostrará la solicitud de conexión. El código se descargará automáticamente.

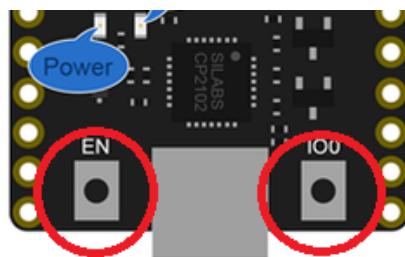
3. En otros modelos de hardware, mostrará una solicitud de conexión. En la parte inferior del IDE se mostrará la leyenda “conectando” así como se muestra en la siguiente figura:



... dependiendo del hardware adquirido, puede probar **UNA** de las acciones siguientes:

- Presione momentáneamente el botón IO0 y el código deberá empezar a descargarse a su módulo NodeMCU-32S.
- Presione y mantenga presionado el botón IO0, seguidamente presionar momentáneamente el botón de EN ó reset y por último liberar el botón IO0. Siguiendo éstos pasos el código deberá empezar a descargarse a su módulo NodeMCU-32S.

Ambos botones están ubicados a los costados del puerto microUSB de su módulo NodeMCU-32S y deberá empezar a descargar el código a su módulo NodeMCU32S.



4. Si la barra de notificación (intermedia entre la zona del código y la zona negra) del IDE cambia a naranja indica que ocurrió un error en la descarga, para corregirlo siga el procedimiento descrito al final del punto 1.3 “Errores en la descarga del código”.

... y ahora analicemos el siguiente ejemplo:

```
// Las constantes no pueden cambiarse. Usado para asignar el número de pin:  
const int ledPin = LED_BUILTIN;// el número del pin donde está el LED
```

```
// Las variables cambiarán:  
int ledState = HIGH; // El estado del led es ajustado  
  
// Generalmente, debería usar el tipo "unsigned long" para variables que guardan  
el tiempo  
// El valor de esta variable rápidamente crecerá convirtiéndose muy grande para  
ser almacenado en un tipo "int"  
unsigned long previousMillis = 0; // Almacenará el último valor de tiempo  
en que el led fue actualizado de estado  
  
// constantes no cambiarán:  
const long interval = 1000; // lapso de tiempo en la cual cambiará de  
estado (valor en milisegundos)  
  
void setup() {  
    // ajusta el pin como salida:  
    pinMode(ledPin, OUTPUT);  
}  
  
void loop() {  
    // Aquí es donde Usted pone el código que necesita estar corriendo todo el  
tiempo.  
  
    // Checa si es tiempo de cambiar el estado del LED, esto es, si la  
    // diferencia entre el tiempo actual y el último tiempo en que el led  
    // cambió de estado es más grande que el intervalo que usted ajustó  
    // previamente, cambiará de estado el LED.  
    unsigned long currentMillis = millis(); // Se lee el tiempo actual y se  
asigna en una variable del tipo entero de 32 bits  
  
    // Mediante una operación aritmética, se calcula la diferencia del tiempo  
transcurrido (currentMillis - previousMillis) y dicho resultado, se compara con  
el tiempo establecido arbitrariamente en la variable "interval". Si la  
diferencia del tiempo calculado es mayor que la establecida en la variable  
"interval" se ejecutarán las sentencias que se ubican entre los corchetes.  
    if (currentMillis - previousMillis >= interval) {  
        // Se almacena el último valor de tiempo en que el LED cambió de estado  
        previousMillis = currentMillis;  
  
        // si el LED está apagado se encenderá y viceversa:  
        if (ledState == HIGH) {  
            ledState = LOW;  
        } else {  
            ledState = HIGH;  
        }  
  
        // ajusta el estado con el valor almacenado en la variable ledState:  
        digitalWrite(ledPin, ledState);  
    }  
}
```

## Actividades:

1. Cambie el valor de la variable por 12, compile y descargue su código a la tarjeta de desarrollo, posteriormente cámbiela por los valores 14 y 27. En cada caso observe los efectos:

```
const int ledPin = 12;
```

2. Ahora, cambie el tiempo almacenado en la variable, pruebe con valores 500, 250 y 100. En cada caso observe los efectos:

```
const long interval = 500;
```

### 2.2.3 Máquina de estados

**Objetivo específico:** Implementará una máquina de estados usando el módulo ESP32.

Este ejemplo enseña a usar las máquinas de estado, los cuales son indispensable para el uso del stack del WiFi. Observe en el ejemplo que no se utiliza la sentencia `delay()`, adicionalmente, se observa que el valor de **una variable controla una secuencia**, o sea, un estado. De ahí proviene el nombre de **máquinas de estado**, ya que el valor de la variable `estadoLed` determina que color de LED enciende.

```
int estadoLed = 0;  
  
estadoLed++; // Esta variable controla los estados y cada vez que se ejecuta ésta  
// sentencia la variable estadoLed se incrementa en uno  
// (estadoLed = estadoLed + 1;)
```

La manera óptima de manejar las secuencias de la máquina de estados es el uso de la secuencia `switch` y `case`:

```
switch (estadoLed) {  
    case 0: // todos los leds apagados  
        // Secuencia a ejecutar cuando estadoled = 0  
        break;  
    case 1: // sólo el led rojo se enciende  
        // Secuencia a ejecutar cuando estadoled = 1  
        break;  
    case 2: // sólo el led verde se enciende  
        // Secuencia a ejecutar cuando estadoled = 2  
        break;  
    case 3: // sólo el led azul se enciende  
        // Secuencia a ejecutar cuando estadoled = 3  
        break;  
    default:  
        // Secuencia a ejecutar cuando estadoled tiene valor diferente a los anteriores  
        break;  
}
```

Para realizar éste ejemplo se usaron los demos: Digital => BlinkWithOutDelay, Control => IfStatementConditional y Control => SwitchCase.

Partiendo de este ejemplo, desarrolle el ejemplo del semáforo visto anteriormente.

```
// Inicio del código de la máquina de estados
// constants won't change. Used here to set a pin number:
// Conecte los leds como sigue:
const int ledR = 12;// the number of the LED pin (rojo)
const int ledG = 14;// the number of the LED pin (verde)
const int ledB = 27;// the number of the LED pin (azul)

// Generally, you should use "unsigned long" for variables that hold time
// The value will quickly become too large for an int to store
unsigned long previousMillis = 0;           // will store last time LED was updated

// constants won't change:
const long interval = 1000;                  // interval at which to blink (milliseconds)
int estadoLed = 0;

void setup() {
    // set the digital pin as output:
    pinMode(ledR, OUTPUT);
    pinMode(ledG, OUTPUT);
    pinMode(ledB, OUTPUT);
    digitalWrite(ledR, HIGH);
    digitalWrite(ledG, HIGH);
    digitalWrite(ledB, HIGH);
}

void loop() {
    // here is where you'd put code that needs to be running all the time.

    // check to see if it's time to blink the LED; that is, if the difference
    // between the current time and last time you blinked the LED is bigger than
    // the interval at which you want to blink the LED.
    unsigned long currentMillis = millis();

    if (currentMillis - previousMillis >= interval) {
        // save the last time you blinked the LED
        previousMillis = currentMillis;
        estadoLed++; // Esta variable controla los estados y cada vez que se ejecuta ésta
                      // sentencia la variable estadoLed se incrementa en uno
                      // (estadoLed = estadoLed + 1)
        if (estadoLed > 3) // Los valores que puede tomar la variable
            estadoLed = 1; // estadoLed son: 1, 2 y 3. (Se descarta el valor 0)

        // do something different depending on the range value:
        switch (estadoLed) {
            case 0:   // todos los leds apagados
                digitalWrite(ledR, HIGH);
                digitalWrite(ledG, HIGH);
                digitalWrite(ledB, HIGH);
                break;
            case 1:   // sólo el led rojo se enciende
                digitalWrite(ledR, LOW);
                digitalWrite(ledG, HIGH);
                digitalWrite(ledB, HIGH);
                break;
            case 2:   // sólo el led verde se enciende
                digitalWrite(ledR, HIGH);
                digitalWrite(ledG, LOW);
                digitalWrite(ledB, HIGH);
                break;
            case 3:   // sólo el led azul se enciende
                digitalWrite(ledR, HIGH);
                digitalWrite(ledG, HIGH);
                break;
        }
    }
}
```

```
    digitalWrite(ledB, LOW);
    break;
default: // Sólo se ejecuta cuando 0 > estadoLed > 3
    digitalWrite(ledR, HIGH);
    digitalWrite(ledG, HIGH);
    digitalWrite(ledB, HIGH);
    estadoLed = 0;
    break;
}

}
} //Fin del código de la maquina de estados
```

El ejemplo está disponible en: <https://github.com/gpoolb/ESP32> en la carpeta “MaquinaDeEstados” y está basado en la conexión del LED que se describe en el punto 2.2.1 de éste manual.

Descargue el ejemplo del link, compile y descargue el código a su módulo NodeMCU.

... **OJO**, No olvide que:



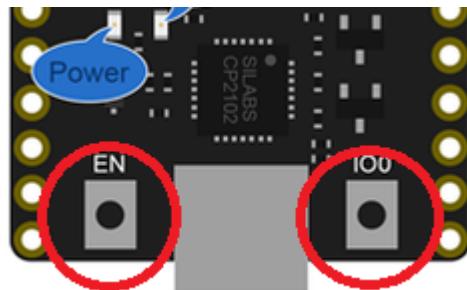
1. Con el botón  se verifica e inicia la solicitud de conexión con el módulo para la descarga del código a la tarjeta de desarrollo.
2. En algunos modelos de NodeMCU-32S no se requiere acción adicional para que el código empiece a descargarse, así que, no se mostrará la solicitud de conexión. El código se descargaría automáticamente.
3. En otros modelos de hardware, mostrará una solicitud de conexión. En la parte inferior del IDE se mostrará la leyenda “conectando” así como se muestra en la siguiente figura:



... dependiendo del hardware adquirido, puede probar **UNA** de las acciones siguientes:

- a) Presione momentáneamente el botón IO0 y el código deberá empezar a descargarse a su módulo NodeMCU-32S.
- b) Presione y mantenga presionado el botón IO0, seguidamente presionar momentáneamente el botón de EN ó reset y por último liberar el botón IO0. Siguiendo éstos pasos el código deberá empezar a descargarse a su módulo NodeMCU-32S.

Ambos botones están ubicados a los costados del puerto microUSB de su módulo NodeMCU-32S y deberá empezar a descargar el código a su módulo NodeMCU32S.



4. Si la barra de notificación (intermedia entre la zona del código y la zona negra) del IDE cambia a naranja indica que ocurrió un error en la descarga, para corregirlo siga el procedimiento descrito al final del punto 1.3 “Errores en la descarga del código”.

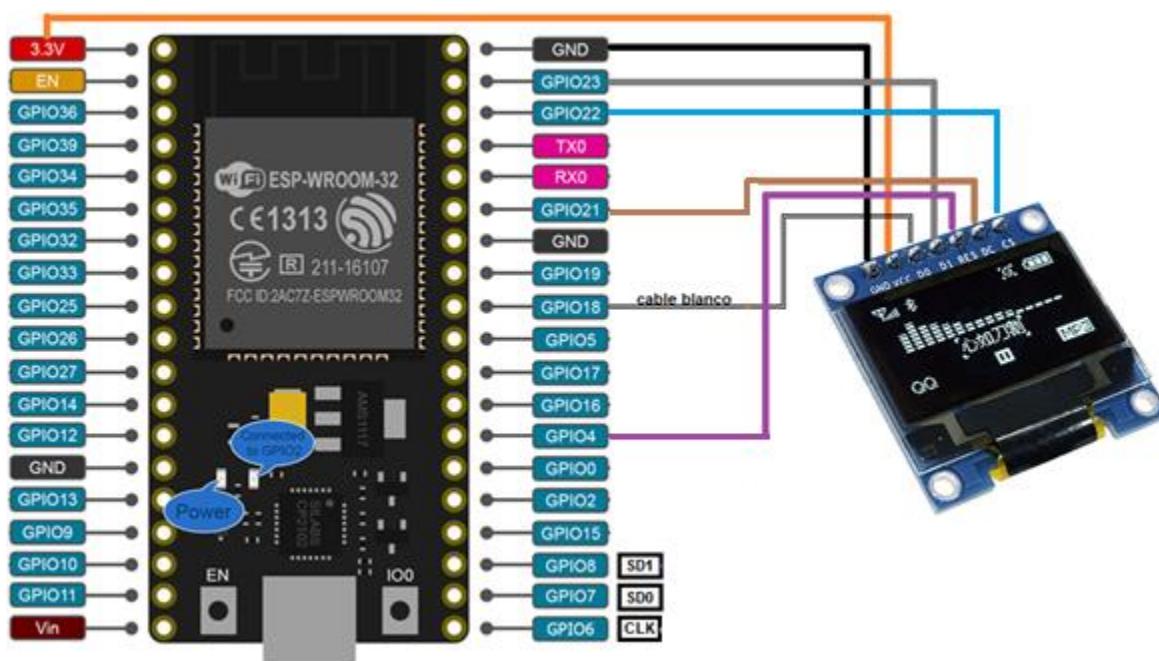
#### Actividades:

¿podría agregar los estados requeridos para que el led verde parpadee dos veces antes de cambiar al led azul?

## 2.2.4 Usando la pantalla OLED SSD1306

**Objetivo específico:** Ejecutará el ejemplo que permite verificar el funcionamiento de la pantalla OLED SSD1306 usando el módulo ESP32.

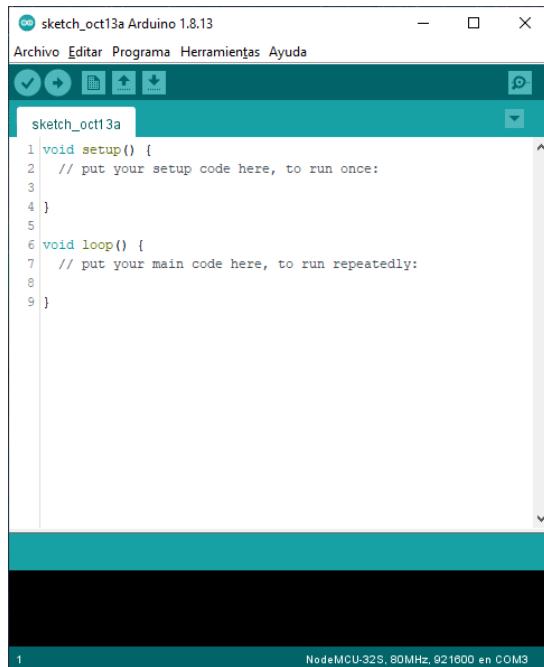
Para iniciar con este punto, se requiere a ver unas conexiones a la pantalla OLED y al módulo NodeMCU-32S así como se muestra en la pantalla:



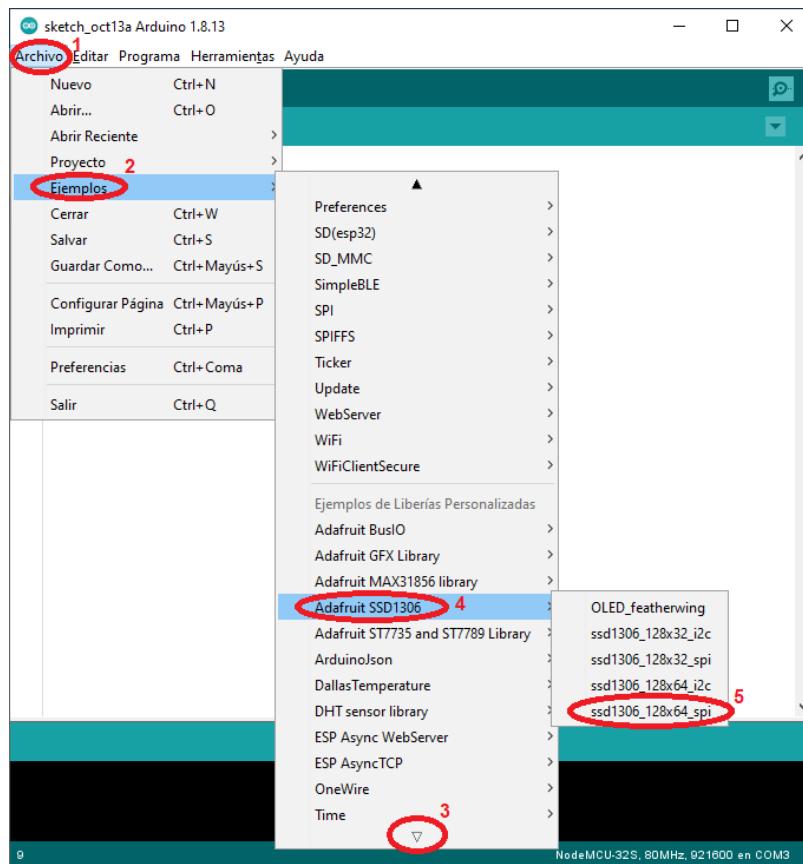
**IMPORTANTE:** Si el conexionado de la pantalla no se hace correctamente, puede dañarse irremediablemente. **Las conexiones que más debe cuidar son las de GND y VCC** (las dos primeras contando de izquierda a derecha) que corresponden a los cables **negro y naranja**.

Adicionalmente, tiene que considerar que la instalación de las bibliotecas (descritas en el punto 1.6 de este manual) ha sido completadas con éxito.

Al terminar las conexiones, se requiere abrir el IDE del Arduino como sigue:



Proceda a abrir el ejemplo siguiente la siguiente secuencia:



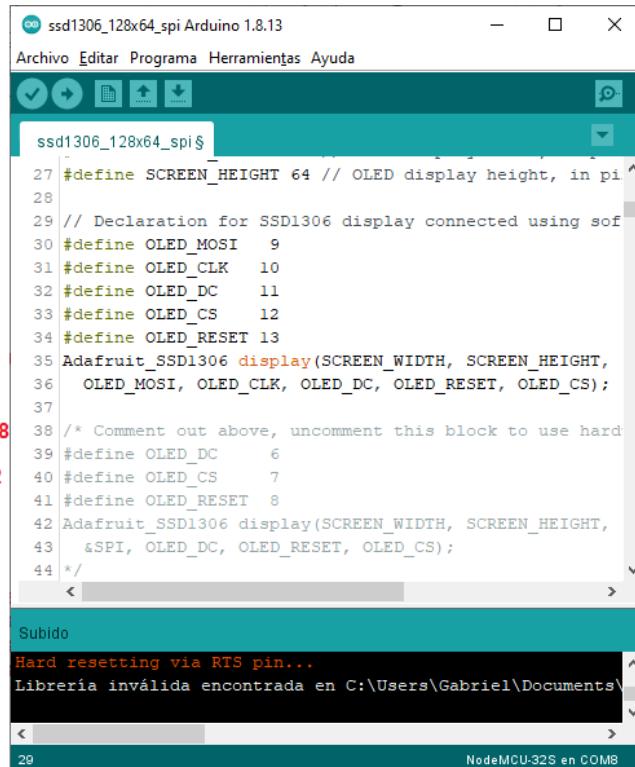
Antes de ejecutar el ejemplo, se requiere hacer unas modificaciones al código, así como se muestra a continuación:

1. Agregue /\* en la línea 28

2. Agregue \*/ en la línea 37

3. Elimine el /\* y cambiela por // en la linea 38
4. Cambie el valor de ésta definición por 21
5. Cambie el valor de ésta definición por 22
6. Cambie el valor de ésta definición por 4

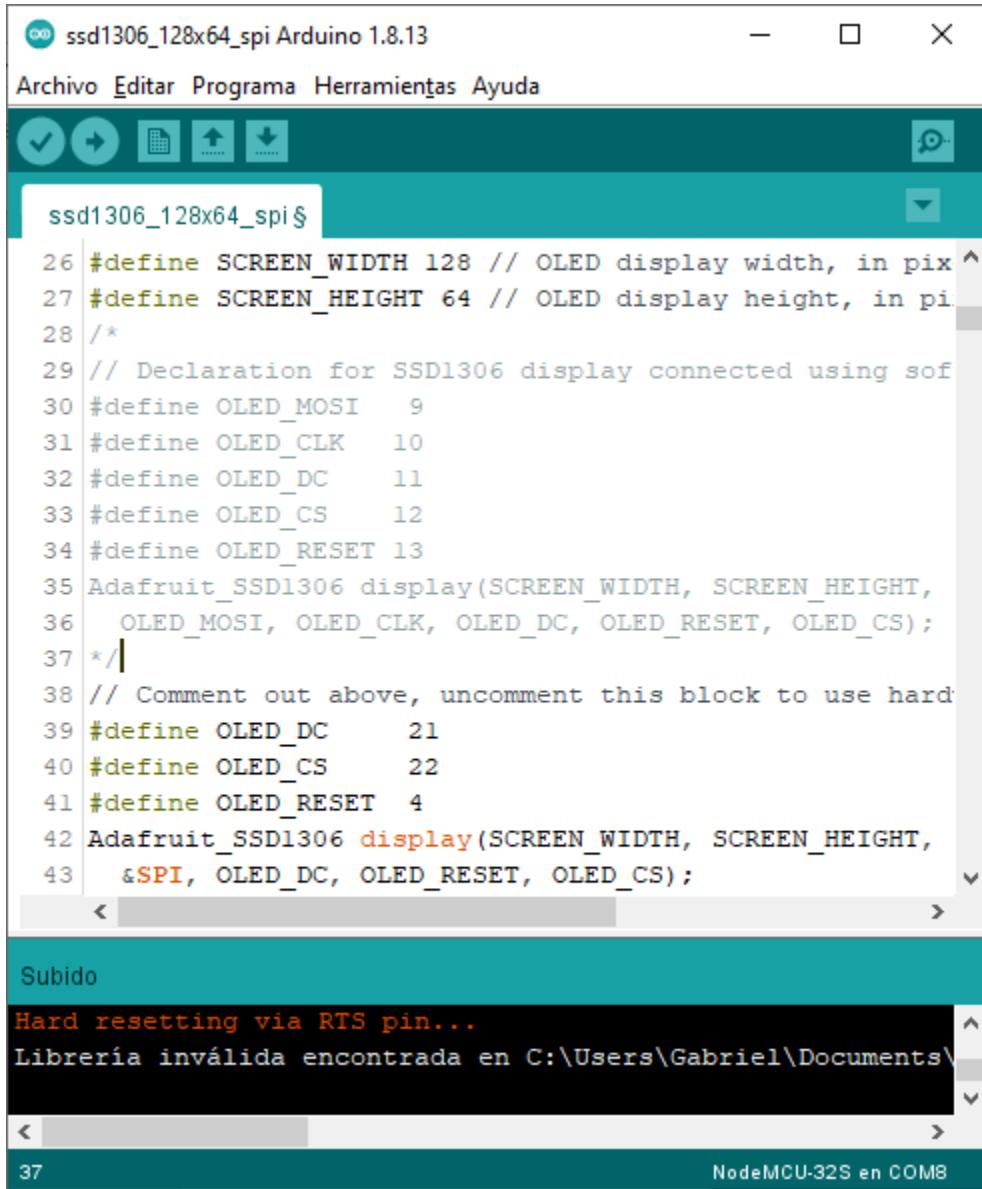
7. Elimine el \*/ de la línea 44



```
ssd1306_128x64_spi Arduino 1.8.13
Archivo Editar Programa Herramientas Ayuda
ssd1306_128x64_spi$ 
27 #define SCREEN_HEIGHT 64 // OLED display height, in pixels
28 // Declaration for SSD1306 display connected using software SPI
29 #define OLED_MOSI 9
30 #define OLED_CLK 10
31 #define OLED_DC 11
32 #define OLED_CS 12
33 #define OLED_RESET 13
34 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
35 OLED_MOSI, OLED_CLK, OLED_DC, OLED_RESET, OLED_CS);
36
37 /* Comment out above, uncomment this block to use hardware SPI
38 #define OLED_DC 6
39 #define OLED_CS 7
40 #define OLED_RESET 8
41 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
42 &SPI, OLED_DC, OLED_RESET, OLED_CS);
43 */
44 */

Subido
Hard resetting via RTS pin...
Librería inválida encontrada en C:\Users\Gabriel\Documents\Arduino\libraries\Adafruit_SSD1306\SSD1306.h:1:2: fatal error: 'Adafruit_GFX.h' file not found
#include "Adafruit_GFX.h"
          ^~~~~~
29
NodeMCU-32S en COM8
```

El código debe quedar, como se muestra a continuación:



```
ssd1306_128x64_spi Arduino 1.8.13
Archivo Editar Programa Herramientas Ayuda
ssd1306_128x64_spi$
```

```
26 #define SCREEN_WIDTH 128 // OLED display width, in pixels
27 #define SCREEN_HEIGHT 64 // OLED display height, in pixels
28 /*
29 // Declaration for SSD1306 display connected using software SPI
30 #define OLED_MOSI 9
31 #define OLED_CLK 10
32 #define OLED_DC 11
33 #define OLED_CS 12
34 #define OLED_RESET 13
35 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
36   OLED_MOSI, OLED_CLK, OLED_DC, OLED_RESET, OLED_CS);
37 */
38 // Comment out above, uncomment this block to use hardware SPI
39 #define OLED_DC 21
40 #define OLED_CS 22
41 #define OLED_RESET 4
42 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
43   &SPI, OLED_DC, OLED_RESET, OLED_CS);
```

Subido

Hard resetting via RTS pin...

Librería inválida encontrada en C:\Users\Gabriel\Documents\

37 NodeMCU-32S en COM8

Compile y descargue a su tarjeta de desarrollo.

... OJO, No olvide que:

1. Con el botón  se verifica e inicia la solicitud de conexión con el módulo para la descarga del código a la tarjeta de desarrollo.
2. En algunos modelos de NodeMCU-32S no se requiere acción adicional para que el código empiece a descargarse, así que, no se mostrará la solicitud de conexión. El código se descargará automáticamente.
3. En otros modelos de hardware, mostrará una solicitud de conexión. En la parte

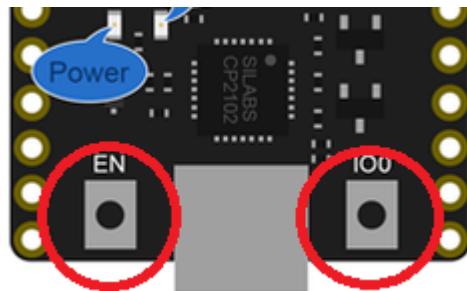
inferior del IDE se mostrará la leyenda “conectando” así como se muestra en la siguiente figura:



... dependiendo del hardware adquirido, puede probar **UNA** de las acciones siguientes:

- Presione momentáneamente el botón IO0 y el código deberá empezar a descargarse a su módulo NodeMCU-32S.
- Presione y mantenga presionado el botón IO0, seguidamente presionar momentáneamente el botón de EN ó reset y por último liberar el botón IO0. Siguiendo éstos pasos el código deberá empezar a descargarse a su módulo NodeMCU-32S.

Ambos botones están ubicados a los costados del puerto microUSB de su módulo NodeMCU-32S y deberá empezar a descargar el código a su módulo NodeMCU32S.



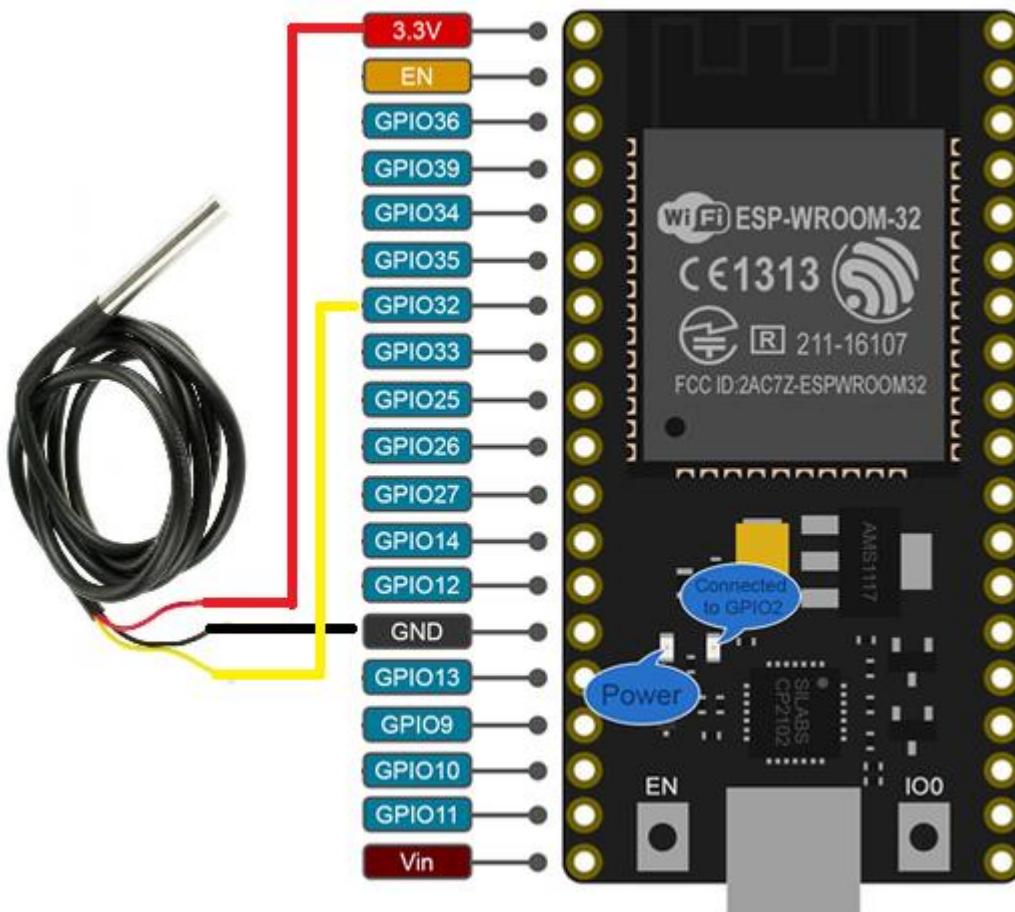
- Si la barra de notificación (intermedia entre la zona del código y la zona negra) del IDE cambia a naranja indica que ocurrió un error en la descarga, para corregirlo siga el procedimiento descrito al final del punto 1.3 “Errores en la descarga del código”.

¿Qué observa en la pantalla?

## 2.2.5 Usando el sensor de temperatura DS18B20 de fabricado por dallas semiconductor

**Objetivo específico:** Ejecutará el ejemplo que permite verificar el funcionamiento del sensor de temperatura usando el módulo ESP32.

Antes de iniciar, se requiere del conexionado del sensor como se muestra en la figura:



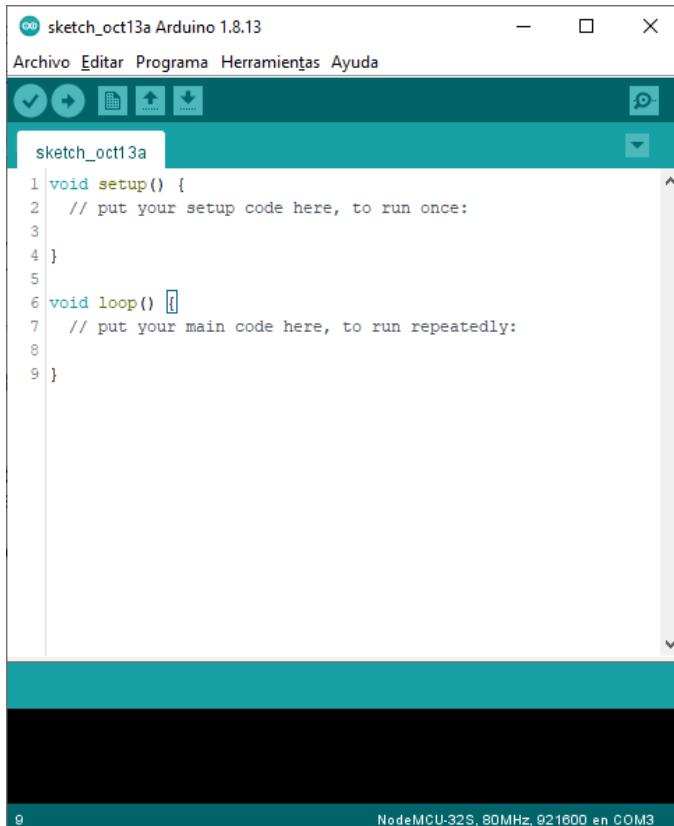
### IMPORTANTE:

**USE EL SENSOR PROPORCIONADO EN EL CURSO,** ya que contiene una resistencia de 4.7 Kohms conectado entre los cables rojo y amarillo, los cuales son necesarios para su funcionamiento.

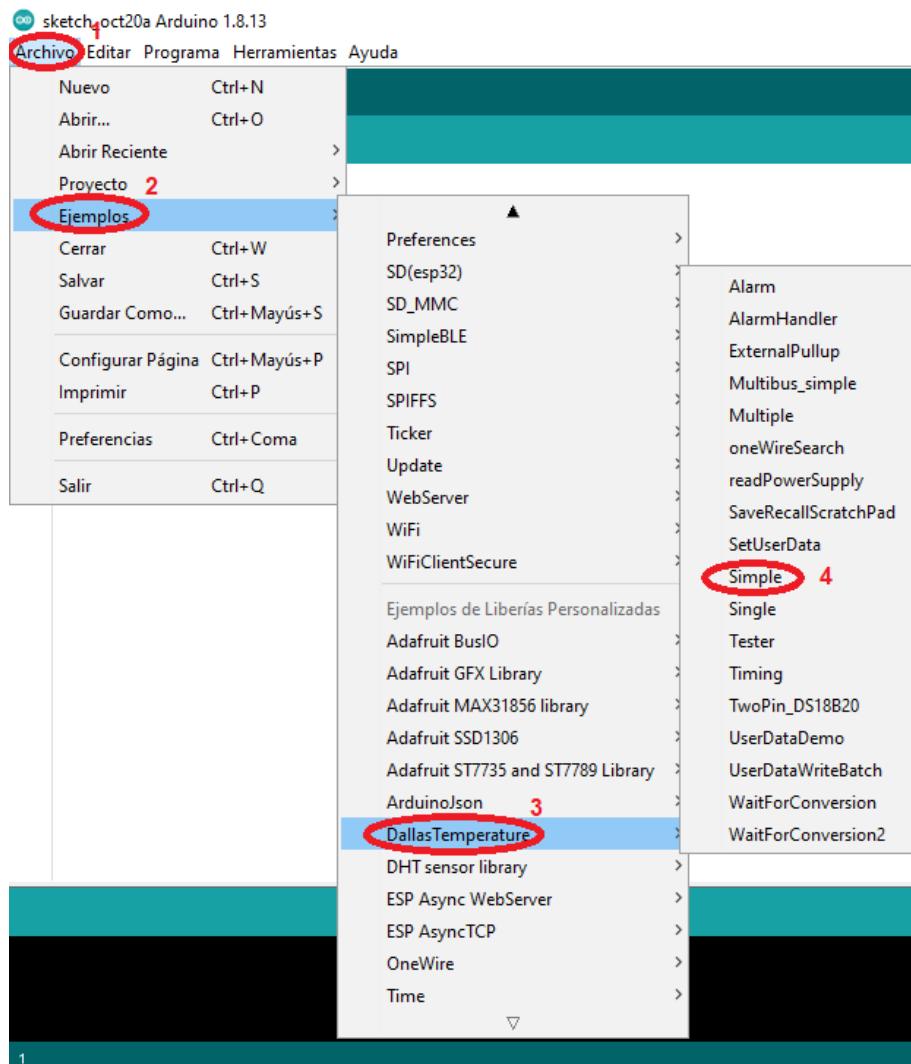
**El dispositivo PUEDE DAÑARSE si es conectado de manera incorrecta.** Los pines críticos son el GND y 3.3V (cable Negro y rojo)

Adicionalmente, tiene que considerar que la instalación de las bibliotecas (descritas en el punto 1.6 de este manual) ha sido completadas con éxito.

Seguidamente, se requiere abrir el IDE del Arduino como sigue:



Ahora siga la secuencia marcada como se muestra a continuación:



Ubique la línea 6 del código:

```

4
5 // Data wire is plugged into port 2 on the Arduino
6 #define ONE_WIRE_BUS 2
7

```

... y cámbiela como sigue:

```
#define ONE_WIRE_BUS 32
```

Compile y descargue su código a la tarjeta de desarrollo.

**... OJO,** No olvide que:

1. Con el botón  se verifica e inicia la solicitud de conexión con el módulo para

la descarga del código a la tarjeta de desarrollo.

2. En algunos modelos de NodeMCU-32S no se requiere acción adicional para que el código empiece a descargarse, así que, no se mostrará la solicitud de conexión. El código se descargará automáticamente.

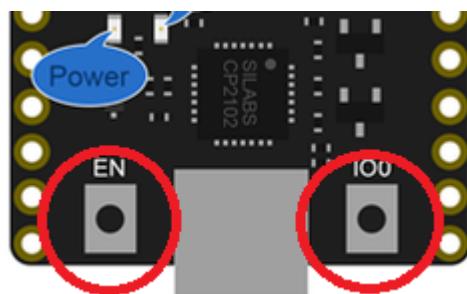
3. En otros modelos de hardware, mostrará una solicitud de conexión. En la parte inferior del IDE se mostrará la leyenda “conectando” así como se muestra en la siguiente figura:



... dependiendo del hardware adquirido, puede probar **UNA** de las acciones siguientes:

- Presione momentáneamente el botón IO0 y el código deberá empezar a descargarse a su módulo NodeMCU-32S.
- Presione y mantenga presionado el botón IO0, seguidamente presionar momentáneamente el botón de EN ó reset y por último liberar el botón IO0. Siguiendo éstos pasos el código deberá empezar a descargarse a su módulo NodeMCU-32S.

Ambos botones están ubicados a los costados del puerto microUSB de su módulo NodeMCU-32S y deberá empezar a descargar el código a su módulo NodeMCU32S.



4. Si la barra de notificación (intermedia entre la zona del código y la zona negra) del IDE cambia a naranja indica que ocurrió un error en la descarga, para corregirlo siga el procedimiento descrito al final del punto 1.3 “Errores en la descarga del código”.

Una vez descargado en su tarjeta de desarrollo se requiere abrir el monitor serial y ajustar el baud rate tal como se describe en el punto 1.5 de este manual:

Observe y memorice el valor señalado a continuación:

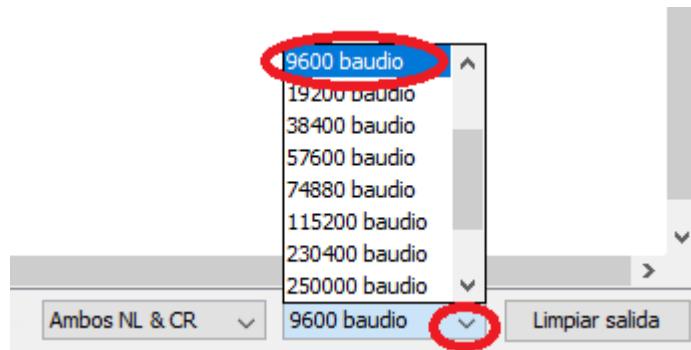
```
17 void setup(void)
18 {
19     // start serial port
20     Serial.begin(9600);
```



Abra el monitor del puerto serie presionando el botón marcado en la siguiente figura:



Ajuste los baudios como se indica a continuación:



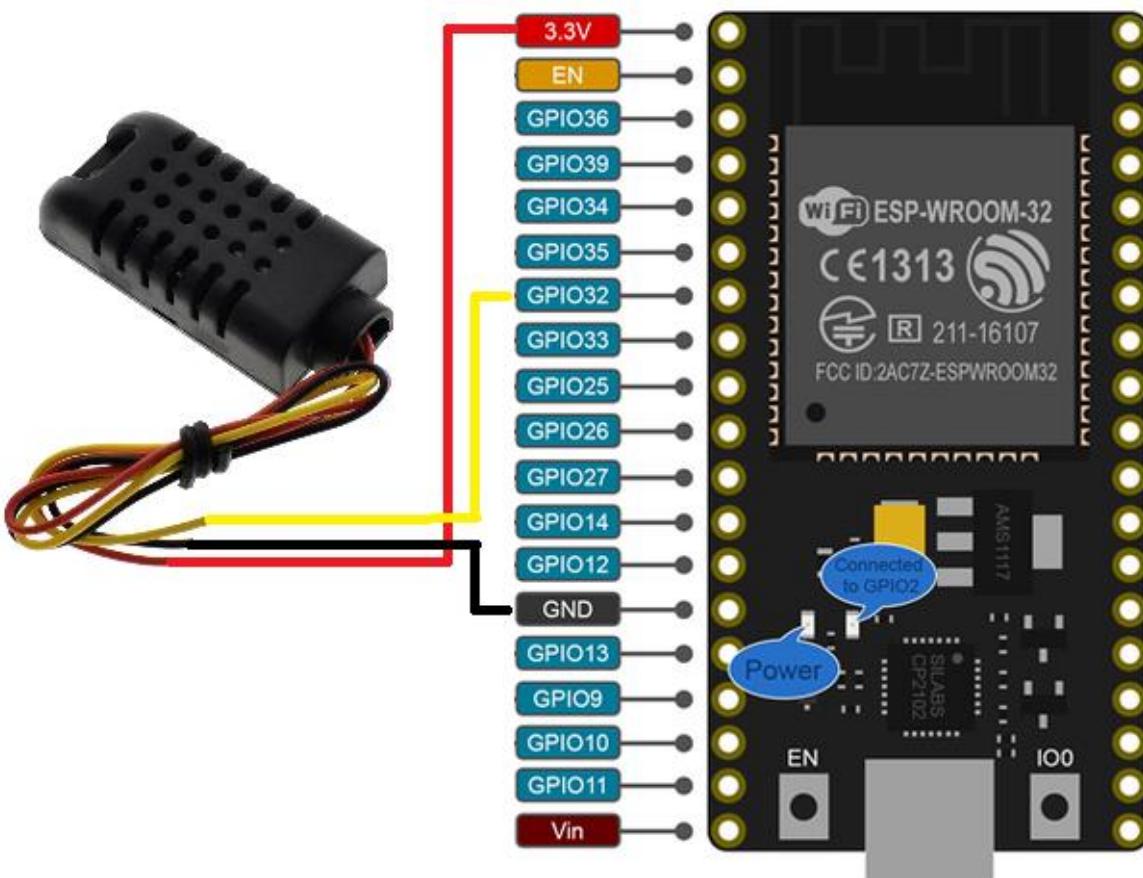
Ahora se observa el monitor de puerto serie mostrando los datos del sensor:

¿Qué datos se aprecian?  
¿Puede ubicar el dato de la temperatura?

## 2.2.6 Usando el sensor de temperatura DHT22 (AM2301) fabricado por AMLOGIC

**Objetivo específico:** Ejecutará el ejemplo que permite verificar el funcionamiento del sensor de temperatura usando el módulo ESP32.

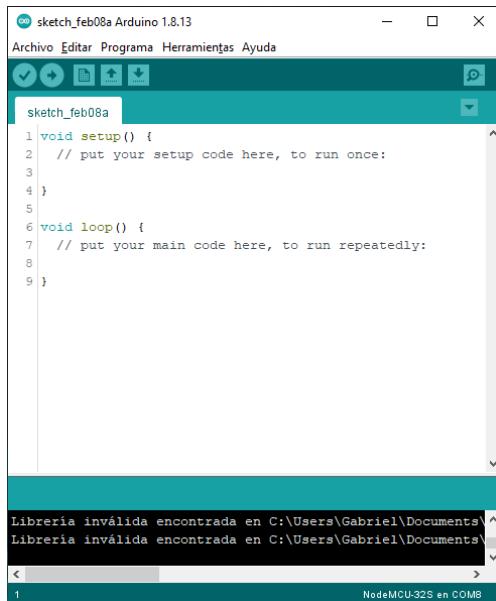
Antes de iniciar, se requiere del conexionado del sensor como se muestra en la figura:



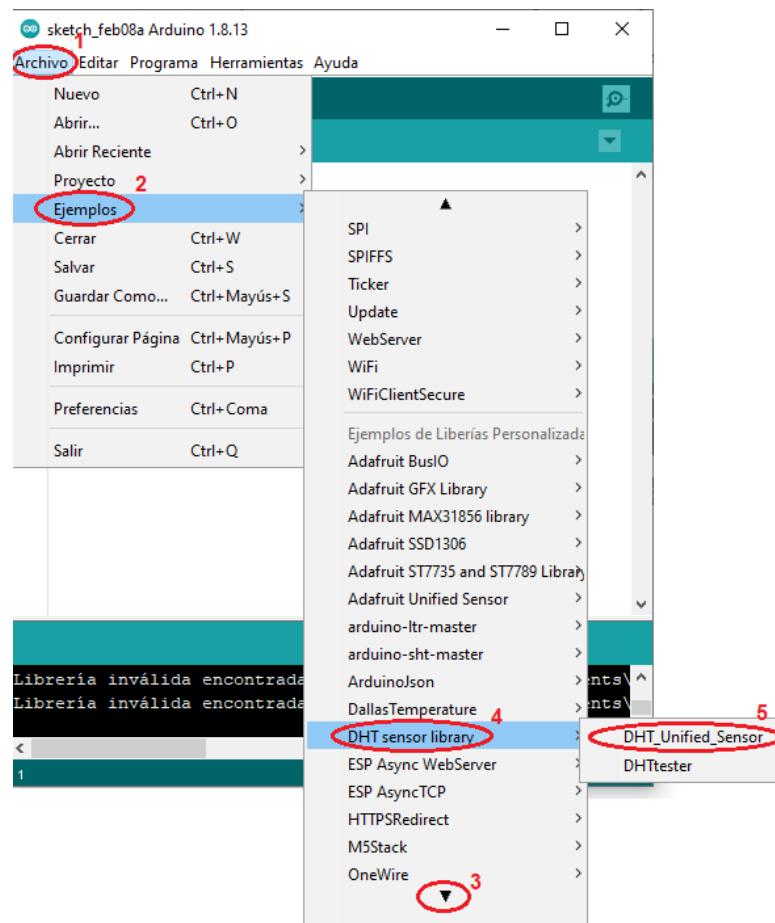
**IMPORTANTE:** El dispositivo puede dañarse si es conectado de manera incorrecta. Los pinos críticos son el GND y 3.3V (cable Negro y rojo)

Adicionalmente, tiene que considerar que la instalación de las bibliotecas (descritas en el punto 1.6 de este manual) ha sido completadas con éxito.

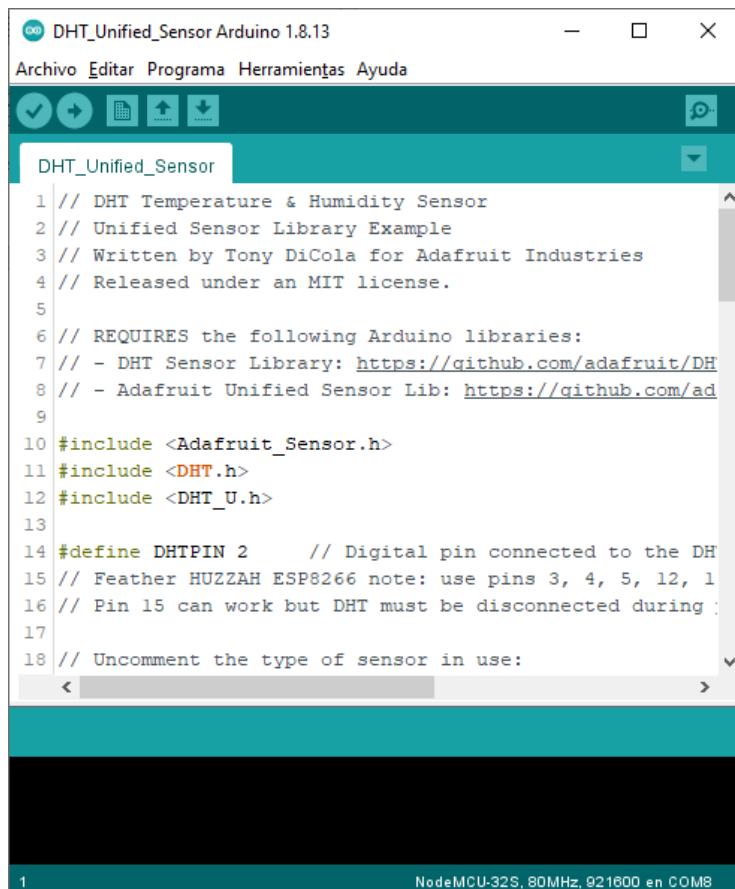
Seguidamente, se requiere abrir el IDE del Arduino como sigue:



Presione en la secuencia indicada los siguientes menús:



... y se abrirá el ejemplo siguiente:



The screenshot shows the Arduino IDE interface with the title bar "DHT\_Unified\_Sensor Arduino 1.8.13". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". Below the menu is a toolbar with icons for file operations. The main window displays the "DHT\_Unified\_Sensor" code. The code is a C++ program for reading DHT temperature and humidity sensors. It includes comments explaining the requirements for Adafruit libraries and defines the digital pin connected to the DHT sensor. The code ends with a note to uncomment the sensor type. At the bottom of the code editor, it says "NodeMCU-32S, 80MHz, 921600 en COM8".

```
1 // DHT Temperature & Humidity Sensor
2 // Unified Sensor Library Example
3 // Written by Tony DiCola for Adafruit Industries
4 // Released under an MIT license.
5
6 // REQUIRES the following Arduino libraries:
7 // - DHT Sensor Library: https://github.com/adafruit/DHT
8 // - Adafruit Unified Sensor Lib: https://github.com/adafruit/Adafruit Unified Sensor Library
9
10 #include <Adafruit_Sensor.h>
11 #include <DHT.h>
12 #include <DHT_U.h>
13
14 #define DHTPIN 2      // Digital pin connected to the DHT
15 // Feather HUZZAH ESP8266 note: use pins 3, 4, 5, 12, 13
16 // Pin 15 can work but DHT must be disconnected during upload
17
18 // Uncomment the type of sensor in use:
```

Ubique la línea 14 y cámbiela como sigue:

```
#define DHTPIN 32      // Digital pin connected to the DHT sensor
```

Compile y descargue su código a la tarjeta de desarrollo.

... OJO, No olvide que:



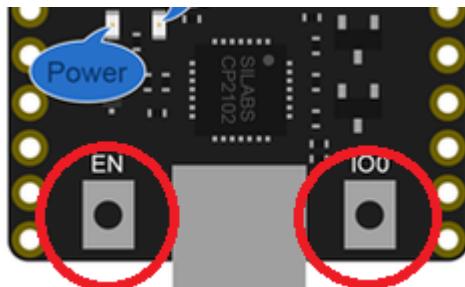
1. Con el botón  se verifica e inicia la solicitud de conexión con el módulo para la descarga del código a la tarjeta de desarrollo.
2. En algunos modelos de NodeMCU-32S no se requiere acción adicional para que el código empiece a descargarse, así que, no se mostrará la solicitud de conexión. El código se descargará automáticamente.
3. En otros modelos de hardware, mostrará una solicitud de conexión. En la parte inferior del IDE se mostrará la leyenda “conectando” así como se muestra en la siguiente figura:



... dependiendo del hardware adquirido, puede probar **UNA** de las acciones siguientes:

- Presione momentáneamente el botón IO0 y el código deberá empezar a descargarse a su módulo NodeMCU-32S.
- Presione y mantenga presionado el botón IO0, seguidamente presionar momentáneamente el botón de EN ó reset y por último liberar el botón IO0. Siguiendo éstos pasos el código deberá empezar a descargarse a su módulo NodeMCU-32S.

Ambos botones están ubicados a los costados del puerto microUSB de su módulo NodeMCU-32S y deberá empezar a descargar el código a su módulo NodeMCU32S.



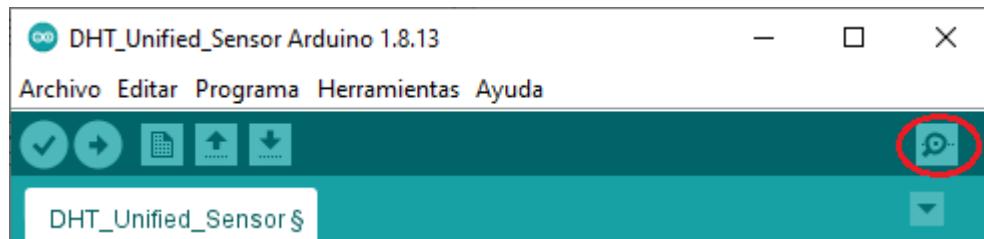
- Si la barra de notificación (intermedia entre la zona del código y la zona negra) del IDE cambia a naranja indica que ocurrió un error en la descarga, para corregirlo siga el procedimiento descrito al final del punto 1.3 “Errores en la descarga del código”.

Una vez descargado en su tarjeta de desarrollo se requiere abrir el monitor serial y ajustar el baud rate tal como se describe en el punto 1.5 de este manual:

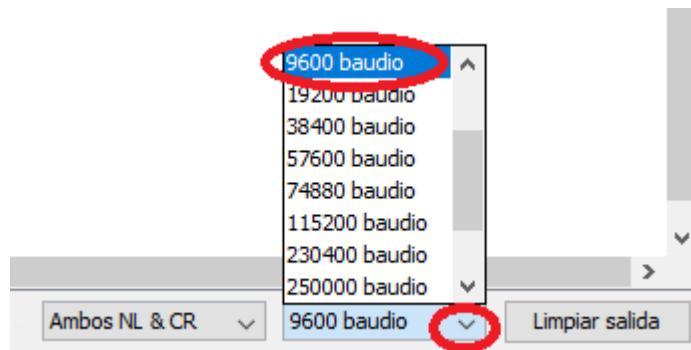
Observe y memorice el valor señalado a continuación:

```
30 void setup() {  
31     Serial.begin(9600);
```

Abra el monitor del puerto serie presionando el botón marcado en la siguiente figura:



Ajuste los baudios como se indica a continuación:



Ahora se observa el monitor de puerto serie mostrando los datos del sensor:

- ¿Qué datos se aprecian?
- ¿Puede ubicar el dato de la temperatura?
- ¿Puede ubicar el dato de la humedad?

## 2.2.7 Mostrando los valores del sensor en la pantalla del SSD1306

**Objetivo específico:** Ejecutará y analizará el ejemplo que permite mostrar los valores del sensor de temperatura DHT22 (AM2301) usando el módulo ESP32.

Antes de iniciar, se requiere descargar el ejemplo del sitio: <https://github.com/gpoolb/ESP32> en la carpeta “ssd1306\_con\_DHT22”. Dicho ejemplo contiene la integración de ambos Hardware basado en los puntos 2.2.4 y 2.2.6 de este manual. Revise los comentarios, ya que, ahí se detalla paso a paso el desarrollo del programa. Lo más relevante a revisar es esta instrucción:

```
// Se muestra el nombre del sensor en la parte superior de la pantalla
// display.fillRect(CoordEjeX, CoordEjeY, AnchoCaracter * NumCaracter * TamanoTexto,
AlturaCaracter * TamanoTexto, Color);
/* No olvidar que el tamaño del texto standart es 5 * 7 pixeles,
* se anade un pixel adicional por la separación de caracteres
* quedando en 6 pixeles de ancho * 8 pixeles de alto
*/
display.fillRect(22, 0, 6 * 14 * 1, 8 * 1, SSD1306_BLACK); // Se borra el texto anterior de 14
caracteres (paso 1)
//display.setFont(&FreeMono9pt7b);
display.setTextSize(1); // Se elige el tamaño del texto (3X) (paso 2)
display.setTextColor(SSD1306_WHITE); // Se elige el color del texto (blanco) (paso 3)
display.setCursor(22, 0); // Se elige las coordenadas donde se coloca el texto (paso 4)

display.print("SENSOR DIGITAL"); // Se coloca en memoria el texto (paso 5)
display.display(); // Se muestra el texto en pantalla
```

... cada vez que se requiera mostrar algo en la pantalla, se necesitan esas instrucciones (No olvidar que el texto en gris son comentarios hechos por el autor y no tienen efecto en el código):

**1. Borre** el campo donde se desea mostrar la información usando la función de rectángulo lleno (fillRect), esta instrucción requiere 5 parámetros:

- La coordenada en el eje ‘X’* donde empieza la parte izquierda del rectángulo, recuerde que la pantalla es de 128 x 64 pixeles, o sea, ‘X’ puede tener un valor máximo de 127
- La coordenada en el eje ‘Y’* donde empieza la parte superior del rectángulo, recuerde que la pantalla es de 128 x 64 pixeles, o sea, ‘Y’ puede tener hasta 63

c) *El ancho del rectángulo*, cuide que este valor no exceda de 127 y se calcula mediante la siguiente ecuación:

AnchoCaracter: Se maneja una fuente de 5 (Ancho) x 7 (Alto) pixeles (px) adicionalmente, se considera 1px adicional de separación tanto en el ancho como en la altura, resumiendo, este valor es de 6.

NumCaracter: Es la cantidad de caracteres que desea mostrar en esa línea.

TamanoTexto: Es el tamaño del texto que desea mostrar (1, 2, 3, ... etc)

- La altura del rectángulo*, cuide que este valor no exceda de 63 y se calcula con

la siguiente ecuación:

AlturaCaracter, esta es la altura del carácter, si considera al punto anterior (el texto de 5x7) y su debida separación (1px) su valor es de 6.

TamanoTexto: Es el tamaño del texto que desea mostrar (1, 2, 3, ... etc)

e) Color, esta pantalla sólo tiene dos colores disponibles: SSD1306\_BLACK y SSD1306\_WHITE, por lo que, con el color black se “borra” la pantalla y con el color blanco se muestra el contenido en pantalla.

**2. Ajuste el tamaño del texto** que desea mostrar los valores podrían ser 1, 2, 3, ... etc., la función que permite hacer esto es setTextSize.

**3. Ajuste el color** del contenido que desea mostrar, recuerde sólo tiene dos colores disponibles: SSD1306\_BLACK y SSD1306\_WHITE, por lo que, con el color blanco se muestra el contenido en pantalla. La función que permite hacer esto es setTextColor. La función que permite hacer esto es setTextColor.

**4. Fije las coordenadas** donde se desea mostrar el contenido, recuerde que la pantalla es de 128 x 64 pixeles, o sea, ‘X’ puede tener un valor entre 0 ~ 127 y el eje ‘Y’ tiene valores de 0 ~ 63 (el valor ‘0,0’ representa la esquina superior izquierda del display). La función que permite hacer esto es setCursor.

**5. Ponga en el buffer** el contenido a mostrar, la función que permite hacer esto es print. Observe que también se puede utilizar variables con contenido numérico (Tipo int, long, etc.) ó texto (Tipo String).

**6. Mostrar texto** en pantalla, la función que permite hacer esto es display. La instrucción anterior, envía el comando para transferir el contenido almacenado en memoria al display.

Compile y descargue el código a su módulo NodeMCU.

... OJO, No olvide que:



1. Con el botón se verifica e inicia la solicitud de conexión con el módulo para la descarga del código a la tarjeta de desarrollo.

2. En algunos modelos de NodeMCU-32S no se requiere acción adicional para que el código empiece a descargarse, así que, no se mostrará la solicitud de conexión. El código se descargará automáticamente.

3. En otros modelos de hardware, mostrará una solicitud de conexión. En la parte inferior del IDE se mostrará la leyenda “conectando” así como se muestra en la siguiente figura:



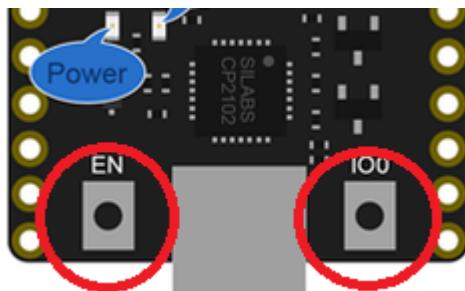
The screenshot shows a terminal window with a red circle highlighting the status bar at the bottom which reads "NodeMCU-32S en COM4". The main text area displays the following output:

```
Subiendo...
esptool.py v2.6
Serial port COM4
Connecting.....
```

... dependiendo del hardware adquirido, puede probar **UNA** de las acciones siguientes:

- Presione momentáneamente el botón IO0 y el código deberá empezar a descargarse a su módulo NodeMCU-32S.
- Presione y mantenga presionado el botón IO0, seguidamente presionar momentáneamente el botón de EN ó reset y por último liberar el botón IO0. Siguiendo éstos pasos el código deberá empezar a descargarse a su módulo NodeMCU-32S.

Ambos botones están ubicados a los costados del puerto microUSB de su módulo NodeMCU-32S y deberá empezar a descargar el código a su módulo NodeMCU32S.



- Si la barra de notificación (intermedia entre la zona del código y la zona negra) del IDE cambia a naranja indica que ocurrió un error en la descarga, para corregirlo siga el procedimiento descrito al final del punto 1.3 “Errores en la descarga del código”.

#### Actividades:

Compare entre los códigos mostrados en los ejemplos del DHT22 y SSD1306, cuáles son las partes extraídas para implementar este ejemplo.

¿Podría mostrar su nombre en la parte inferior de la pantalla?

#### Sugerencias:

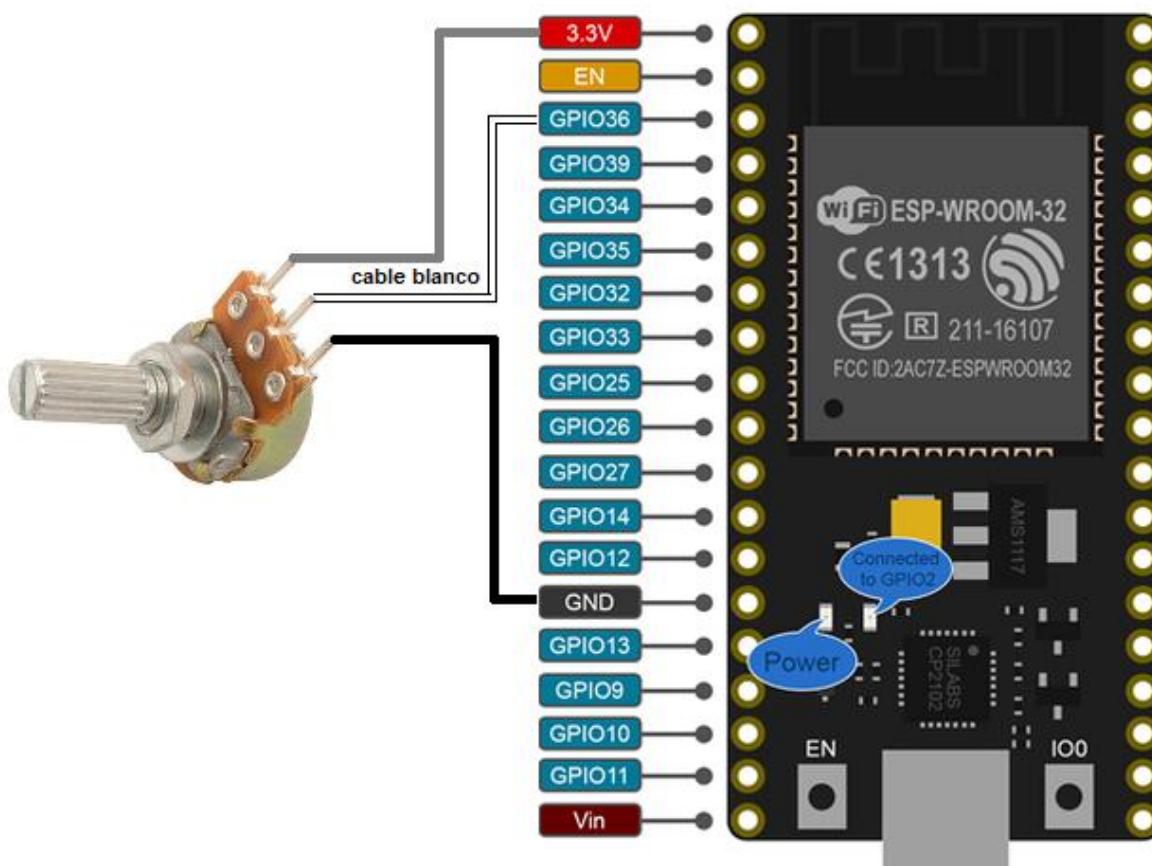
- Use la coordenada (0,55)

- b) Utilice el tamaño de texto con el valor de 1
- c) Sólo es necesario ejecutar las sentencias una vez, ya que, el nombre no cambia.
- d) No es necesario ejecutar la sentencia que borra el texto (fillRect()).

## 2.2.8 Usando el módulo Analog to Digital Converter (ADC)

**Objetivo específico:** Ejecutará el ejemplo que permite verificar el funcionamiento del ADC usando el módulo ESP32.

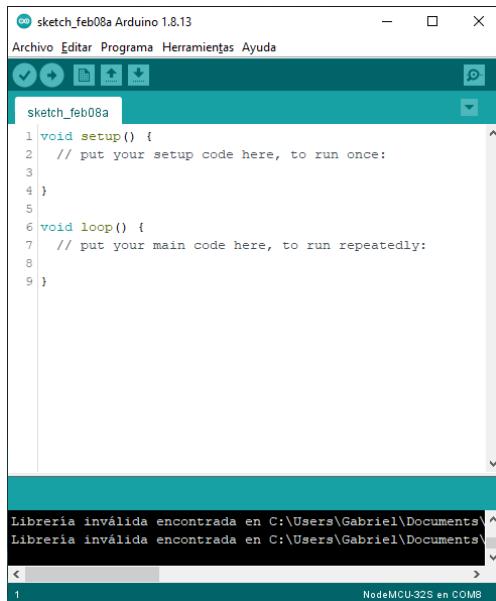
Antes de iniciar, se requiere del conexionado del potenciómetro como se muestra en la figura:



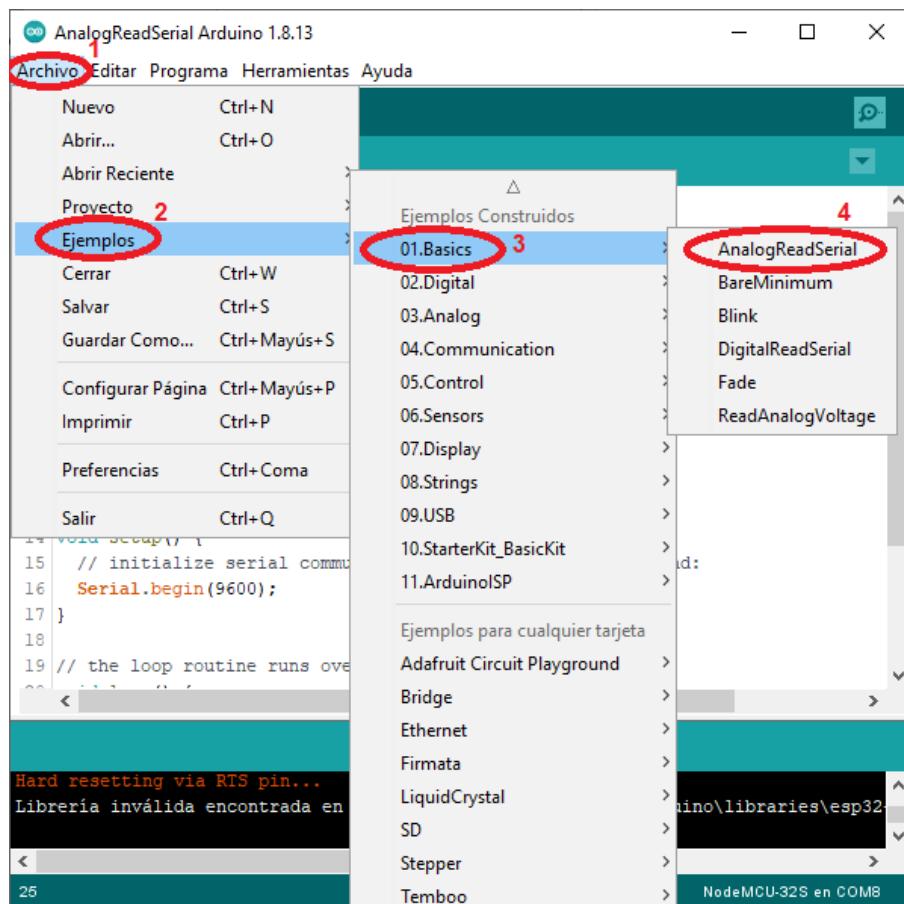
### IMPORTANTE:

**El dispositivo PUEDE DAÑARSE si es conectado de manera incorrecta.** Los pines críticos son el GND y 3.3V (cable Negro y gris).

Seguidamente, se requiere abrir el IDE del Arduino como sigue:



Presione en la secuencia indicada los siguientes menús:



Compile y descargue el código a su módulo NodeMCU.

... **OJO**, No olvide que:



1. Con el botón se verifica e inicia la solicitud de conexión con el módulo para la descarga del código a la tarjeta de desarrollo.
2. En algunos modelos de NodeMCU-32S no se requiere acción adicional para que el código empiece a descargarse, así que, no se mostrará la solicitud de conexión. El código se descargaría automáticamente.
3. En otros modelos de hardware, mostrará una solicitud de conexión. En la parte inferior del IDE se mostrará la leyenda “conectando” así como se muestra en la siguiente figura:

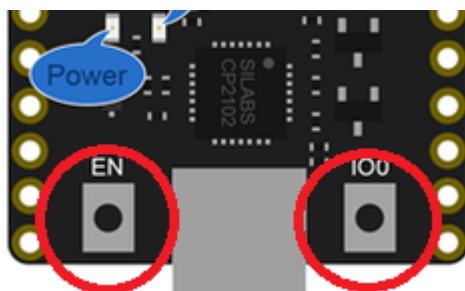


The screenshot shows the Arduino IDE's Serial Monitor window. The title bar says "Subiendo..." (Uploading...). The main text area displays the command "esptool.py v2.6" followed by "Serial port COM4". Below that, it says "Connecting.....". At the bottom right, it shows "NodeMCU-32S en COM4". The entire "Connecting....." text is circled in red.

... dependiendo del hardware adquirido, puede probar **UNA** de las acciones siguientes:

- a) Presione momentáneamente el botón IO0 y el código deberá empezar a descargarse a su módulo NodeMCU-32S.
- b) Presione y mantenga presionado el botón IO0, seguidamente presionar momentáneamente el botón de EN ó reset y por último liberar el botón IO0. Siguiendo éstos pasos el código deberá empezar a descargarse a su módulo NodeMCU-32S.

Ambos botones están ubicados a los costados del puerto microUSB de su módulo NodeMCU-32S y deberá empezar a descargar el código a su módulo NodeMCU32S.



4. Si la barra de notificación (intermedia entre la zona del código y la zona negra)

del IDE cambia a naranja indica que ocurrió un error en la descarga, para corregirlo siga el procedimiento descrito al final del punto 1.3 “Errores en la descarga del código”.

Una vez descargado en su tarjeta de desarrollo se requiere abrir el monitor serial y ajustar el baud rate tal como se describe en el punto 1.5 de este manual:

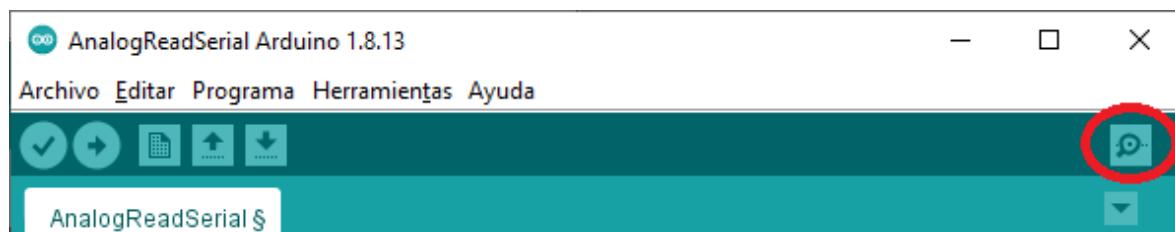
Observe y memorice el valor señalado a continuación:

```

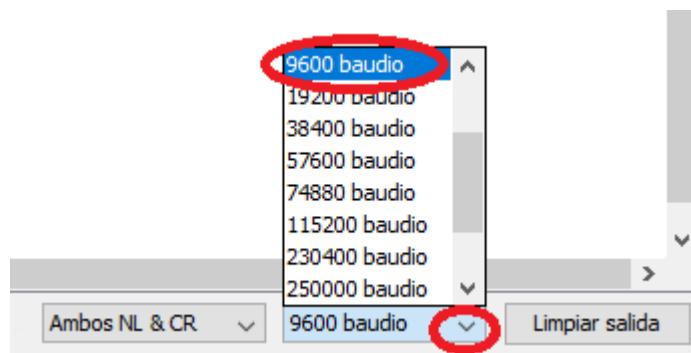
13 // the setup routine runs once when you press reset:
14 void setup() {
15     // initialize serial communication at 9600 bits per second:
16     Serial.begin(9600); // A red arrow points to this line
17 }

```

Abra el monitor del puerto serie presionando el botón marcado en la siguiente figura:



Ajuste los baudios como se indica a continuación:



Ahora se observa el monitor de puerto serie mostrando los datos del sensor:

Actividades:

Mueva el potenciómetro de izquierda a derecha y viceversa.

Observe las lecturas en el puerto serie.

Determine cuál es el valor máximo y mínimo que se obtiene al mover el

potenciómetro.

Podría mostrar esta lectura en la pantalla OLED descrito en el punto 2.2.7?

Podría condicionar que, para el primer tercio del rango del potenciómetro encienda el LedB, el segundo tercio del rango de lecturas apague el LedB y encienda el LedG y por último en el tercio de lecturas más alto encienda el LedR y apague los LedB y LedG, basándose del código descrito en el punto 2.2.3?

Sugerencia:

Declare los valores constantes como sigue:

```
// El ADC es de 12 bits por lo que el valor  
// máximo es 2^12 = 4096, por lo que,  
// ese valor se divide entre tres.  
const int primerLimite = 1365; // 4096 * (1 / 3)  
const int segundoLimite = 2730; // 4096 * (2 / 3)
```

Luego, en la rutina principal loop() puede hacer las comparaciones pertinentes:

```
if ((0 <= sensorValue) && (sensorValue < primerLimite)){  
    // Enciende el ledR y apague ledG y ledB  
} else if ((primerLimite <= sensorValue) && (sensorValue < segundoLimite)){  
    // Enciende el ledG y apague ledR y ledB  
} else if ((segundoLimite <= sensorValue) && (sensorValue < 4096)){  
    // Enciende el ledB y apague ledR y ledG  
}
```

Como algunas declaraciones son redundantes, se reescribe el código:

```
if (sensorValue < primerLimite){  
    // 0 < sensorValue < primerLimite  
    // Enciende el ledR y apague ledG y ledB  
} else if ((primerLimite <= sensorValue) && (sensorValue < segundoLimite)){  
    // primerLimite <= sensorValue < segundoLimite  
    // Enciende el ledG y apague ledR y ledB  
} else {  
    // segundoLimite <= sensorValue < 4096  
    // Enciende el ledB y apague ledR y ledG  
}
```

Como podrá observar algunas comparaciones fueron eliminadas sin embargo, ambos segmentos de código hacen la misma función.

## Modulo III

### **3. Diseño de WebServer para NodeMCU-32S.**

**Objetivo General:** Aprenderá el procedimiento para publicar una página web en la red local y en Google Apps Script.

### 3.1 El Servidor Web implementado en NodeMCU-32S

En este punto se desarrollará una página web estática la cual, mostrará una gráfica con datos aleatorios y podrá manipular el estado de un led desde la página web.

#### 3.1.1 MDNS

**Objetivo específico:** Usará las herramientas existentes para el protocolo mDNS y encontrará al servidor en la red por nombre.

El sistema de nombres de dominio (o DNS, del inglés domain name system) podría considerarse una extensa guía telefónica: mediante este servicio, los usuarios introducen la dirección web en el navegador y el propio sistema identifica la dirección IP relacionada. En este proceso de resolución de nombre, el propio dispositivo envía la solicitud al servidor DNS correspondiente, donde hay una lista en la que cada nombre de equipo (es decir, la dirección web) tiene asignada la dirección IP correcta. En cambio, el multicast DNS sigue otro sistema. ¿Cómo funciona esta alternativa al DNS clásico?

El multicast DNS (mDNS) es un servicio diseñado para llevar a cabo la resolución de nombres en redes más pequeñas. Para ello, mDNS sigue un sistema distinto al conocido DNS: en lugar de enviar la solicitud a un servidor de nombres, todos los participantes de la red reciben la solicitud. El cliente correspondiente envía un multicast a la red y, de este modo, pregunta con qué integrante de la red coincide el nombre del equipo. El multicast o multidifusión es una forma especial de comunicación en la que un solo mensaje se envía a un grupo de destinatarios. El grupo puede consistir, por ejemplo, en toda la red o en una subred.

Resumiendo, es posible ubicar al servidor por nombre, pero requiere de la instalación de software adicional en la computadora como sigue:

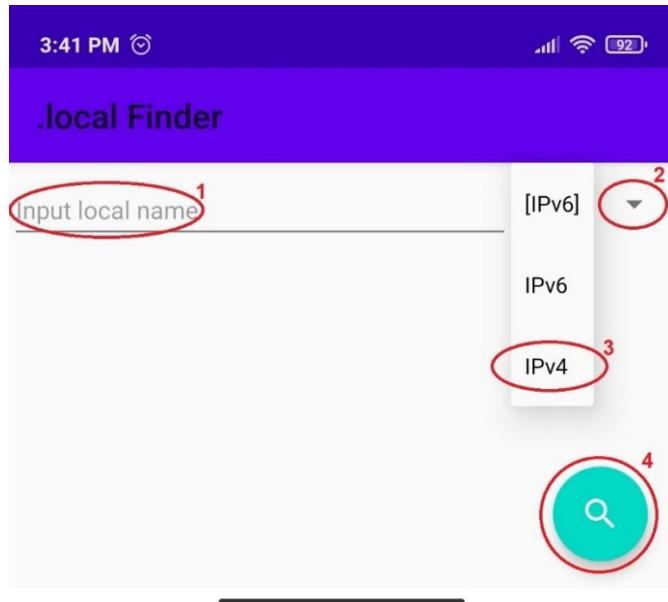
En Android se requiere instalar el programa “.local Finder” disponible en Play Store:



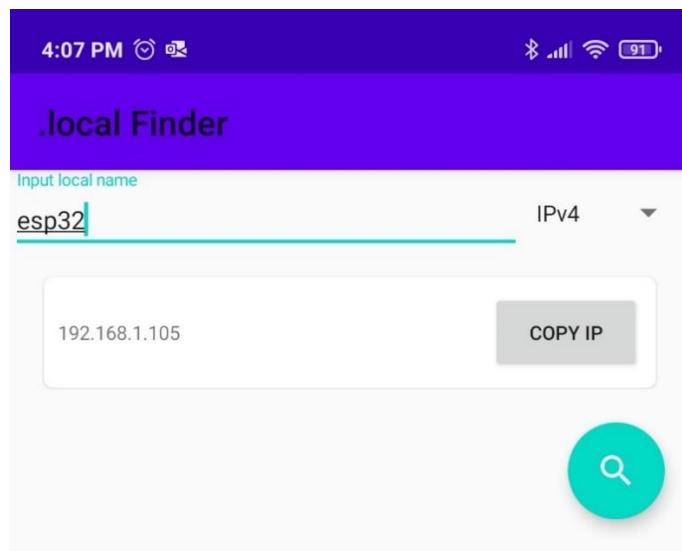
... una vez instalado el logo cambia a color azul oscuro:



Para utilizarlo, se requiere escribir el nombre en el campo: "Input local name", seleccionar IPv4 y presionar el botón con forma de lupa:



**Suponiendo que Ud. tiene un NodeMCU funcionando como servidor** en su red local (se implementará en el punto 3.1.2), al presionar el botón con forma de lupa nos dará este resultado:



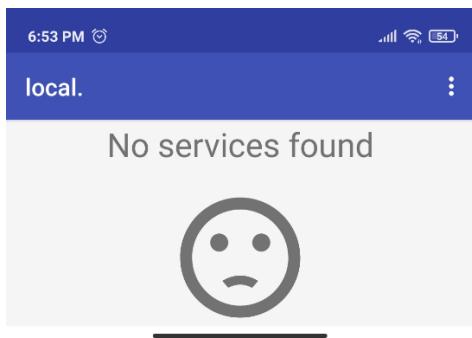
Para utilizar esa información, sólo bastaría con presionar el botón “COPY IP” y pegarlo en la barra de direcciones de su navegador web en Android.

En caso de que se active la búsqueda de Google e intente buscar su dirección web en internet, agregue a la dirección ip anterior lo siguiente: http:// y al final de la dirección ip ponga el carácter de diagonal (/), quedando de esta manera:  
<http://192.168.1.105/>

Existe otro programa interesante para Android y se encarga de encontrar todos los servicios ubicados en la red, en su dispositivo Android, descargue el programa “Service Browser” ubicado en la Play Store que tiene el logo que se muestra:



Instale, abra el programa y se deberá mostrar lo siguiente:



**Suponiendo que Ud. tiene un NodeMCU funcionando como servidor en su red local (se implementará en el punto 3.1.2), adicionalmente le ha agregado la siguiente línea al final del método setup() en su código:**

```
MDNS.addService("http", "tcp", 80);
} // fin del método setup()
```

Al abrir nuevamente el programa se mostrará lo siguiente:

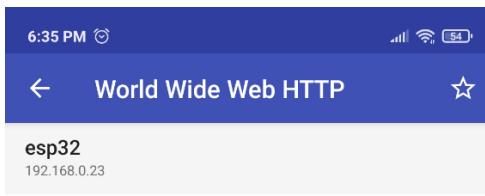


97/224

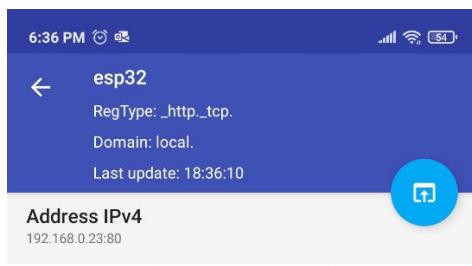
Es importante recalcar que, si se encuentra en un sitio con muchos servicios publicados en la red, le saldrá una lista en ocasiones muy extensa, para ello siempre ubique el nombre de este servicio y con el dedo, presione el área del texto mostrado en la imagen (el área se marca en un rectángulo rojo):



Y observará lo siguiente:



Nuevamente, presione el área de texto marcado con el nombre del servidor que usted trata de encontrar (en caso de salir una lista de nombres), en este caso “esp32” y se mostrará lo siguiente:

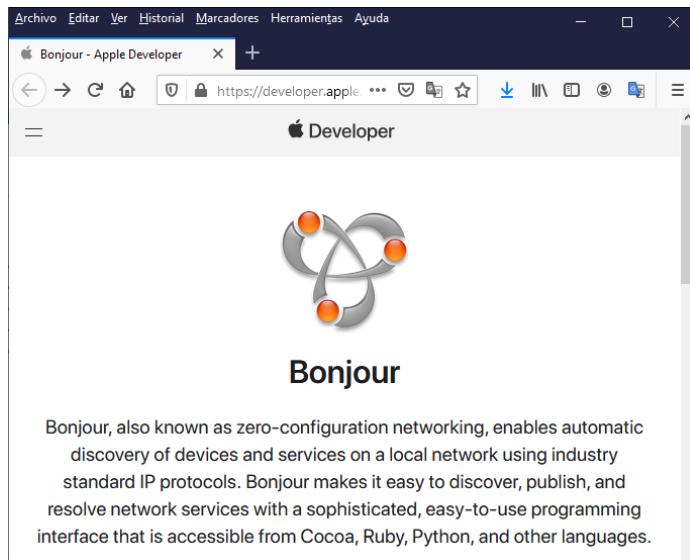


En caso de NO mostrar el botón azul, regrese un paso anterior (presionando la flecha blanca que indica hacia la izquierda) y repita el paso anterior. Ahora presione el botón azul claro con forma de circulo e incluye un cuadro con flecha hacia arriba y observará lo siguiente:

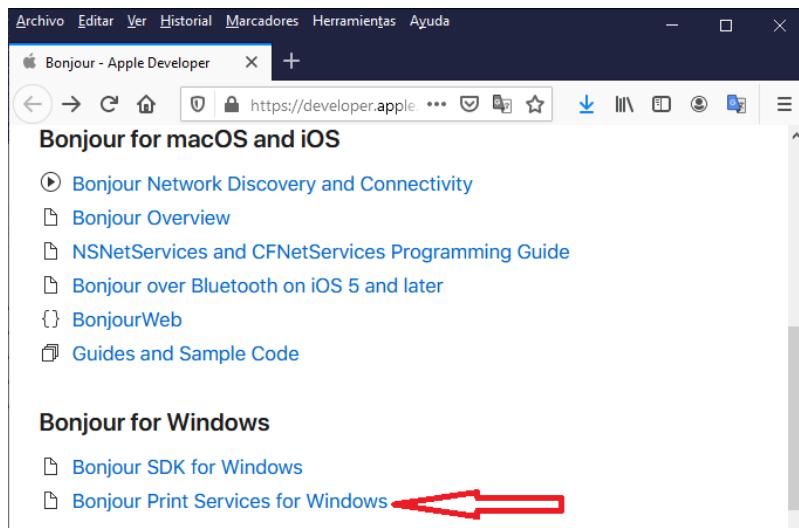


... el cual, muestra el contenido del servidor que se encuentra publicando en la red.

En **Windows** se requiere instalar el Bonjour Print Services for Windows ubicado en el sitio de apple: <https://developer.apple.com/bonjour/>:

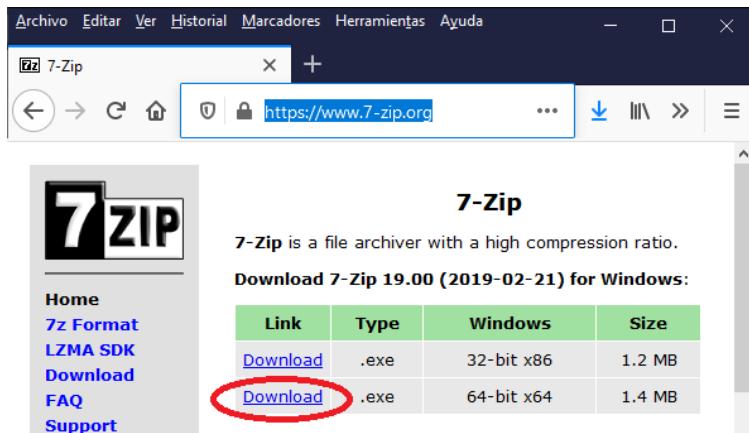


Deslice la página web hacia abajo hasta encontrar lo siguiente:



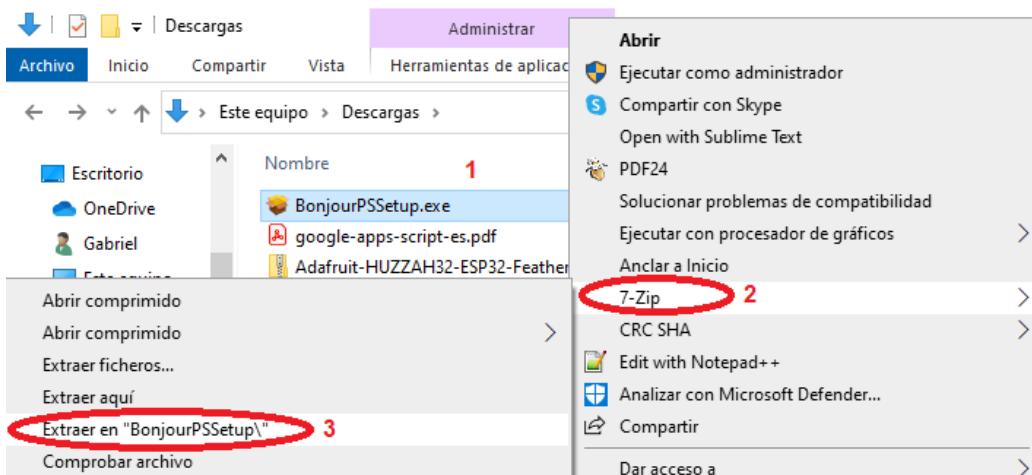
... déle un click al texto que se encuentra marcado en la figura anterior y descargue el archivo.

Para no tener que instalar todos los programas que se incluyen en el paquete, se requiere que descargue el programa 7zip en el sitio <https://www.7-zip.org/> así como sigue:

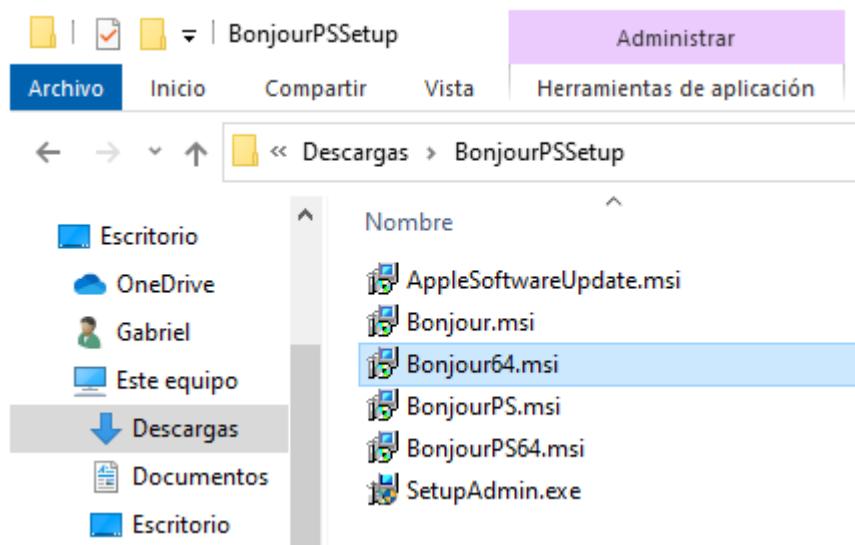


Suponiendo que su Windows sea de la versión de 64 bits descargue el programa en el link marcado en la figura anterior, si desconoce su versión de Windows instalada, de todas maneras, descargue esta versión y si su sistema operativo le informa de algún error, y no permite instalarlo, entonces pruebe con la versión de 32 bits.

Descargue e instale el 7zip en su computadora con las opciones que trae ya predeterminadas. Una vez instalado el 7zip y haber descargado el Bonjour desde el sitio web, descomprima como se muestra en la figura, presione el **botón derecho** del mouse sobre el nombre del archivo y en el menú que sale a continuación, siga la secuencia indicada en la siguiente figura:

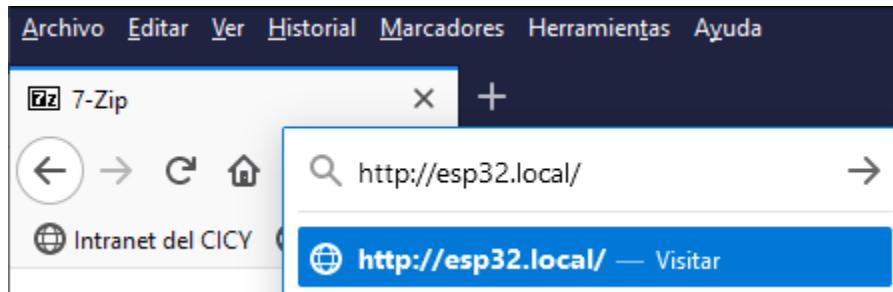


Entre a la carpeta recién creada por el 7zip en el punto anterior:



... e instale el Bonjour64.msi, en caso de que su sistema operativo le informe de algún error, y no permite instalarlo, entonces pruebe con el Bonjour.msi

**Suponiendo que Ud. tiene un NodeMCU funcionando como servidor** en su red local (se implementará en el punto 3.1.2), abra su navegador web y teclee lo siguiente:



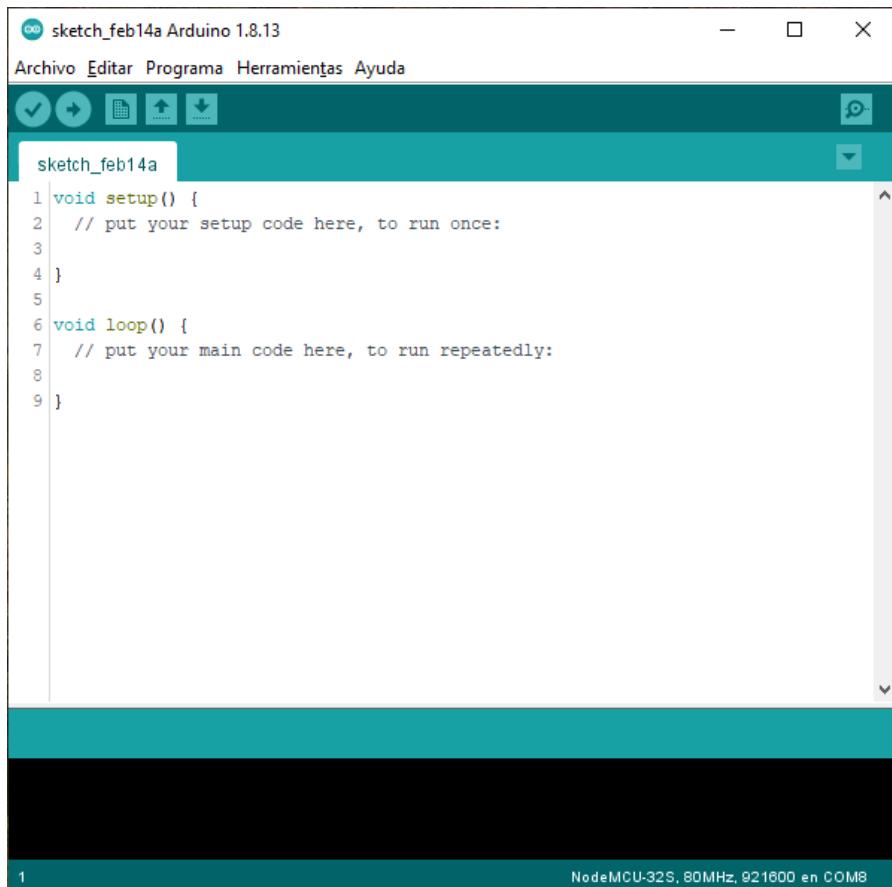
En Linux se requiere la instalación del programa llamado avahi, que, por lo general, ya lo trae de manera nativa, es decir, no se requiere instalar programa adicional.

Ahora, para poder dar uso a lo anterior, se implementará el servidor en el NodeMCU en el siguiente punto.

### 3.1.2 Advanced Web Server

**Objetivo específico:** Aprenderá a obtener los recursos configurados en un servidor implementado en un NodeMCU-32S, identificará los métodos asociado al recurso.

El primer ejemplo relevante para conocer la estructura de un servidor web implementado en el NodeMCU-32S es el que se verá a continuación, partimos del sketch en el IDE del Arduino:

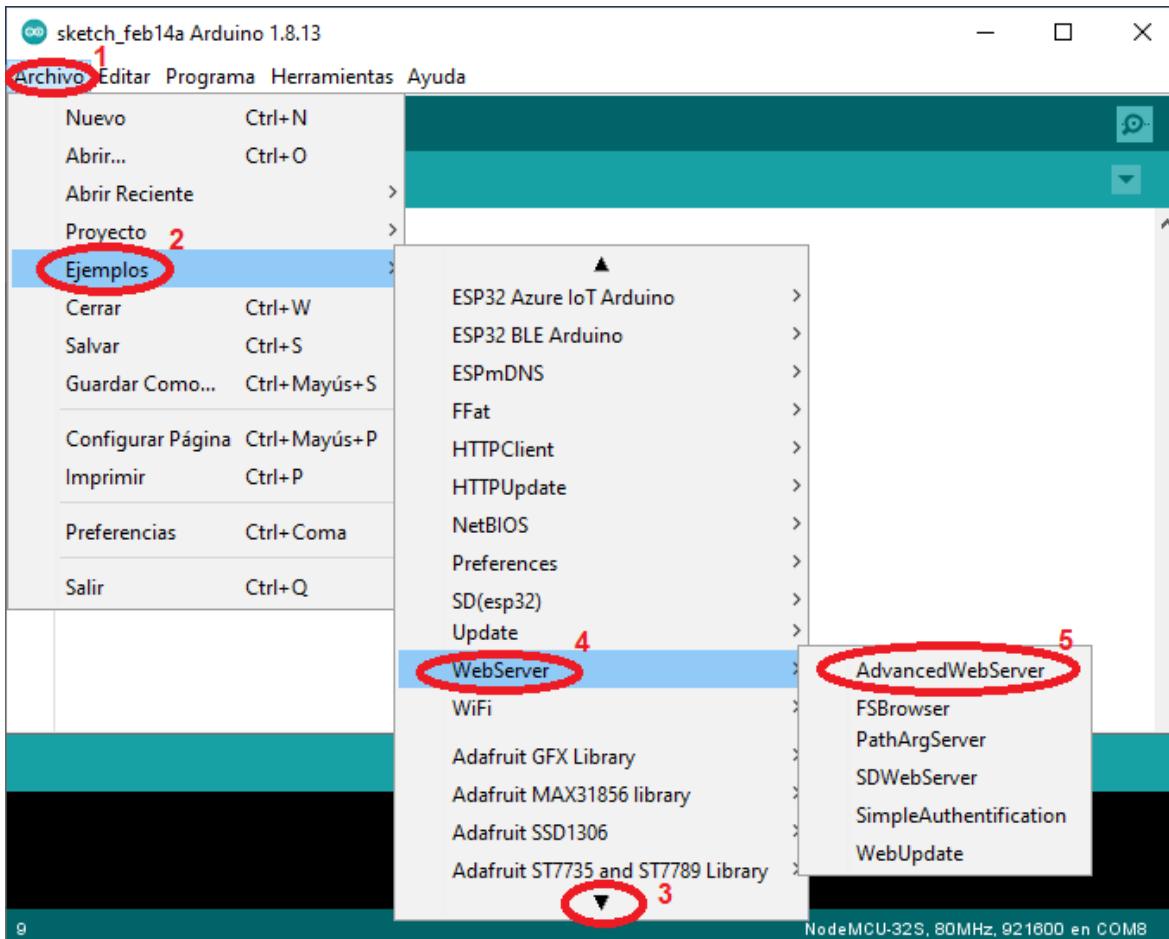


The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** sketch\_feb14a Arduino 1.8.13
- Menu Bar:** Archivo Editar Programa Herramientas Ayuda
- Toolbar:** Includes icons for Save, Run, Upload, and others.
- Sketch Area:** Displays the code for 'sketch\_feb14a'.

```
1 void setup() {  
2   // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly:  
8 }  
9 }
```
- Status Bar:** Shows the connection status: NodeMCU-32S, 80MHz, 921600 en COM8

... abra el ejemplo siguiendo la secuencia indicada en la siguiente figura:



Antes de descargar este código al NodeMCU es necesario completar las siguientes configuraciones, cambie las definiciones de las líneas indicadas a continuación:

```

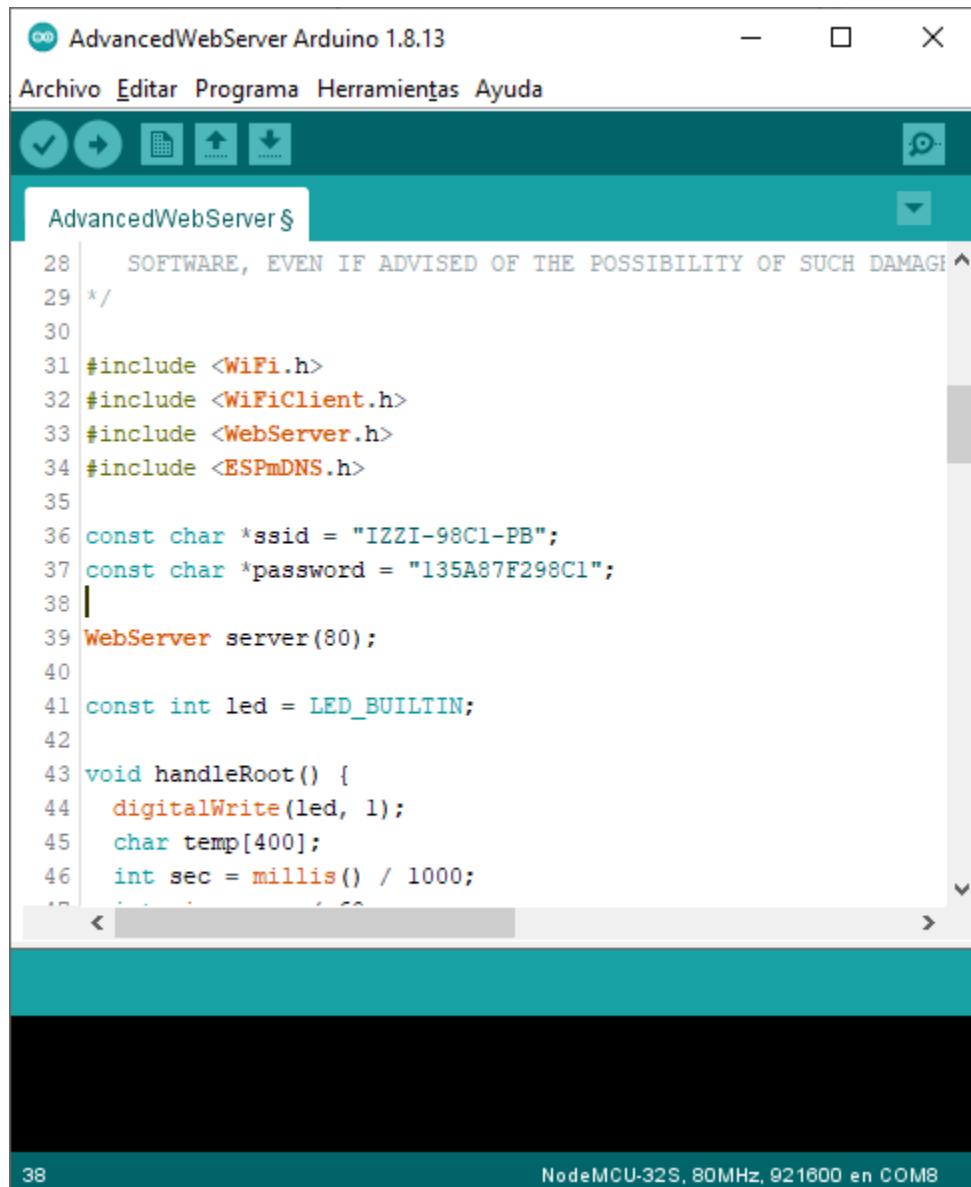
36 | const char *ssid = "YourSSIDHere"; // Establezca el nombre del WiFi (Internet)
37 | const char *password = "YourPSKHere"; // Establezca la clave del WiFi (Internet)

41 | const int led = 13; // Cambie el valor por 2 ó LED_BUILTIN
42 |

```

Compile y descargue el programa a su NodeMCU siguiendo los pasos descritos en la sección 2.2.1 a partir del sexto paso.

**ES IMPORTANTE** considerar que, si por alguna razón omite un punto y coma, alguna comilla o no respeta la estructura de las definiciones anteriores, el código **NO COMPILARA** y genera errores. Para evitar eso, respete el modo de escritura. En este caso, el código queda así:



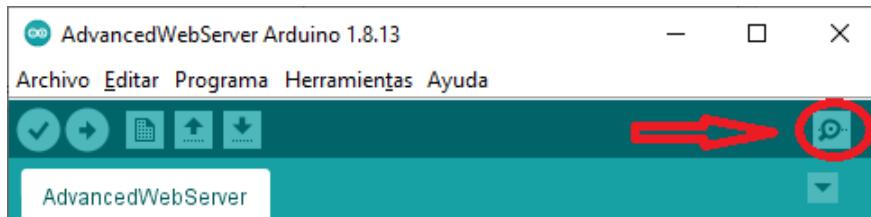
The screenshot shows the AdvancedWebServer Arduino IDE interface. The title bar reads "AdvancedWebServer Arduino 1.8.13". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". The toolbar has icons for save, upload, and refresh. The main window displays the following C++ code:

```
28 SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE
29 */
30
31 #include <WiFi.h>
32 #include <WiFiClient.h>
33 #include <WebServer.h>
34 #include <ESPmDNS.h>
35
36 const char *ssid = "IZZI-98C1-PB";
37 const char *password = "135A87F298C1";
38
39 WebServer server(80);
40
41 const int led = LED_BUILTIN;
42
43 void handleRoot() {
44     digitalWrite(led, 1);
45     char temp[400];
46     int sec = millis() / 1000;
47     sprintf(temp, "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n
48     <html>
49         <head>
50             <title>NodeMCU-32S</title>
51         </head>
52         <body>
53             <h1>NodeMCU-32S</h1>
54             <p>Connected to %s</p>
55             <p>IP: %s</p>
56             <p>LED: %d</p>
57             <p>Time: %d sec</p>
58         </body>
59     </html>
60 
```

The status bar at the bottom shows "NodeMCU-32S, 80MHz, 921600 en COM8".

... compile y descargue el código a su NodeMCU, posteriormente, al subir el código a su NodeMCU, abra el puerto serie (los pasos detallados se describen en la última sección del punto 1.5):

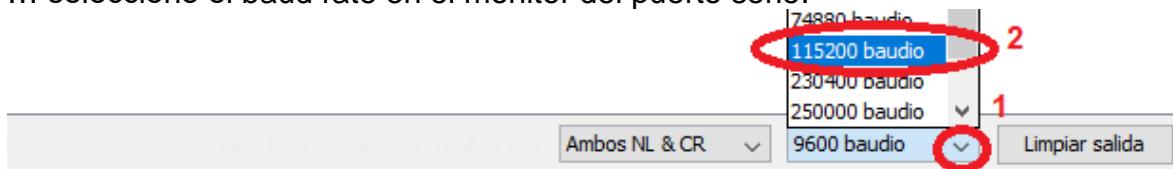
Abra el monitor del puerto serie:



... antes de abrir el monitor de puerto serie, memorice la velocidad de transferencia del puerto serie (baud rate) como sigue:

```
94 | digitalWrite(led, 0);
95 | Serial.begin(115200); ← Red arrow pointing to this line
96 | WiFi.mode(WIFI_STA);
```

... seleccione el baud rate en el monitor del puerto serie:



... presione el botón de RESET ó EN de su tarjeta NodeMCU y observe el monitor del puerto serial:

```
Connected to 192.168.1.105
IP address: 192.168.1.105
MDNS responder started
HTTP server started
```

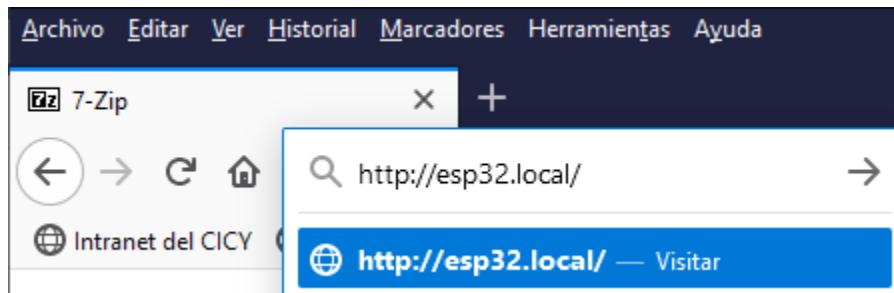
Se observa una serie de números marcadas con la leyenda “IP address”, copie el número y péguelo en su navegador de internet con el siguiente formato <http://192.168.1.105/>, así como se muestra:



.. y presione la tecla “ENTER” y espere a que el navegador muestre su contenido:



... describa lo que observa en pantalla. Haciendo uso del programa instalado en el punto anterior (3.1.1) teclee en su navegador web lo siguiente:

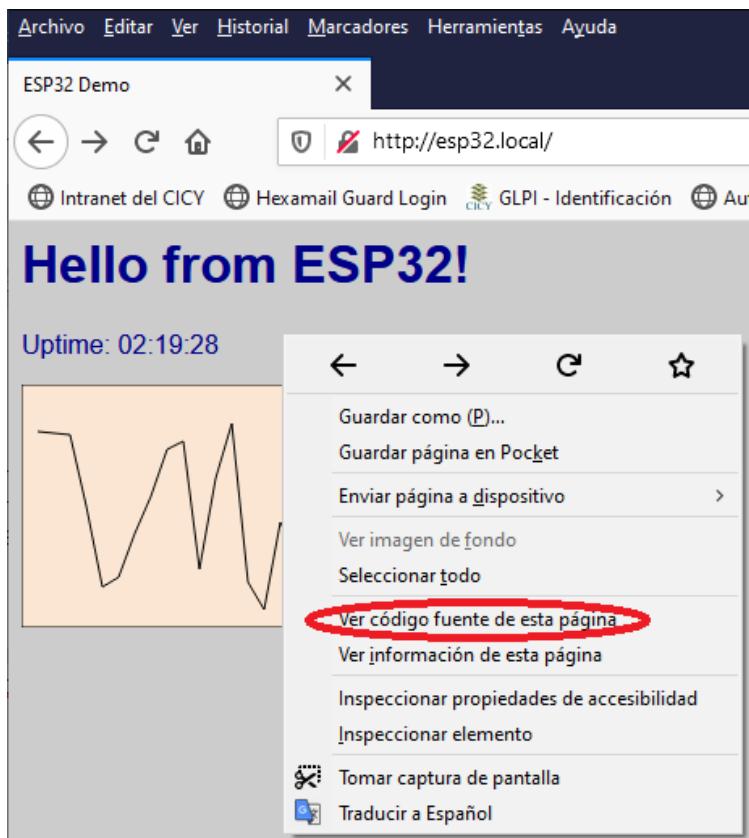


... y observe nuevamente el resultado:



A partir de este punto ya no se harán referencias a la dirección IP, ya que, en cada caso es diferente y puede prestarse a confusiones. Ahora, con las herramientas instaladas en el punto 3.1.1 se hará referencia al servidor por nombre (este no cambia en todos los casos)

Ahora presione el botón derecho del mouse en el área fuera de la gráfica y en el menú que sale a continuación seleccione “ver código fuente de la página”:



... y se mostrará lo siguiente:

```
1 <html> <head> <meta http-equiv='refresh' content='5' />
```

Esto es una línea bastante larga que contiene el texto que envía el servidor embebido en el NodeMCU-32S, para evitar eso, necesitamos agregarle al código el carácter \n a cada línea, **ES IMPORTANTE** considerar que, si por alguna razón omite un punto y coma, alguna comilla o no respeta la estructura de las definiciones anteriores, el código **NO COMPILARA** y genera errores. Para evitar eso, respete el modo de escritura. El código debe quedar, así como se muestra a continuación:

AdvancedWebServerConComentarios Arduino 1.8.13

Archivo Editar Programa Herramientas Ayuda

AdvancedWebServerConComentarios

```

43 void handleRoot() { // Método que envía la página WEB al cliente que lo solicite
44   digitalWrite(led, 1); // Enciende el Led incorporado en su módulo NodeMCU
45   char temp[400]; // Se define una variable la cual, contendrá la página WEB en forma c
46   int sec = millis() / 1000; // Se define una variable que almacene los segundos a partir
47   int min = sec / 60; // Se define una variable que almacene el cálculo de los minutos
48   int hr = min / 60; // Se define una variable que almacene el cálculo de las horas
49   sprintf(temp, 400, \
50   /******INICIA LA PAGINA WEB *****/
51   "<html>\n\
52   <head>\n\
53     <meta http-equiv='refresh' content='5' />\n\
54     <title>ESP32 Demo</title>\n\
55     <style>\n\
56       body { background-color: #cccccc; font-family: Arial, Helvetica, Sans-Serif; Color: \
57     </style>\n\
58   </head>\n\
59   <body>\n\
60     <h1>Hello from ESP32!</h1>\n\
61     <p>Uptime: %02d:%02d:%02d</p>\n\
62     <img src=\"/test.svg\" />\n\
63   </body>\n\
64 </html>\n",
65   /******TERMINA LA PAGINA WEB *****/
66   hr, min % 60, sec % 60 // Aquí se calcula del tiempo en que inicia la ejecución
67 );
68 server.send(200, "text/html", temp); // Sentencia que envía la página web al cliente
69   digitalWrite(led, 0); // Se apaga el led incorporado en su módulo NodeMCU
70 } // Termina método que envía la página WEB al cliente que lo solicita

```

Subido

Hard resetting via RTS pin...

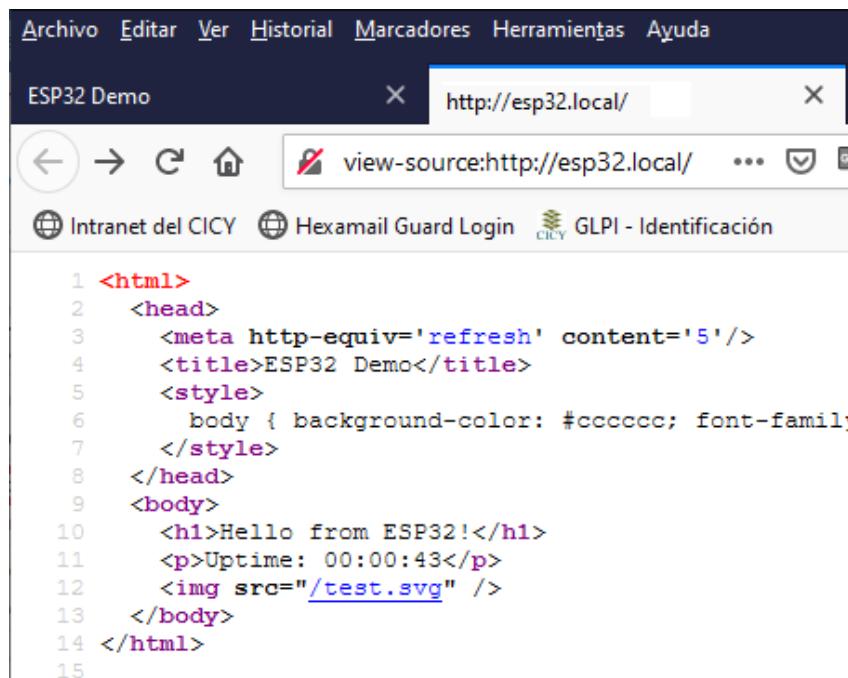
Librería inválida encontrada en C:\Users\Gabriel\Documents\Arduino\libraries\esp32-camera: No e

64 NodeMCU-32S en COM8

... compile y descargue nuevamente el código a su módulo NodeMCU. **ES NECESARIO CERRAR LAS PESTANAS, BORRAR DATOS E HISTORIAL** del navegador web para observar el cambio. Una vez borrado el historial, teclee en el navegador web la dirección anterior <http://esp32.local/> y obtendrá la misma página. Nuevamente, **vea el código fuente de la página**, así como sigue:



... e inmediatamente observará:



```

1 <html>
2   <head>
3     <meta http-equiv='refresh' content='5' />
4     <title>ESP32 Demo</title>
5     <style>
6       body { background-color: #cccccc; font-family:
7     </style>
8   </head>
9   <body>
10    <h1>Hello from ESP32!</h1>
11    <p>Uptime: 00:00:43</p>
12    
13  </body>
14 </html>
15

```

... con el salto de línea implementado en el código, ahora se observa mejor la estructura de la página web, ahora ponga el cursor del mouse en **el hipertexto marcado como "/test.svg"**, dele **un click con el mouse** y observe lo que sucede:



The screenshot shows a web browser window with the title "ESP32 Demo". The address bar displays "http://esp32.local/test.svg". Below the address bar, there are standard browser controls: back, forward, search, and refresh. The main content area of the browser shows the following SVG code:

```
1 <svg xmlns="http://www.w3.org/2000/svg" version="1.1" width="400" height="150" fill="rgb(250, 230, 210)" stroke="black">
2 <rect width="400" height="150" fill="rgb(250, 230, 210)" stroke="black"/>
3 <g stroke="black">
4   <line x1="10" y1="138" x2="20" y2="36" stroke-width="1" />
5   <line x1="20" y1="36" x2="30" y2="44" stroke-width="1" />
6   <line x1="30" y1="44" x2="40" y2="125" stroke-width="1" />
7   <line x1="40" y1="125" x2="50" y2="71" stroke-width="1" />
8   <line x1="50" y1="71" x2="60" y2="32" stroke-width="1" />
9   <line x1="60" y1="32" x2="70" y2="14" stroke-width="1" />
10  <line x1="70" y1="14" x2="80" y2="52" stroke-width="1" />
```

... sólo observamos 10 líneas de 44. Este último código corresponde a la gráfica, el cual, es una gráfica vectorial y su contenido son comandos de líneas, rectángulos, etc., en formato de texto.

No olvide el concepto de HyperText Markup Language (HTML), es decir, Lenguaje de Marcas de Hipertexto el cual, es en el que se basan los navegadores web para construir la página que muestran al usuario.

Ahora se analiza el código:

Se han añadido comentarios que explican cada línea del código y está disponible en: <https://github.com/gpoolb/ESP32> en la carpeta "AdvancedWebServerConComentarios", aún así es relevante explicar cómo funciona el programa a nivel de ejecución de métodos:

### 1er Recurso:

Según la línea 115, se configura al servidor para que ejecute al método llamado handleRoot, el cual, es el encargado de enviar la página web la cual, ya hemos observado anteriormente, adicionalmente, contiene la gráfica. Para ser redundantes, este primer recurso es el que llama el navegador al momento de teclear solamente la dirección ip: <http://esp32.local/> y puede observarlo en el código como sigue:

```

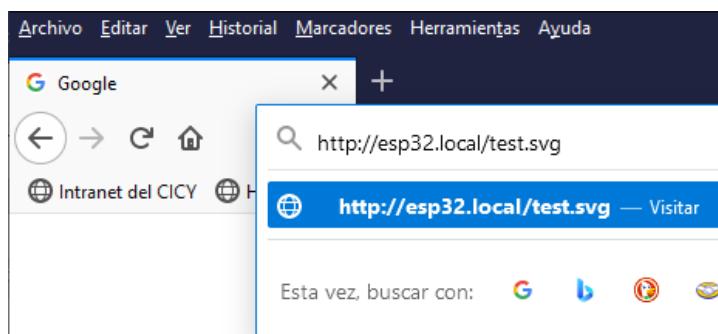
111 server.on("/", handleRoot); // Se establece el nombre de
112 server.on("/test.svg", drawGraph); // Se establece el nombre del recurso
113 server.on("/inline", []() { // Se establece el método a ejecutar
114     server.send(200, "text/plain", "this works as well");
115 });
116 // Termina el método a ejecutar cuando el cliente indica que no
117 server.onNotFound(handleNotFound); // Se establece el nombre del recurso
118 server.begin(); // Se inicia las funciones del servidor

```

En la figura anterior se puede apreciar los recursos disponibles en el servidor, ahí se observan 4 recursos declarados. El primero ya se ha observado y se llama “/”. Cuando se teclea la dirección IP ó el nombre del servidor seguido de “.local” en el navegador WEB, pero... ¿Cómo se hacen las peticiones a los demás recursos?

## 2º Recurso

Ahora en el navegador WEB teclee <http://esp32.local/test.svg> así como se muestra:



... observe el resultado:



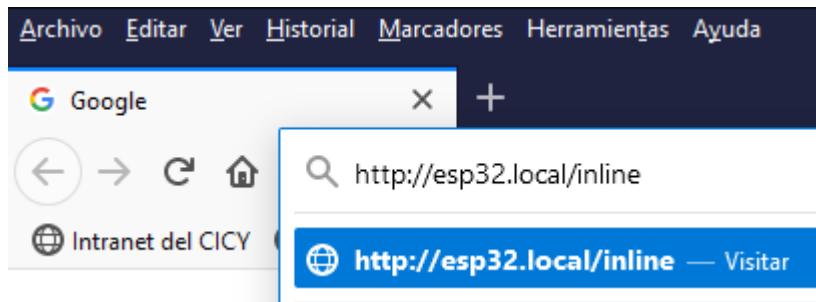
¿Qué ha pasado con la página Web? ¿Por qué luce diferente?, ¿Qué método fue el que se ejecutó al llamar a la gráfica?

```
115 |     server.on("/", handleRoot); // Se establece el nombre de
116 |     server.on("/test.svg", drawGraph); // Se establece el nom
117 |     server.on("/inline", []() { // Se establece el método a
118 |         server.send(200, "text/plain", "this works as well");
119 |     }); // Termina el método a ejecutar cuando el cliente i
120 |     server.onNotFound(handleNotFound); // Se establece el n
121 |     server.begin(); // Se inicia las funciones del servidor
```

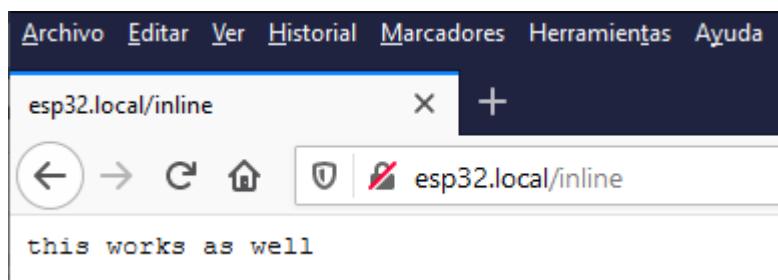
... en la figura anterior, podemos observar la línea 116, la cual, contiene el código que llama al método drawGraph y es el que se encarga de construir la gráfica y enviarlo al cliente solicitante.

### 3er Recurso

Para llamar al tercer recurso, sólo se tiene que teclear en el navegador web el nombre del servidor más el nombre de dominio (.local) seguido de /inline como se muestra a continuación:



... al presionar la tecla de “enter” se observa lo siguiente:



Sólo se aprecia el texto: “this works as well”. Observe con atención la línea 116, donde se declara la función embebida le envía al cliente.

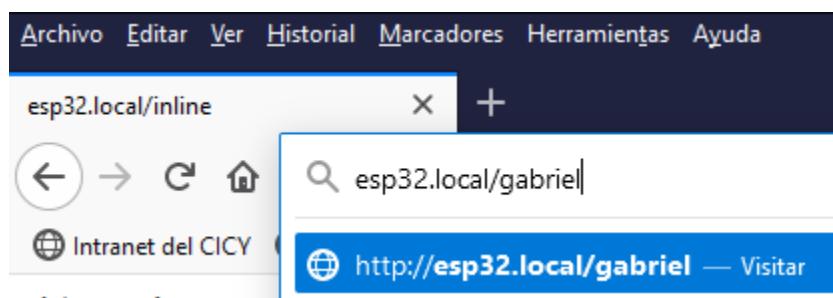
```

115  server.on("/", handleRoot); // Se establece el nombre de
116  server.on("/test.svg", drawGraph); // Se establece el nom
117  server.on("/inline", []() { // Se establece el método a
118      server.send(200, "text/plain", "this works as well");
119  }); // Termina el método a ejecutar cuando el cliente i
120  server.onNotFound(handleNotFound); // Se establece el n
121  server.begin(); // Se inicia las funciones del servidor

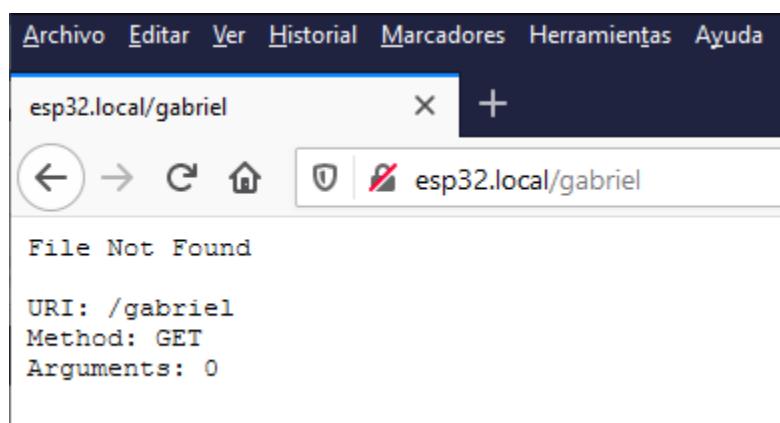
```

## 4o Recurso

Ahora se pretende solicitar al servidor el último recurso, pero no tiene un texto asociado... ¿Cómo se le puede llamar?, ¿qué texto se tiene que utilizar?. El último recurso se llama automáticamente al hacer una petición en la dirección ip del servidor, pero con un texto diferente a los declarados anteriormente. Un ejemplo, ponga la dirección ip correspondiente al servidor seguido de su nombre:



... presione la tecla “enter” y observe lo siguiente:



... puede apreciar que sólo se muestra un texto, y si analizamos la función de la línea 120 se puede apreciar que ahí se invoca al método handleNotFound:

```
115 server.on("/", handleRoot); // Se establece el nombre de
116 server.on("/test.svg", drawGraph); // Se establece el nom
117 server.on("/inline", []() { // Se establece el método a
118     server.send(200, "text/plain", "this works as well");
119 }); // Termina el método a ejecutar cuando el cliente i
120 server.onNotFound(handleNotFound); // Se establece el n
121 server.begin(); // Se inicia las funciones del servidor
```

Este método es relevante conocerlo, ya que tiene implementada la extracción de argumentos que el cliente le envía. Los argumentos son las “variables” que se pretende enviar al servidor para que éste lo procese, este tema se analizará en el punto 3.1.3 de este manual.

## mDNS

En el punto 3.1.1 de este manual, se instaló un software (en Windows y Android) que permitía encontrar en la red local al servidor instalado en su NodeMCU por nombre (no se requiere saber la dirección IP). La sección de código que hace posible esta acción es la que se muestra a continuación:

```
110
111 if (MDNS.begin("esp32")) { // Se configura la instancia para difusión del n
112     Serial.println("MDNS responder started"); // Si la configuración fué un é
113 } // Si la configuración falló no se mostrará lo anterior en el monitor del
114
```

En la línea 111 de la figura anterior se declara el nombre “esp32” el cual, es el nombre que usted elige de manera arbitraria. Elija preferentemente nombres en minúsculas, sin caracteres especiales (\$%&, etc.) y que le sea fácil recordar. El nombre de dominio se declara de manera implícita (.local) y sirve para “crear” un grupo en la red de trabajo y compartir los nombres entre los integrantes del grupo. Es por esa razón que al teclear en su navegador web tiene que usar esta nomenclatura: <http://esp32.local/>

Para que el programa “Service Browser” instalado en Android encuentre a su dispositivo en la red se requiere que agregue la línea al código justo al final del método setup() así como se muestra a continuación.

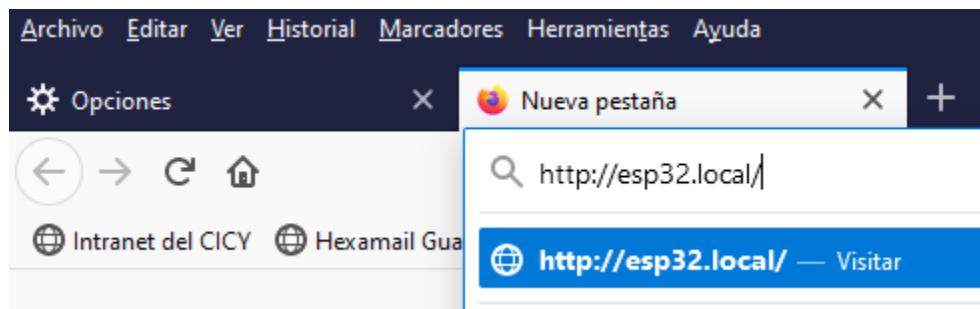
```
138 | Serial.println("HTTP server started"); // Se imprime la leyenda
139 |
140 | MDNS.addService("http", "tcp", 80);
141 |
142 |
```

Compile y descargue el código en su NodeMCU, ahora en su diapositivo Android, abra el programa “Service Browser” y ubique su servidor, siguiendo los pasos que se describen al final del punto 3.1.1.

### 3.1.3 Paso de argumentos al servidor WEB

**Objetivo específico:** Transferirá argumentos entre la página web (cliente) y su NodeMCU (servidor).

Antes de iniciar, se requiere descargar el ejemplo del sitio: <https://github.com/gpoolb/ESP32> en la carpeta “PasoDeArgumentos”. Obtenga del sitio web, compile y descargue el código a su módulo NodeMCU. ES **NECESARIO CERRAR LAS PESTAÑAS, BORRAR DATOS E HISTORIAL** del navegador web para observar el cambio. Una vez borrado el historial, teclee en el navegador web la dirección anterior <http://esp32.local> y obtendrá la misma página del ejemplo anterior con algunos cambios.



... presione la tecla “enter” y observa la página web del ejemplo:



Como podrá observar se han añadido dos botones, uno para encender y otro para apagar el LED incorporado en el NodeMCU. Presione los botones y observe su comportamiento. Ahora...

¿Qué cambios se le hizo al código anterior para lograr esto?

El primer cambio para lograr esto fue agregar los botones en la página web:

```
49 /*****INICIA LA PAGINA WEB *****/\n50 "<html>\n51 <head>\n52   <meta http-equiv='refresh' content='5;URL=/'>\n53   <title>ESP32 Demo</title>\n54   <style>\n55     body { background-color: #cccccc; font-family: Arial, Helvetica, Sans-Serif;\n56   </style>\n57 </head>\n58 <body>\n59   <h1>Hello from ESP32!</h1>\n60   <p>Uptime: %02d:%02d:%02d</p>\n61   <img src=\"/test.svg\" />\n62 </p>\n63   <form method=\"GET\" action=\"/led\" id=\"ledForm\" />\n64   <input type=\"submit\" form=\"ledForm\" name=\"ledBuiltIn\" value=\"on\">\n65   <input type=\"submit\" form=\"ledForm\" name=\"ledBuiltIn\" value=\"off\">\n66 </body>\n67 </html>*\n68 ****TERMINA LA PAGINA WEB *****/
```

... adicionalmente se cambia la declaración “content” para evitar que la página envíe el mismo argumento varias veces.

El segundo cambio, se requiere aumentar el tamaño del buffer de memoria, tanto de la variable “temp” y de la sentencia sprintf, así como se muestra:

```
42\n43 void handleRoot() { // Método que envía la página WEB al client\n44   char temp[800]; // Se define una variable la cual, contendrá\n45   int sec = millis() / 1000; // Se define una variable que almacene el tiempo en milisegundos\n46   int min = sec / 60; // Se define una variable que almacene el minuto\n47   int hr = min / 60; // Se define una variable que almacene el hora\n48   snprintf(temp, 800, \\
```

En tercer lugar, se requiere de construir el método que controla el estado del led incorporado del NodeMCU:

```
76
77 void controlLed (void){ // Metodo que controla el estado del led incorporado
78     for ( uint8_t i = 0; i < server.args(); i++ ) { // Se analiza el número de argumentos
79         Serial.print (server.argName ( i )); // Se imprime en el monitor serial el nombre del argumento
80         Serial.print (": ");
81         Serial.println (server.arg ( i )); // Se imprime al valor del argumento
82         if ( server.argName ( i ).equalsIgnoreCase("ledBuiltIn") ) { // Si el nombre es el establecido
83             if (server.arg ( i ).equalsIgnoreCase("on")) // Si nombre del argumento es "on"
84                 digitalWrite (led, LOW); // Si el nombre y el valor de argumento coinciden
85             else
86                 digitalWrite (led, HIGH); // Si el nombre y el valor de argumento no coinciden
87         }
88     }
89     handleRoot(); // El servidor nos devuelve a la página principal.
90 }
91
92 void handleNotFound() { //Metodo que devuelve al cliente el aviso de que
```

En cuarto lugar, se requiere eliminar la sentencia que está al inicio de los métodos handleRoot y handleNotFound:

```
digitalWrite (led, 1); // Enciende el Led incorporado en su módulo NodeMCU
```

... y la sentencia que está al final de los mismos métodos mencionados anteriormente:

```
digitalWrite (led, 0); // Enciende el Led incorporado en su módulo NodeMCU
```

... ya que interfieren con el propósito propuesto.

Por último, se tiene que declarar el nuevo recurso en la sección de configuración del servidor:

```
132
133     server.on("/", handleRoot); // Se establece el nombre de la página principal
134     server.on("/test.svg", drawGraph); // Se establece el nombre de la página de gráficos
135     server.on("/inline", []() { // Se establece el método a ejecutar
136         server.send(200, "text/plain", "this works as well");
137     }); // Termina el método a ejecutar cuando el cliente indica
138     server.on("/led", controlLed); // Se añade un nuevo recurso
139     server.onNotFound(handleNotFound); // Se establece el nombre de la página de error
```

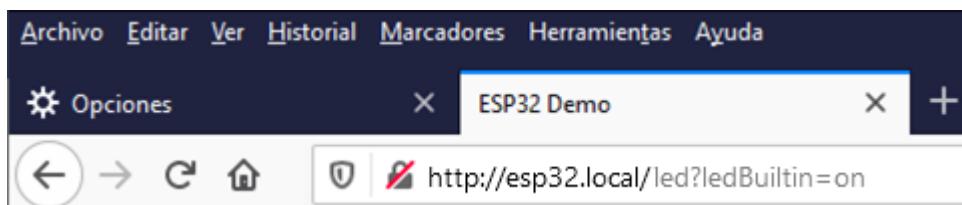
**NO OLVIDE QUE** para que el programa “Service Browser” instalado en Android encuentre a su dispositivo en la red se requiere que agregue la línea al código justo al final del método setup() así como se muestra a continuación.

```
MDNS.addService("http", "tcp", 80);
} // fin del método setup()
```

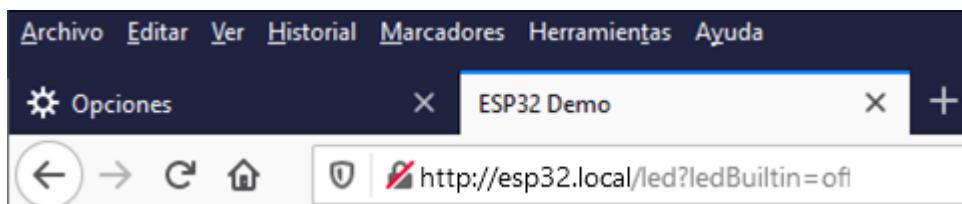
Compile y descargue el código en su NodeMCU, ahora en su dispositivo Android, abra el programa “Service Browser” y ubique su servidor, siguiendo los pasos que se describen al final del punto 3.1.1.

Ahora... ¿cómo funciona?, en el momento de presionar cualquier botón (“on” o “off”) observe con atención la barra de dirección del navegador WEB:

Para el botón on:



Para el botón off:



¿qué significan los caracteres que le siguen al nombre de servidor y nombre de dominio?

Ya se sabe que posterior al nombre de servidor y nombre de dominio sigue el recurso a solicitar, en este caso es: “/led”, eso le indica al servidor qué método va a ejecutar.

Ahora... ¿Qué significan los caracteres que siguen posterior al símbolo de interrogación?

El símbolo de interrogación es utilizado para separar el recurso de los argumentos. Los argumentos son las variables que enviamos al servidor para que las procese. Su notación es:

?nombre1=valor1, nombre2=valor2

En este caso es:

?ledBuiltIn=on para encender el led y,  
?ledBuiltIn=off para apagar el led

¿Cuántos argumentos se puede enviar?

Para el método GET que es el que se está utilizando se puede enviar un máximo de un paquete de 2048 caracteres, o sea, si los nombres y valores de los argumentos son largos, reducirá el número de argumentos a enviar.

¿Es posible enviar más argumentos?

Si, es posible enviar más argumentos, pero se requiere usar el método POST ya que los argumentos son enviados en paquetes separados de la dirección ip y el recurso solicitado. Es decir, en el primer paquete se envía la dirección y el recurso solicitado, cuando el servidor lo reciba, inmediatamente se le envían los argumentos, y éstos pueden ser tantos como se requiera.

Actividades:

¿Podría implementar el método POST en ambos botones?

Sugerencia:

En la sección de la página web cambie la forma en la que envía la petición los botones on y off:

```
<form method="POST\" action="/led\" id="ledForm\" />\n<input type="submit" form="ledForm" name="ledBuiltIn" value="on">\n<input type="submit" form="ledForm" name="ledBuiltIn" value="off">\n
```

Posteriormente, declare en la configuración del servidor, el método a utilizar:

```
server.on("/led", HTTP_POST, controlLed); // Se añade un nuevo recurso para controlar el led incorporado del NodeMCU
```

¿Podría implementar los botones para controlar el Led RGB descrito en el punto 2.2.1?

Sugerencias:

Incremente el valor del buffer y del espacio de la instrucción sprintf:

```
char temp[1050];
```

```
snprintf(temp, 1050, \
```

Implemente los botones en la página web:

```
<p></p>\n<input type=\"submit\" form=\"ledForm\" name=\"ledR\" value=\"on\">\n<input type=\"submit\" form=\"ledForm\" name=\"ledR\" value=\"off\">\n<p></p>\n<input type=\"submit\" form=\"ledForm\" name=\"ledG\" value=\"on\">\n<input type=\"submit\" form=\"ledForm\" name=\"ledG\" value=\"off\">\n<p></p>\n<input type=\"submit\" form=\"ledForm\" name=\"ledB\" value=\"on\">\n<input type=\"submit\" form=\"ledForm\" name=\"ledB\" value=\"off\">\n\n
```

Declare las variables al inicio del código que correspondan a los pines asignado al led RGB

```
const int led = LED_BUILTIN;  
const int ledR = 12;  
const int ledG = 14;  
const int ledB = 27;
```

En el método setup() configure los pines de los led's como salida:

```
pinMode(ledR, OUTPUT);  
digitalWrite(ledR, 1); // Se apaga el led incorporado en su módulo NodeMCU  
pinMode(ledG, OUTPUT);  
digitalWrite(ledG, 1); // Se apaga el led incorporado en su módulo NodeMCU  
pinMode(ledB, OUTPUT);  
digitalWrite(ledB, 1); // Se apaga el led incorporado en su módulo NodeMCU
```

Posteriormente, implemente el código para controlar el estado de los tres colores de led:

```
if ( server.argName ( i ).equalsIgnoreCase("ledBuiltIn") ) {  
    if (server.arg ( i ).equalsIgnoreCase("on"))  
        digitalWrite(led, LOW);  
    else  
        digitalWrite(led, HIGH);  
} else if ( server.argName ( i ).equalsIgnoreCase("ledR") ) {  
    if (server.arg ( i ).equalsIgnoreCase("on"))  
        digitalWrite(ledR, LOW);  
    else  
        digitalWrite(ledR, HIGH);  
} else if ( server.argName ( i ).equalsIgnoreCase("ledG") ) {  
    if (server.arg ( i ).equalsIgnoreCase("on"))  
        digitalWrite(ledG, LOW);  
    else  
        digitalWrite(ledG, HIGH);  
} else if ( server.argName ( i ).equalsIgnoreCase("ledB") ) {  
    if (server.arg ( i ).equalsIgnoreCase("on"))  
        digitalWrite(ledB, LOW);  
    else  
        digitalWrite(ledB, HIGH);  
}
```

## 3.2 El Servidor Web con página dinámica en NodeMCU-32S

En este punto se desarrollará una página web dinámica la cual, mostrará el valor del sensor de temperatura y podrá manipular el estado de un led desde la página web.

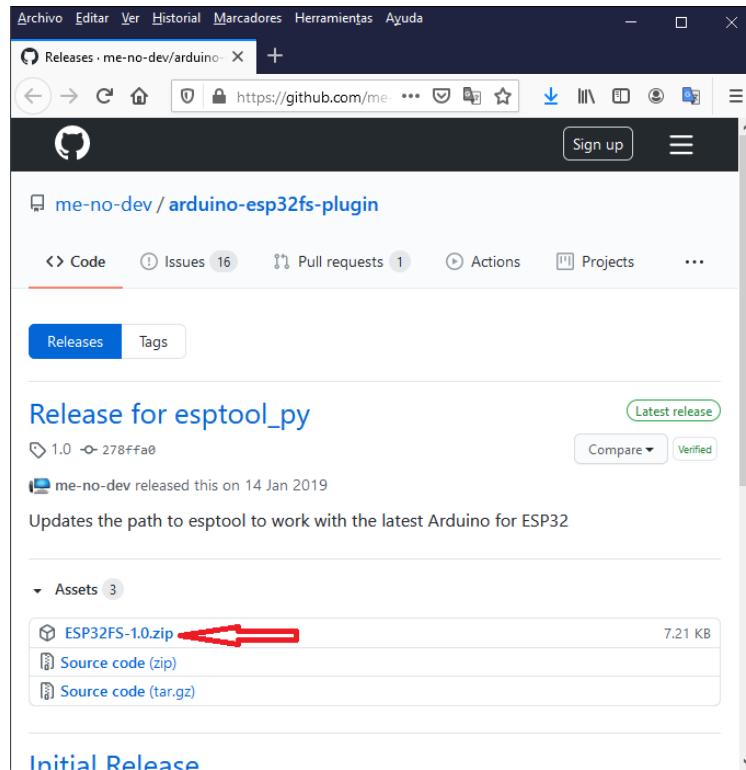
### 3.2.1 ESP32 Filesystem Uploader Plugin

**Objetivo específico:** Instalará la herramienta que permite subir archivos y carpetas en la memoria Flash del ESP32.

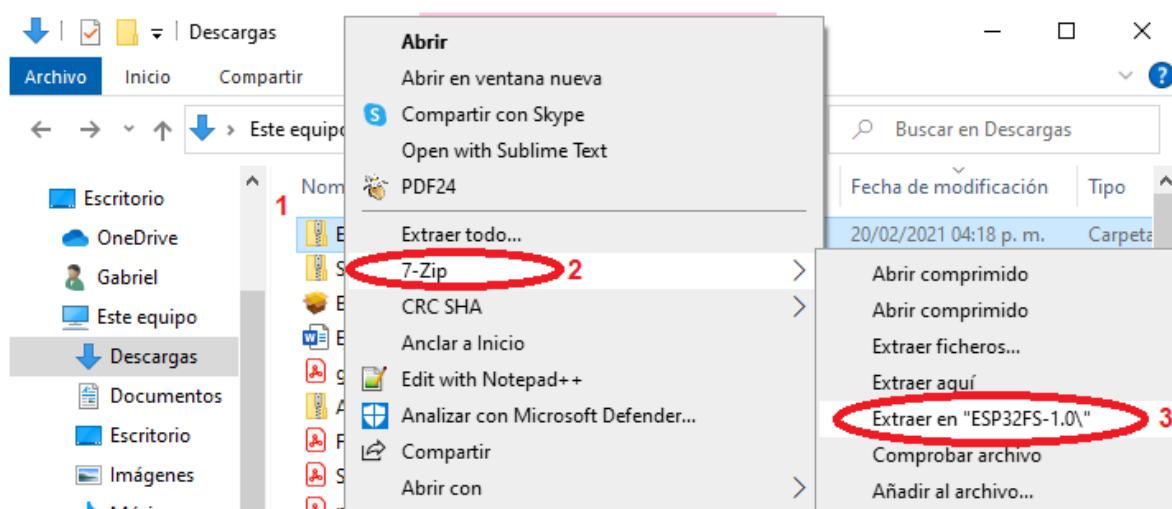
El ESP32 contiene un Sistema de Archivos en la memoria Flash Interna con Interfaz Serial (SPIFFS). SPIFFS es un sistema de archivos muy ligero creado para microcontroladores con una memoria flash. Usted puede leer, escribir, cerrar y borrar archivos. En la literatura menciona que no soporta directorios, pero se ha probado introducir archivos en una carpeta y ha funcionado. Adicionalmente, tiene la limitante de utilizar nombre de 8 caracteres como máximo y tres caracteres como extensión.

En el sitio <https://randomnerdtutorials.com/install-esp32-filesystem-uploader-arduino-ide/> se describen los pasos para la instalación del plugin, según lo anterior, se requiere seguir los siguientes pasos:

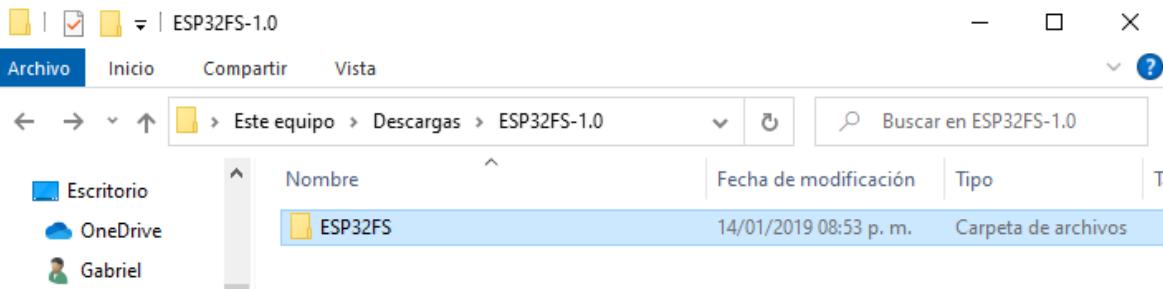
- 1) Obtenga el plugin del sitio <https://github.com/me-no-dev/arduino-esp32fs-plugin/releases/> y descargue el archivo con terminación .zip así como se muestra:



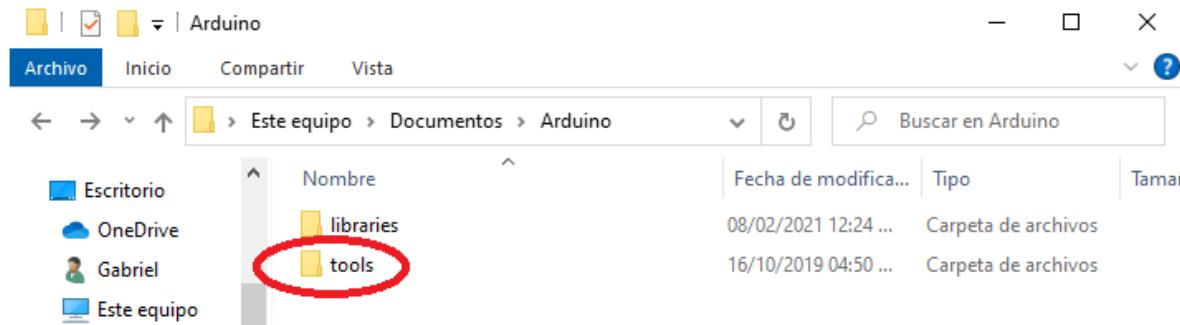
**2)** Al dar un click con el mouse en el sitio marcado en la figura anterior, iniciará la descarga. Ahora, con el archivo descargado, se requiere descomprimirlo para esto, dé un click con el botón derecho del mouse sobre el archivo recién descargado y siga la secuencia en el menú que sale a continuación, así como se muestra (se requiere tener instalado el programa 7zip vista en el punto 3.1.1):



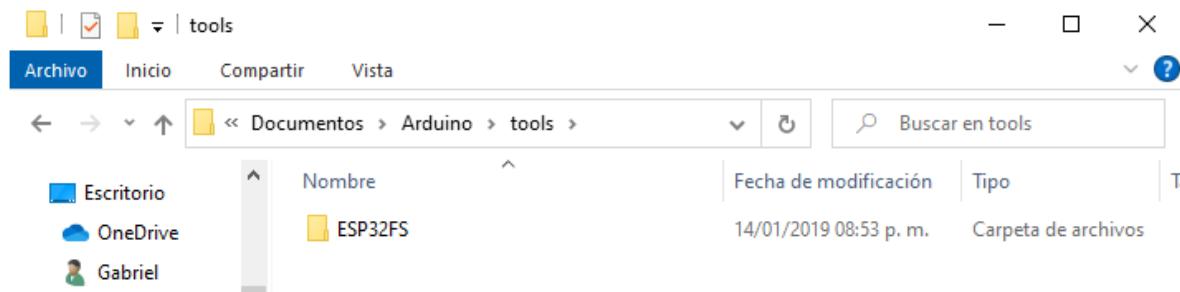
3) Al terminar de descomprimir, se genera una nueva carpeta llamada “ESP32FS-1.0”, entre a esa carpeta y verá otra llamada “ESP32FS”



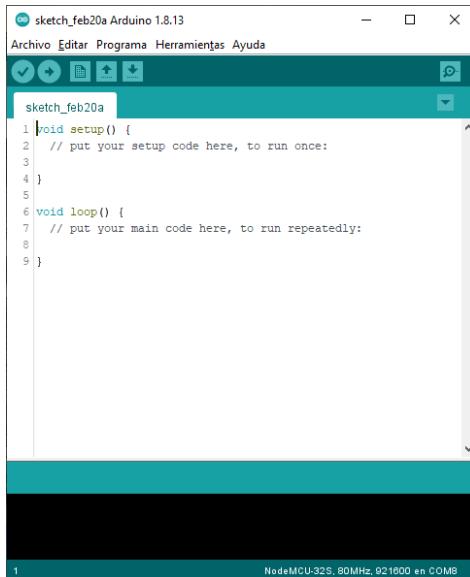
4) Ahora abra OTRA ventana del explorador de archivos, ubíquese en Mis Documentos, entre a la carpeta de Arduino y en el interior de la carpeta Arduino cree otra carpeta llamada “tools” así como se muestra:



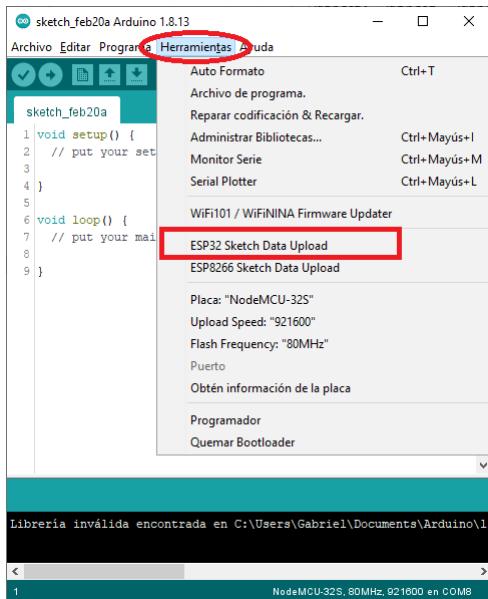
5) Entre a la carpeta “tools” recién creada y copie la carpeta ESP32FS (vista en el punto 3) así como se muestra:



**6)** Cierre ambas ventanas del explorador y abra el IDE del Arduino (Si lo tenía abierto, cierre el IDE del Arduino y vuelva a abrirlo):



**7)** Al presionar el menú de herramientas en el IDE del Arduino, podrá observar que ahora se tiene una nueva opción llamada “ESP32 Sketch Data Upload”, el cual, es el que se utilizará para la descarga de archivos al NodeMCU.

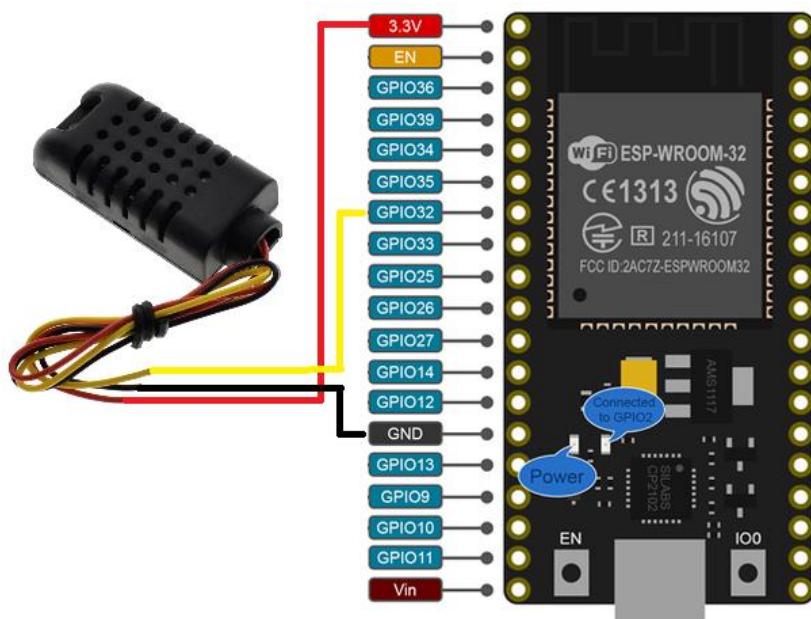


Por el momento NO presione esa opción, ya que, se requiere de la creación de la carpeta “data” en el proyecto, que, contiene los archivos que serán transferidos al NodeMCU. Esta opción se utilizará en el proyecto que se verá más adelante.

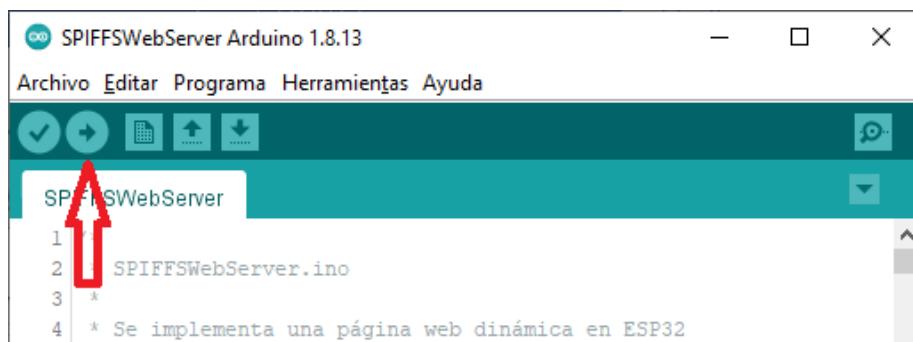
### 3.2.2 SPIFFS WebServer

**Objetivo específico:** Implementará una página web dinámica en el módulo ESP32.

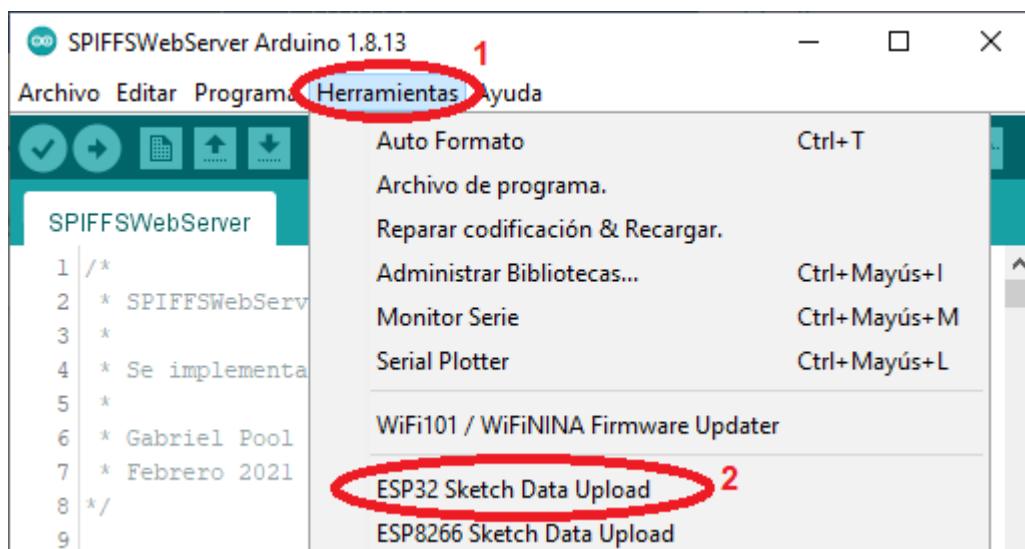
Antes de iniciar, es relevante comentar que, en el punto 2.2.6 de este manual, se implementó un proyecto donde se utiliza el sensor DHT 22:



Para implementar este proyecto, se requiere que ya se hayan descargado las bibliotecas necesarias para que el DHT22 funcione, adicionalmente, se requiere conectar el DHT22 en el pin 32 de su NodeMCU-32S. Seguidamente, se requiere descargar el ejemplo del sitio: <https://github.com/gpoolb/ESP32> en la carpeta "SPIFFSWebServer". Obtenga del sitio web, compile y descargue el código a su módulo NodeMCU de la misma manera que se ha manejado en los ejemplos anteriores:



Al terminar de descargar su código anterior, es necesario, ejecutar un paso adicional, utilizando la herramienta instalada en el punto 3.2.1, se requiere descargar en el módulo NodeMCU los archivos incluidos en la carpeta “data” que se encuentra en la carpeta del proyecto, así como se muestra:



... y en la parte inferior del IDE mostrará lo siguiente:

```

18
19 #include <time.h>          // Biblioteca requerida para manejo de tiempos
<>

SPIFFS Uploading Image...
Writing at 0x00290000... (50 %)
Writing at 0x00294000... (100 %)

<>
8                                         NodeMCU-32S en COM6

```

Espere a que la barra azul muestre lo siguiente:

```

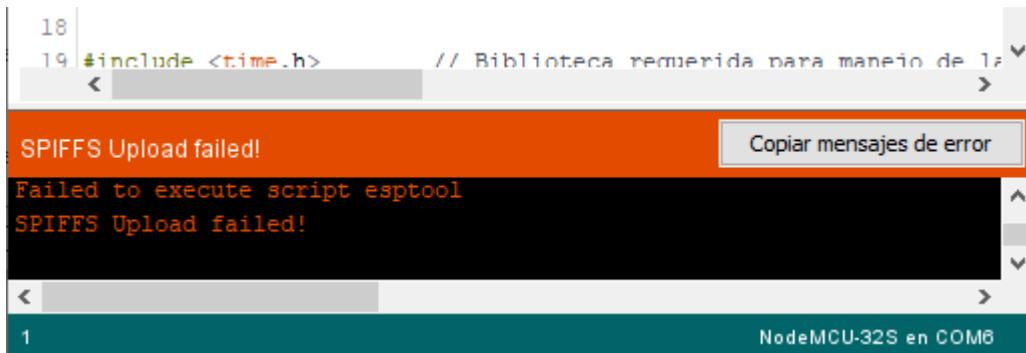
18
19 #include <time.h>          // Biblioteca requerida para manejo de tiempos
<>

SPIFFS Image Uploaded
Leaving...
Hard resetting via RTS pin...

<>
8                                         NodeMCU-32S en COM6

```

Si al momento de utilizar esta herramienta, muestra el siguiente error:

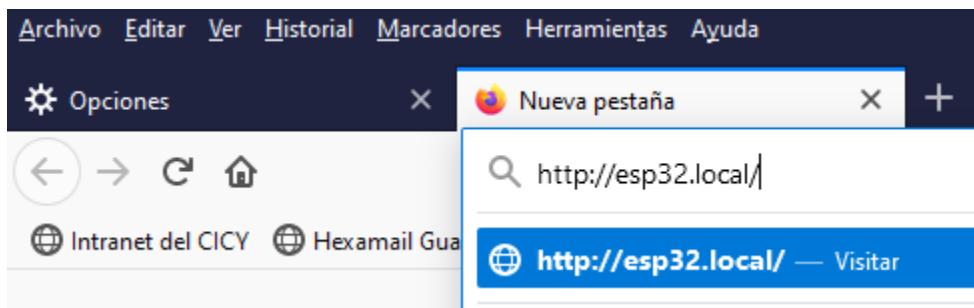


A screenshot of a serial monitor window titled "NodeMCU-32S en COM6". The window shows a code editor at the top with lines 18 and 19 of a C program. Below it is a text area with the following error message:  
SPIFFS Upload failed!  
Failed to execute script esptool  
SPIFFS Upload failed!

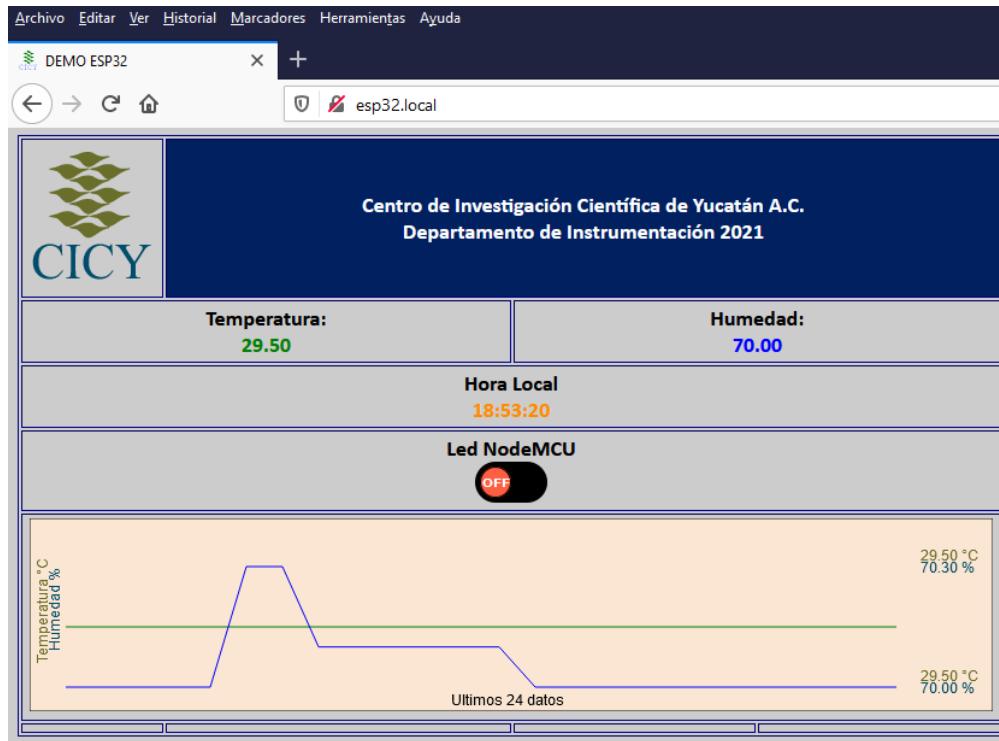
... se debe a que tiene abierto la ventana del monitor serial. Es necesario cerrar dicha ventana para que la herramienta pueda transferir los archivos al módulo NodeMCU.

El contenido guardado en la memoria flash por esta herramienta (ESP32 Data Upload), es independiente del código de programa que se compila y se descarga en el módulo NodeMCU, es decir, no es necesario compilar y descargar de nuevo el código de programa al módulo NodeMCU cuando la herramienta sea utilizada (ESP32 Data Upload).

Una vez descargados ambos (código y archivos) en el NodeMCU, **ES NECESARIO CERRAR LAS PESTAÑAS, BORRAR DATOS E HISTORIAL** del navegador web para observar el cambio. Una vez borrado el historial, teclee en el navegador web la dirección anterior <http://esp32.local>, cabe aclarar que se está considerando que usa Windows en su PC y tiene instalado el Bonjour descrito en el punto 3.1.1, para Android use el programa Service Browser que también se describe en el punto 3.1.1:



... presione la tecla “enter” y observa la página web del ejemplo:



Como podrá observar de manera más relevante: el botón del LED cambió su apariencia, la gráfica se desplaza horizontalmente y se ha añadido un logo. Presione el botón y observe su comportamiento. Los textos y la gráfica se han vuelto dinámicos. Ahora... ¿Qué cambios se le hizo al código anterior para lograr esto?

Todo el código de programa está comentado, así que analizaremos por secciones las partes más relevantes:

```

169
170 Serial.println("Contacting Time Server");
171 configTime(3600*timezone, daysavetime*3600, "time.nist.gov", "0.
172     delay(2000);
173     tmstruct.tm_year = 0;
174     getLocalTime(&tmstruct, 5000); // Se obtiene el tiempo actual
175     Serial.printf("\nNow is : %d-%02d-%02d %02d:%02d:%02d\r\n", (tmstru
176     Serial.println("");
177
178     if(!SPIFFS.begin()){
179         Serial.println("Card Mount Failed");
180         return;
181     }
182

```

En el método `setup()`, está declarado la configuración para la obtención de la hora y fecha, es posible tener la hora y fecha actual desde un servidor que se

especializa en proporcionar ése servicio. Adicionalmente, se configura al módulo ESP32 para tener acceso al sistema de archivos en su memoria flash (es equivalente a tener una microSD para almacenar archivos, aunque de capacidad muy limitada).



```

SPIFFSWebServer
197 //Se inicializa el response (contiene los últimos 24 valores del
198 int apuntador = 0;
199 sprintf(&responseDataGraphic[apuntador], 1800,"<response>\n");
200 apuntador += 11; // Se apunta al inicio de la segunda linea
201 for(int i = 0; i < 24; i++){
202     sprintf(&responseDataGraphic[apuntador], 1800, \
203             /***** INICIA EL FORMATO XML *****
204             "\t<TEMP%02d>-N/A-</TEMP%02d>\n" \
205             "\t<HUM%02d>-N/A-</HUM%02d>\n" \
206             "\t<TIME%02d>HH:MM:SS</TIME%02d>\n",
207             /***** TERMINA EL FORMATO XML *****
208             i, i, \
209             i, i, \
210             i, i
211         );
212     apuntador += 73; // Se apunta al inicio de la linea de la siq
213 }
214 sprintf(&responseDataGraphic[apuntador], 1800,"</response>\n");
215 }

```

En esta sección (Lin 197) se construye un archivo xml (Extensible Markup Language) los cuales se componen de etiquetas en este caso están de forma **individual**. La estructura que se genera es ésta:

```

<response>
    <TEMP00>XX.XX</TEMP00>
    <HUM00>XX.XX</HUM00>
    <TIME00>HH:MM:SS</TIME00>
    <TEMP01>XX.XX</TEMP01>
    <HUM01>XX.XX</HUM01>
    <TIME01>HH:MM:SS</TIME01>
    .
    .
    .
    <TEMP23>XX.XX</TEMP23>
    <HUM23>XX.XX</HUM23>
    <TIME23>HH:MM:SS</TIME23>
</response>

```

Este archivo se utiliza para enviar los datos de la gráfica al cliente, contiene 24 series de 3 datos: Humedad, Temperatura y el Tiempo con formato HH:MM:SS. Observe que tienen etiquetas de apertura y de cierre, en medio de estos se encapsula la información a enviar. El cliente al recibirla, por el nombre del tag ya sabe donde asignar el valor en el lugar que le corresponde.

```
222 void sendDataGraph() { // Método que dibuja una gráfica tipo Scal  
223  
224 server.send(200, "text/xml", responseDataGraphic); // Sentencia  
225 //Serial.println("Response: ");  
226 //Serial.println(responseDataGraphic);
```

Es importante declarar al momento del envío el tipo de archivo, ya que el navegador estará esperando un archivo con formato xml (el cual se declara en el ajax.js, específicamente, el método pollAJAX())

```
230  
231 void leeSensor (void) {  
232
```

En la método leeSensor(), se encarga de leer los datos del sensor e ir insertando los datos en el archivo xml cada segundo. Cada vez que se lee un nuevo dato, este será insertado en la primera posición (00) y el dato que ocupaba la posición '00' es desplazado a la posición '01' y así sucesivamente hasta llegar a la posición '23', el último dato (que ocupaba la posición '23') es desecharo.

```
52 void sendDataTemp() { // Método que envía la página WEB al cliente  
53     char temp[150]; // Se define una variable la cual, contendrá  
54     //get time  
55     getLocalTime(&tmstruct, 0); // Se obtiene el tiempo local almacenado  
56  
57     sprintf(temp, 150, \  
58             /***** INICIA EL FORMATO XML ****/  
59             "<response>\n"\\  
60             "\t<TEMP>%s</TEMP>\n"\\  
61             "\t<HUM>%s</HUM>\n"\\  
62             "\t<ledBuiltIn>%s</ledBuiltIn>\n"\\  
63             "\t<TIME>%02d:%02d:%02d</TIME>\n"\\  
64             "</response>\n",  
65             /***** TERMINA EL FORMATO XML ****/  
66             /***** INICIA LISTA DE VARIABLES A SER ENVIADAS ****/  
67             charTemp, \  
68             charHum, \  
69             estadoLed, \  
70             tmstruct.tm_hour, \  
71             tmstruct.tm_min, \  
72             tmstruct.tm_sec // Aquí se calcula el tiempo en que inicia la medición  
73             /***** TERMINA INICIA LISTA DE VARIABLES A SER ENVIADAS ****/  
74         );  
75         server.send(200, "text/xml", temp); // Sentencia que envía la página WEB al cliente que lo solicita  
76 } // Termina método que envía la página WEB al cliente que lo solicita
```

En la rutina sendDataTemp(), se encarga de construir otro archivo xml más pequeño que el anterior, éste se utiliza para enviar los valores instantáneos del sensor (mas recientes), la hora y el estado del LED.

```
<response>
    <TEMP>XX.XX</TEMP>
    <HUM>XX.XX</HUM>
    <ledBuiltIn>false</ledBuiltIn>
    <TIME>HH:MM:SS</TIME>
</response>
```

... al igual que en el archivo anterior, es importante especificar el tipo de archivo que se envía al cliente (ver línea 76 de la imagen anterior) ya que el mismo método pollAJAX() se encarga de recibirlo.

```
78 void controlled (void){ // Método que controla el estado del led }
```

La pregunta que surge es... ¿porqué enviar dos archivos xml cuando se pueden enviar todos en un solo archivo? La respuesta es porque los valores de la gráfica pueden ser actualizados en tiempos diferentes de acuerdo con el tiempo que decida el programador.

El método controlLed() ya se había analizado y se encarga de controlar el led de acuerdo a los parámetros recibidos, aquí existe un cambio con respecto al ejemplo anterior, el valor del parámetro ha cambiado por 'true' o 'false' ya que se envía el valor del checkbox declarado en el archivo index.htm

```
187 server.on("/dataGraphics.xml", HTTP_GET, sendDataGraph); // Se
188 server.on("/temp.xml", HTTP_GET, sendDataTemp );
189 server.on("/led", HTTP_GET, controlled); // Se añade un nuevo r
190 server.onNotFound(handleNotFound); // Se establece el nombre de
191 server.begin(); // Se inicia las funciones del servidor.
```

Los recursos declarados en el servidor son sólo cuatro:

1. El recurso para obtener los valores de la gráfica
2. El recurso para obtener los valores instantáneos de la temperatura, humedad, hora y estado del Led.
3. El recurso que controla el estado del led
4. Por último, las peticiones que no se han declarado pasan por este punto.

Pregunta: ¿Dónde se lee la página web y sus dependencias?

```
134 void handleNotFound() { //Método que devuelve al cliente el aviso
135     if(loadFromSdCard(server.uri())) return;
136     String message = "File Not Found\n\n"; // Definición de la vari
```

Todas las peticiones que no se han declarado pasan obligatoriamente en el cuarto recurso (`handleNotFound()`), en dicho método, en la línea 135 se declara un método llamado `loadFromSDCard()` al cual se le proporciona como parámetro el recurso solicitado por el cliente. Así que, el programa salta nuevamente a la ejecución de la línea 98:

```
98 bool loadFromSdCard(String path) {
99     String dataType = "text/plain";
100    if(path.endsWith("/")) path += "index.htm";
101
102    if(path.endsWith(".src")) path = path.substring(0, path.lastIndexOf('/'));
103    else if(path.endsWith(".htm")) dataType = "text/html";
104    else if(path.endsWith(".css")) dataType = "text/css";
105    else if(path.endsWith(".js")) dataType = "application/javascript";
106    else if(path.endsWith(".png")) dataType = "image/png";
107    else if(path.endsWith(".gif")) dataType = "image/gif";
108    else if(path.endsWith(".jpg")) dataType = "image/jpeg";
109    else if(path.endsWith(".ico")) dataType = "image/x-icon";
110    else if(path.endsWith(".xml")) dataType = "text/xml";
111    else if(path.endsWith(".pdf")) dataType = "application/pdf";
112    else if(path.endsWith(".zip")) dataType = "application/zip";
113    else if(path.endsWith(".svg")) dataType = "image/svg+xml";
```

En dicho método (extraído del ejemplo `SDWebServer`) se compara la extensión para que le sea asignado el tipo de archivo a enviar al cliente (esto es necesario, recuerde que el explorador web NO aceptará otro tipo de archivo diferente al que espera), posteriormente, en la línea 118 se gestiona en el sistema de archivos (`SPIFFS`) el recurso solicitado por el cliente:

```
118 File dataFile = SPIFFS.open(path.c_str(), "r");
119
120 if (!dataFile)
121     return false;
122
123 if (server.hasArg("download")) dataType = "application/octet-stream";
124
125 if (server.streamFile(dataFile, dataType) != dataFile.size()) {
126     Serial.println("Sent less data than expected!");
127 }
128
129 dataFile.close();
130 return true;
131 }
```

Recuerde que, hasta este punto, sólo llegan los recursos que no se declararon anteriormente, así que, se esperan TODOS los nombres de archivos que se han guardado en el `SPIFFS` con la herramienta (ESP32 Data Upload), en este caso

son:

index.htm  
ajax.js  
ajax2.js  
logo.svg  
test.svg  
favicon.ico

Si llegara otro diferente a los anteriores, el método terminará en la línea 121 e indicará que no encontró el recurso solicitado con un valor ‘false’.

Si el recurso es encontrado, será enviado al cliente en la línea 125, el sistema de archivos será cerrado en la línea 129 y el método finalizará devolviendo un valor ‘true’ que indica que la respuesta fue enviada exitosamente.

Hasta este punto, se ha descrito el funcionamiento del servidor web por parte del módulo NodeMCU ESP32, ahora se describirá a nivel de métodos cómo funciona en la parte del cliente (navegador WEB).

### El cliente

Al momento de teclear la dirección IP ó el nombre MDNS, el navegador le envía el carácter ‘/’ al servidor, en respuesta, el servidor envía el archivo index.htm. Típicamente, el primer archivo que el navegador recibe de un servidor es el index.html, así que, el ESP32 se apega al estándar. Al recibir el index.htm del NodeMCU, el navegador lo analiza e inmediatamente, gestiona las dependencias con el servidor, en este caso son los demás archivos (ya mencionados anteriormente), el cual, cuando el servidor recibe las peticiones envía como respuesta los archivos solicitados.

En el momento de recibir todas las dependencias, el navegador web (cliente) inicia la construcción de la página web que muestra al usuario e inmediatamente empieza la ejecución de los códigos en lenguaje JavaScript (archivos con extensión .js) los cuales, no son visibles al usuario.

Si abre con el editor de texto (preferentemente Notepad++) el archivo index.htm, se puede identificar los códigos JavaScript por estar entre los siguientes tags:

```
<script src="*.js" TYPE="text/javascript" ></script>
```

Las imágenes que se utilizan en este proyecto están en formato svg y se identifican por estar entre los siguientes tags:

```
<iframe id="myImage" type="image/svg+xml" src = "*.svg" frameborder= "0" height=" " width=" " >
```

```
</iframe>
```

... los cuales están embebidos en una tabla.

Los estilos declarados para la página web y el botón (on-off) se delimitan por los siguientes tags:

```
<style> </style>
```

La página web fue construida de la manera más sencilla posible para poder agregar más ítems. Por ejemplo, si desea agregar un texto, tiene que declararlo en la página web como:

```
<text id="miText"></text>
```

Y desde el servidor, agregue en el response un tag de la siguiente manera:

```
"\t<miText>Hola Mundo</miText>\n"\
```

... cuando el navegador lo reciba, asignará el texto “Hola Mundo” en el sitio donde declaró los tags.

Se construyó un método para gestionar recursos desde la página web, sólo requiere dos parámetros:

```
EnviaDatos(recurso,parámetros);
```

Si quisiera enviar la instrucción de apagar led sería de la siguiente manera:

```
EnviaDatos('/led?','ledBuiltin=false');
```

En este caso, se requería enviar el estado del checkbox, así que se utilizan las propiedades de éste para poder enviar su nombre y su estado:

```
onclick="EnviaDatos('./led?', this.id + '=' + this.checked)"
```

... el cual:

this.id es el equivalente a leer el nombre asignado en la propiedad id = “ledBuiltin”  
this.checked es el equivalente a leer el estado del checkbox: ‘true’ o ‘false’

... en pocas palabras, esta última declaración podría ser usado para cualquier checkbox, lo único que tendría que cuidar es que el nombre asignado en la propiedad ‘Id’ sea diferente.

Si desea que el botón indique si el led esté encendido o apagado (o siendo manipulado por otro cliente), es necesario agregar en el response que envía el

servidor lo siguiente:

```
"\t<ledBuiltIn>true</ledBuiltIn>\n"\n0\n"\t<ledBuiltIn>false</ledBuiltIn>\n"\n
```

... el cual indica si el led está encendido ó apagado en el servidor. En este caso particular, el estado del led está almacenado en una variable del tipo string llamada: estadoLed y al usar %s en el código del NodeMCU se indica que lea la variable como una del tipo string (estudie la sentencia sprintf y snprintf usada en la línea 59 del código en el NodeMCU).

Se agregan comentarios a todos los archivos que dependen de la página web de este proyecto para su mejor comprensión. Considere estudiar lenguaje HTML, JavaScript y CSS. Existen alternativas interesantes como el Bootstrap para una mejor presentación en la página web.

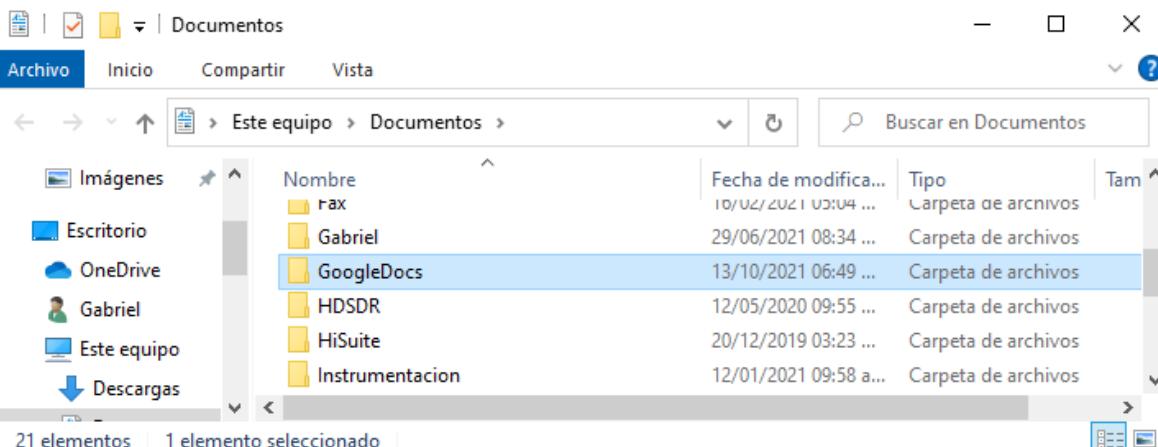
### 3.3 Servidor web implementado en Google Sheets

En este punto se implementará un script el cual, guardará el valor del sensor de temperatura y humedad en un documento de Google.

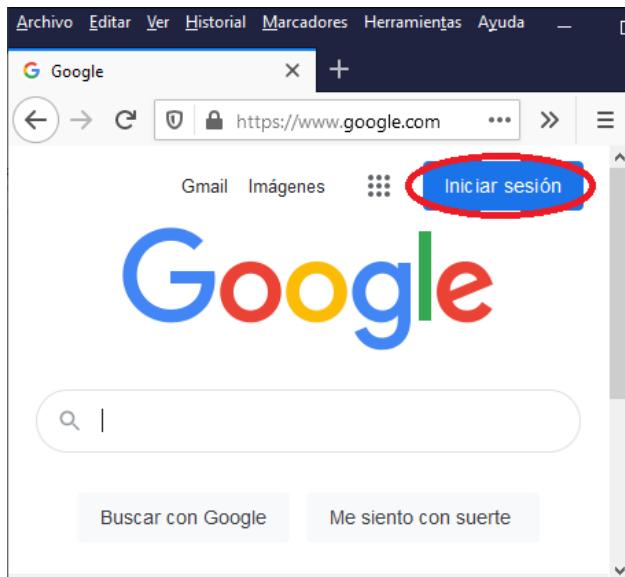
#### 3.3.1 Simple Google Apps Script

**Objetivo específico:** Configurará e implementará los servicios necesarios para guardar la información enviada por el dispositivo IoT en un documento de Google.

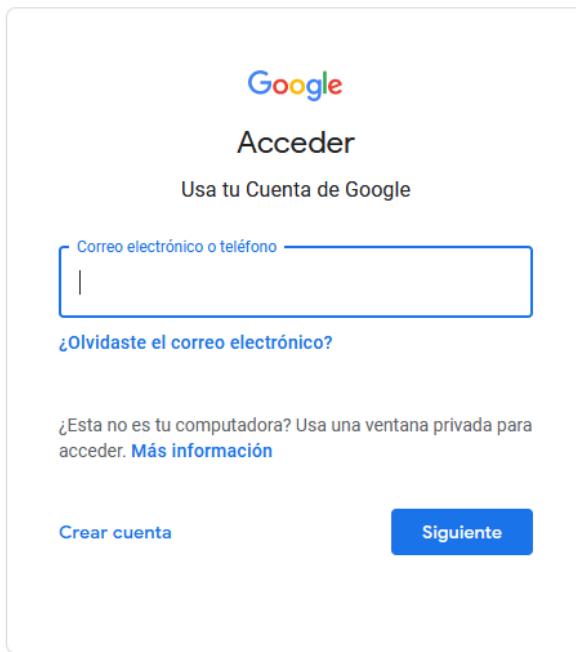
En la actualidad, las herramientas de Google han ganado popularidad y con el crecimiento de los dispositivos IoT, Google no se queda atrás, ha implementado una herramienta llamada Google Apps Script el cual, está basado en el lenguaje JavaScript. Dicho lenguaje es de alto nivel y permite la implementación rápida de soluciones. El JavaScript es ampliamente usado en las páginas web para gestionar servicios en segundo plano. En el sitio <https://developers.google.com/apps-script> se puede encontrar la ayuda necesaria para la implementación de soluciones, este proyecto se enfocará a las WebApps y puede encontrar la documentación en este sitio: <https://developers.google.com/apps-script/guides/web>. Como dice un dicho muy popular “Para muestra basta un botón”. Para implementar este proyecto se requiere de la descarga de los archivos del proyecto ubicado en <https://github.com/gpoolb/ESP32>. Descargue los archivos ubique la carpeta llamada “GoogleDocs” y copie la carpeta en Mis Documentos, así como se muestra:



Seguidamente, se parte del hecho que Ud ya tiene una cuenta en Google. En la página principal de Google, presione el botón “Iniciar sesión”:



Introduzca su nombre de usuario:



The image shows the Google sign-in page. It features the Google logo at the top, followed by the text "Acceder" (Sign in) and "Usa tu Cuenta de Google". Below this is a large input field labeled "Correo electrónico o teléfono" (Email or phone number) with a placeholder " ". To the right of the input field is a blue "Siguiente" (Next) button. At the bottom left is a "Crear cuenta" (Create account) link. A link "¿Olvidaste el correo electrónico?" (Forgot email address?) is also present. At the very bottom, there are links for "Español (Latinoamérica)", "Ayuda", "Privacidad", and "Condiciones".

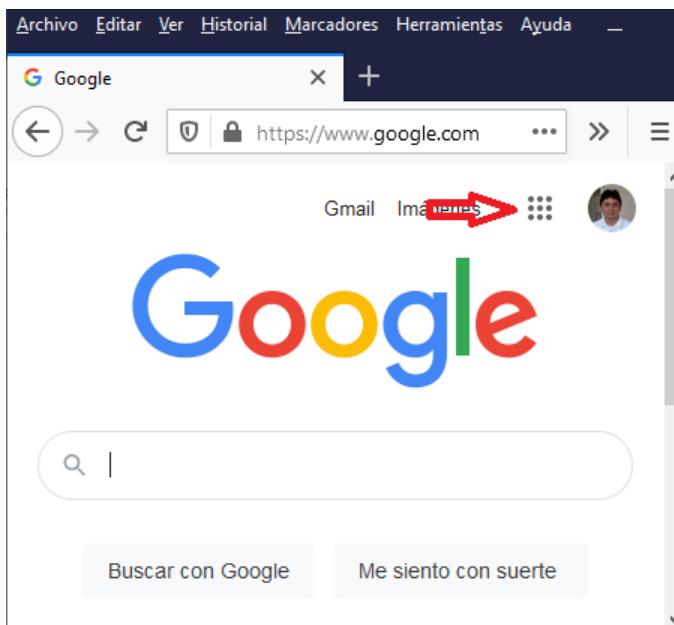
y contraseña:



The image shows a Google sign-in form. At the top, it says "Google" and "Gabriel Pool". Below that is a dropdown menu showing "gabriel.jesus.pool@gmail.com". There is a large input field for "Ingresa tu contraseña" (Enter your password) with a placeholder "I". Below the password field is a checkbox labeled "Mostrar contraseña" (Show password). At the bottom left is a link "¿Olvidaste la contraseña?" (Forgot password?) and at the bottom right is a blue "Siguiente" (Next) button.

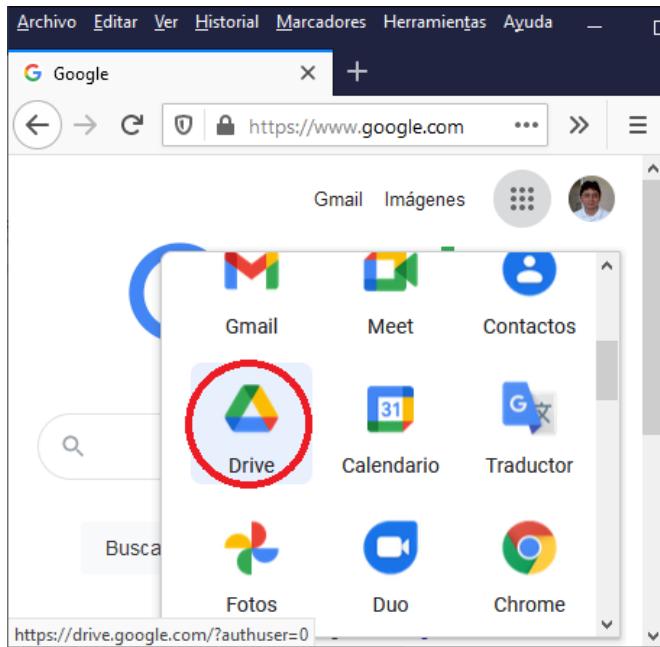
Español (Latinoamérica) ▾ Ayuda Privacidad Condiciones

... e inmediatamente se mostrará una ventana como ésta:

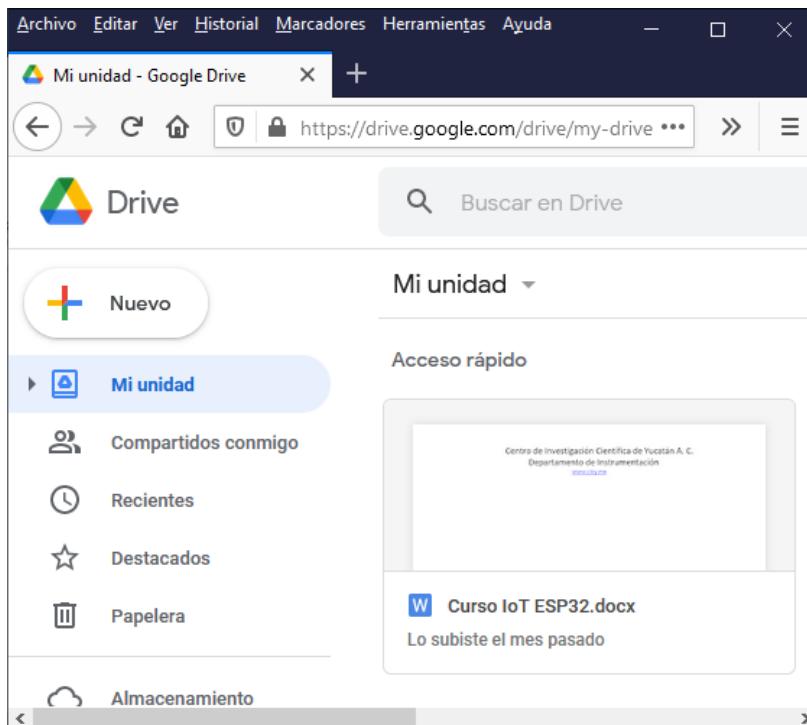


Presione el botón con forma de puntitos que se señala en la figura anterior con

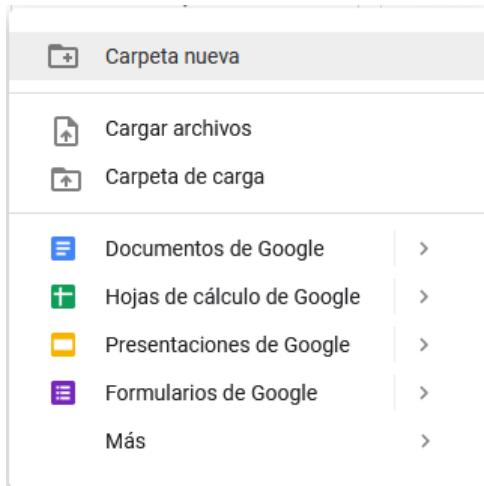
una flecha roja y observará lo siguiente:



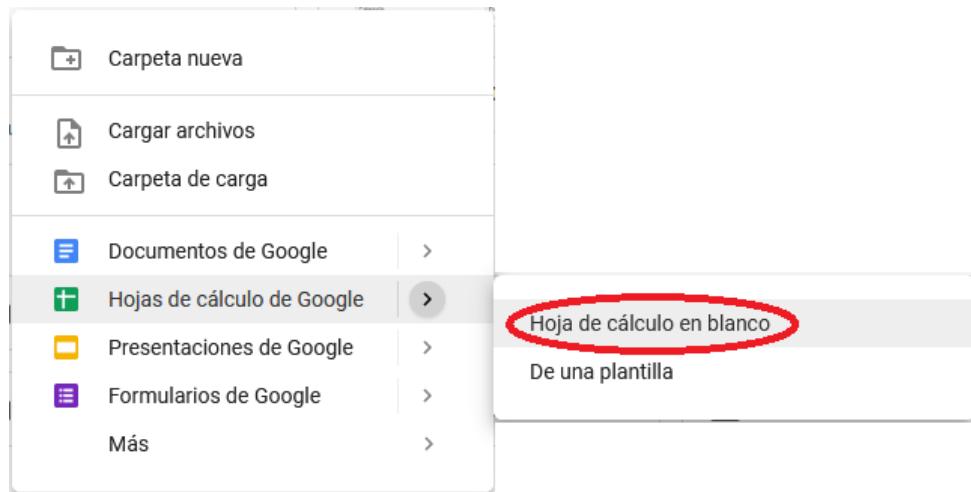
Presione el botón Drive, así como se muestra en la figura anterior y se abrirá otra ventana:



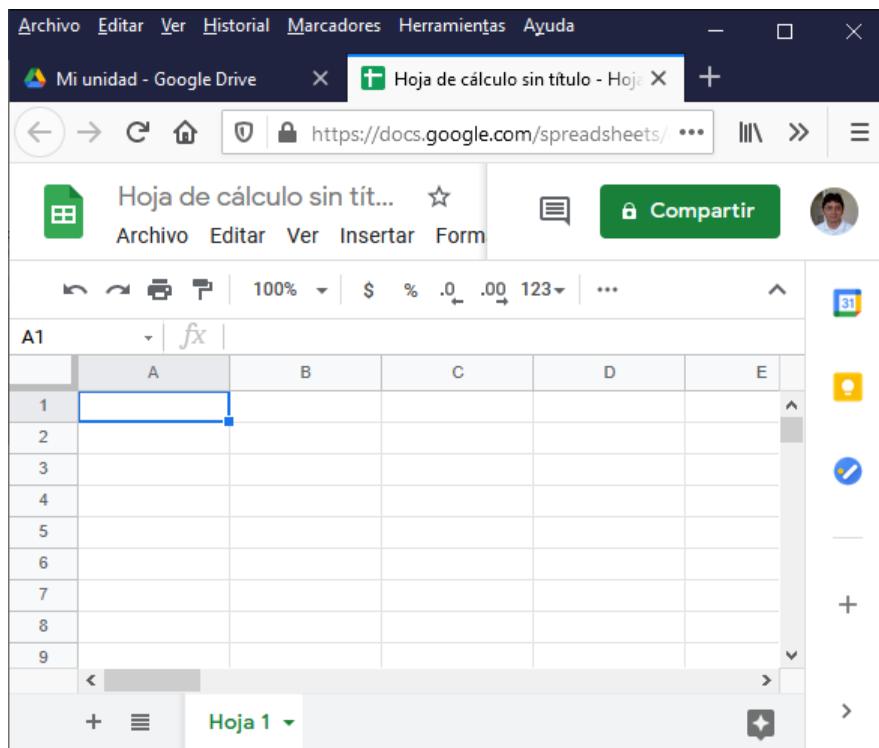
Ubique un área en blanco del lado derecho de la página web (en la sección de archivos o carpetas, NO ponga el ratón encima de las carpetas o archivos) y presione el botón derecho del ratón y le saldrá una ventana como ésta:



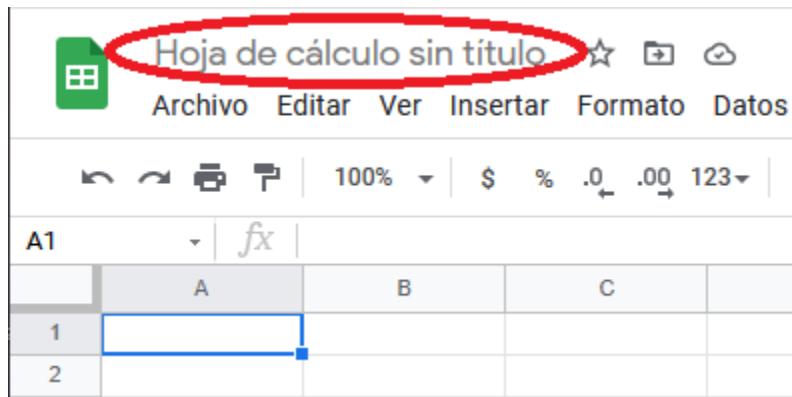
Ubíquese en el área de “Hojas de cálculo de Google” y saldrá un menú, así como se muestra a continuación:



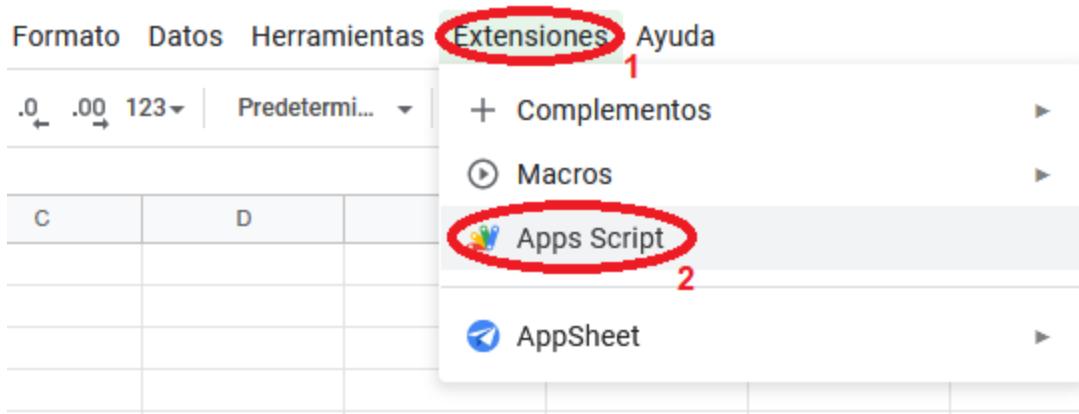
... y seleccione Hoja de cálculo en blanco:



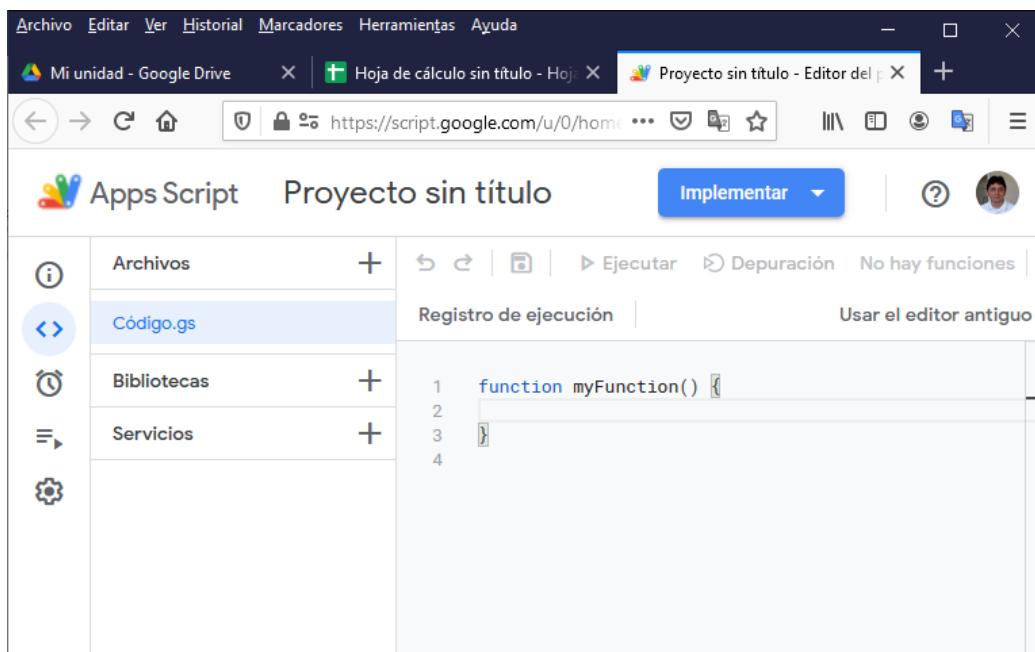
... e inmediatamente se mostrará la hoja de cálculo lista para ser usada. Asigne un nombre presionando en el área marcada, así como se muestra en la siguiente figura:



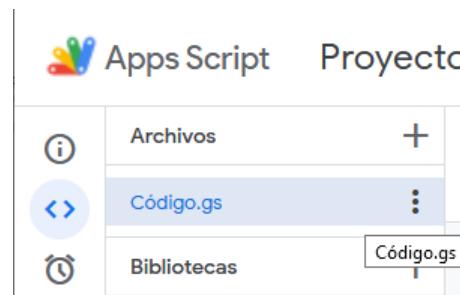
Asigne el nombre SimpleScript y presione la tecla “ENTER” para guardar los cambios. Seguidamente, ubique la barra de herramientas en la parte superior de la hoja de cálculo y seleccione en secuencia el menú, así como se muestra a continuación:



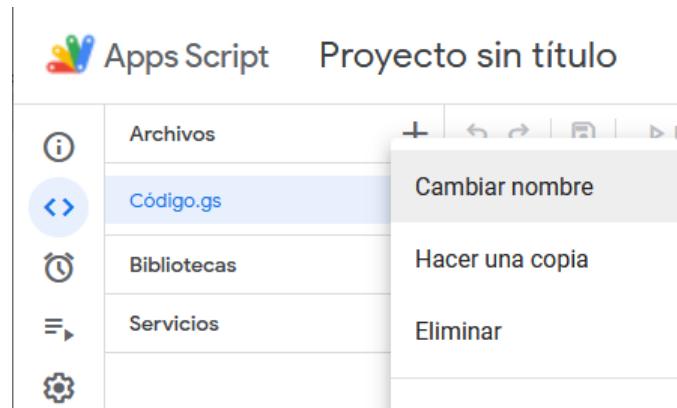
... y se abrirá una nueva pestaña, así como se muestra a continuación:



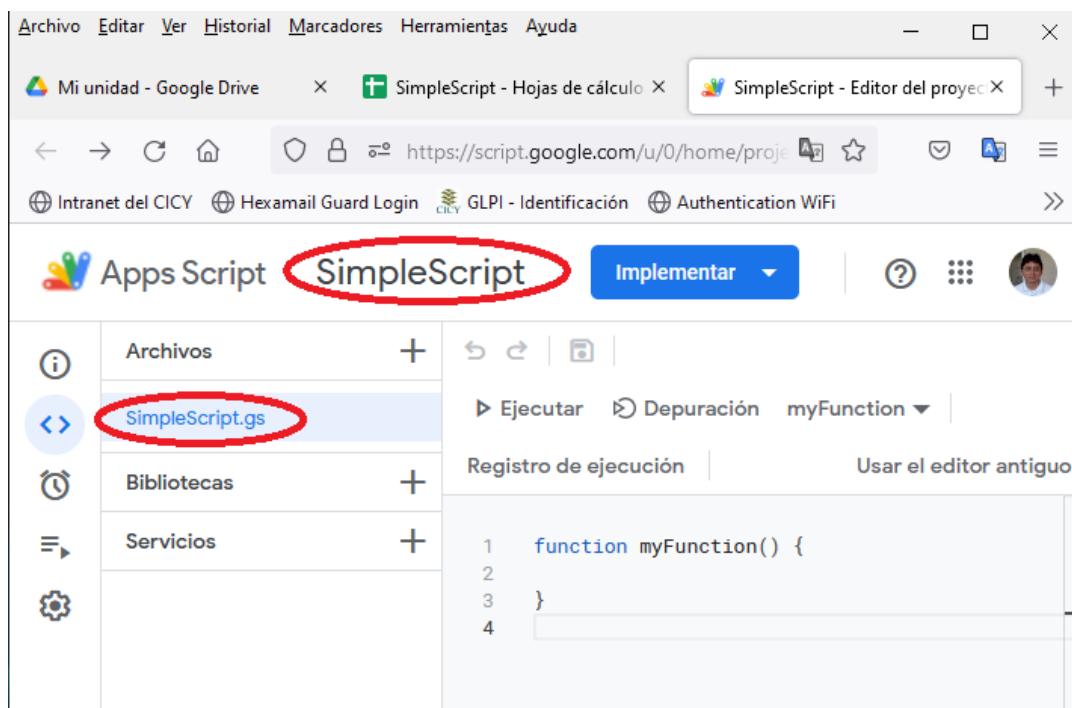
Ubíquese en el campo donde se encuentra el nombre del código y se activará un botón con tres puntitos:



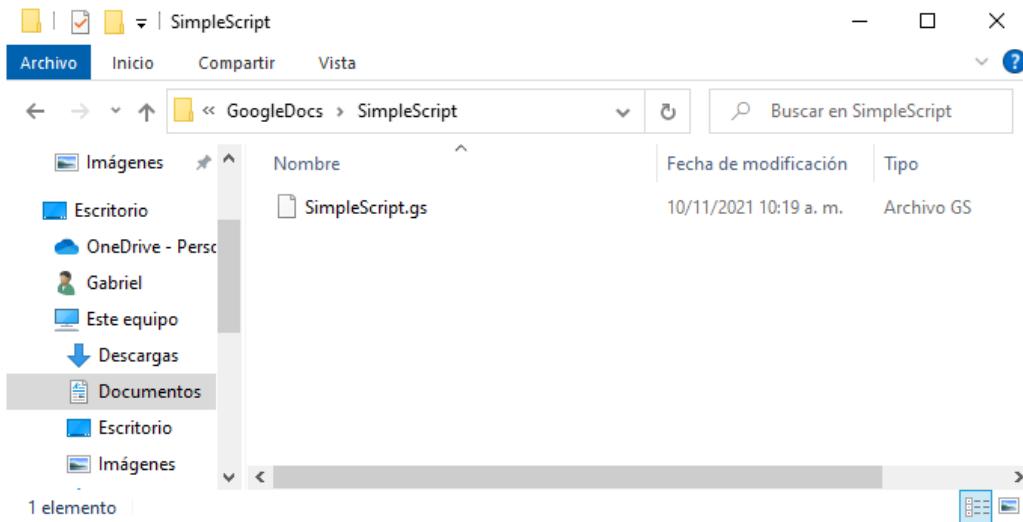
... presione el botón para activar el menú emergente:



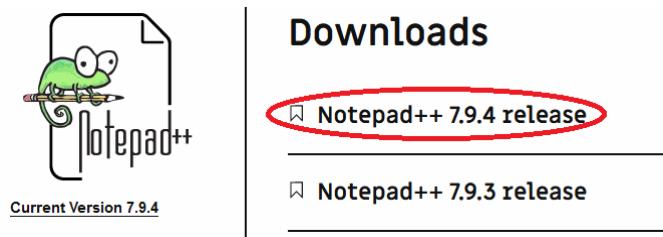
... y cambie el nombre del código a SimpleScript, haga lo mismo en la sección del nombre de proyecto y cambie el nombre a SimpleScript, así como se muestra:



Ahora, regresemos a la carpeta recién descargada y copiada a la carpeta de Documentos, entre a la carpeta GoogleDocs -> ScriptGoogle, así como se muestra a continuación:



Por facilidad, se requiere la instalación del programa llamado Notepad++ y lo puede obtener en <https://notepad-plus-plus.org/downloads/>:

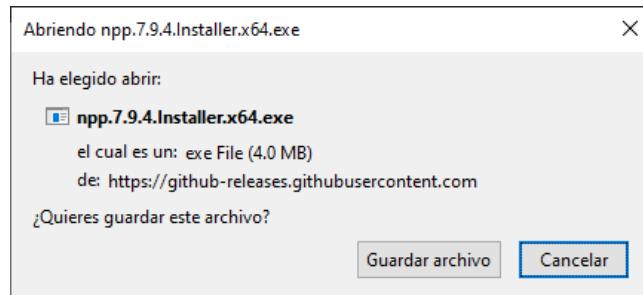


Descargue la versión correspondiente a la versión de su sistema operativo, si desconoce la versión, prueba instalando primero la versión de 64 bits, si falla la instalación, pruebe con la versión de 32 bits:

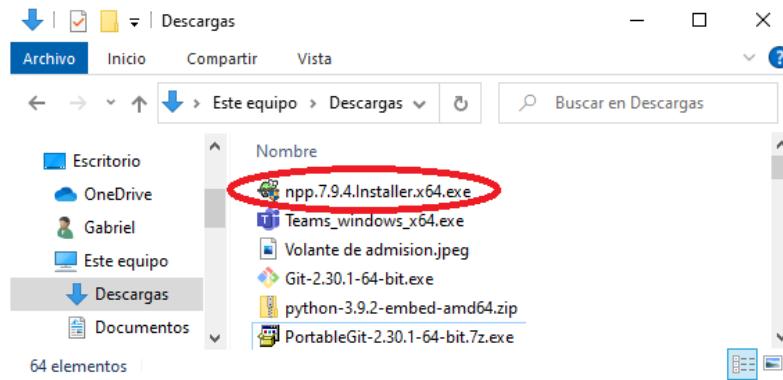
#### Download 64-bit x64

- [Installer | GPG Signature](#)
- [Portable \(zip\) | GPG Signature](#)
- [Portable \(7z\) | GPG Signature](#)
- [Mini-portable \(7z\) | GPG Signature](#)

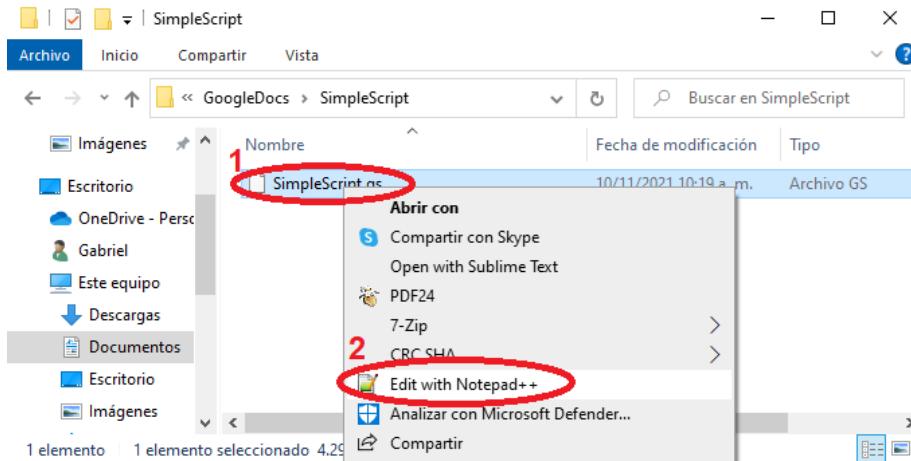
Descargue el archivo:



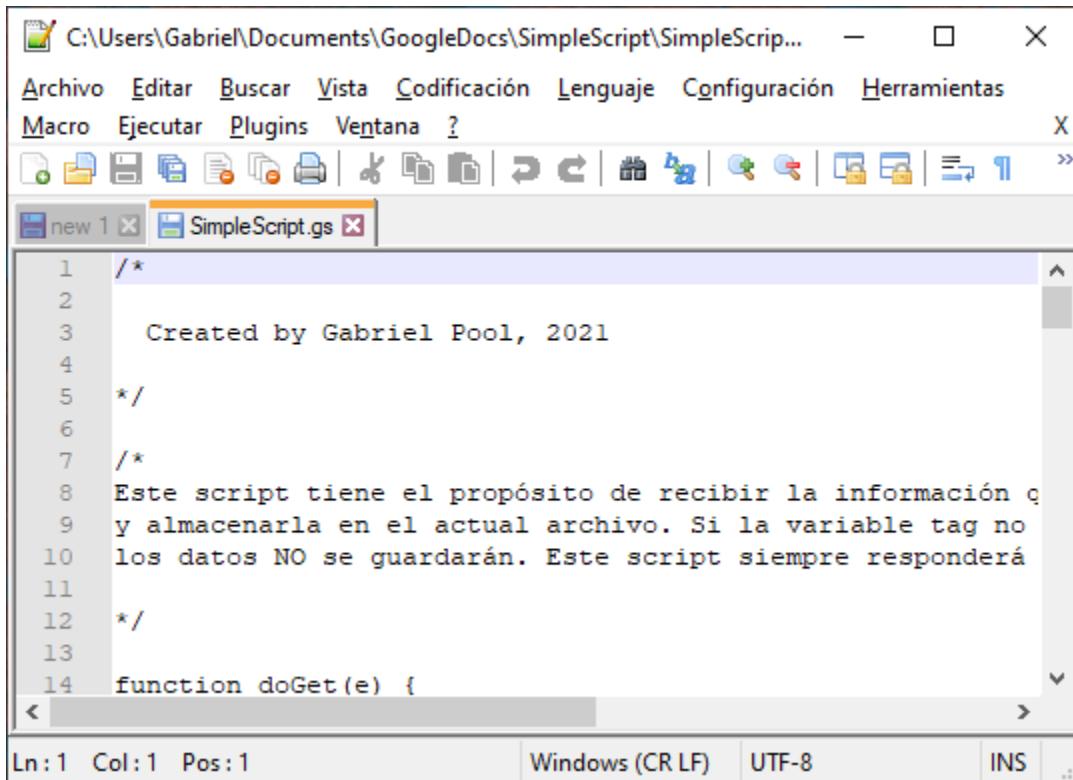
... e instale en su PC con las opciones que trae predeterminadas (sólo presione el botón “siguiente” hasta que el programa se instale):



Una vez instalado el Notepad++, regresamos a la ventana que se muestra donde se ubica el Script de Google, ubique el archivo SimpleScript.gs y, con el botón derecho del mouse, dé un click sobre el archivo “SimpleScript.gs” y seleccione la opción que se muestra a continuación:



... e inmediatamente se abrirá una ventana como ésta:



The screenshot shows a script editor window with the following details:

- Title bar: C:\Users\Gabriel\Documents\GoogleDocs\SimpleScript\SimpleScript...
- Menu bar: Archivo, Editar, Buscar, Vista, Codificación, Lenguaje, Configuración, Herramientas, Macro, Ejecutar, Plugins, Ventana, ?
- Toolbar: Includes icons for file operations like Open, Save, Print, and various script-related functions.
- Tab bar: new 1 (disabled) and SimpleScript.gs (active).
- Code editor:

```
1  /*
2
3     Created by Gabriel Pool, 2021
4
5 */
6
7 /*
8 Este script tiene el propósito de recibir la información q
9 y almacenarla en el actual archivo. Si la variable tag no
10 los datos NO se guardarán. Este script siempre responderá
11
12 */
13
14 function doGet(e) {
```
- Status bar: Ln:1 Col:1 Pos:1, Windows (CR LF), UTF-8, INS.

... presione las teclas Ctrl + a (Seleccionar todo el texto) y posteriormente Ctrl + c para copiar el texto, ubíquese en el navegador web, borre el contenido SimpleScript.gs en el navegador web y pegue el contenido copiado, así como se muestra:

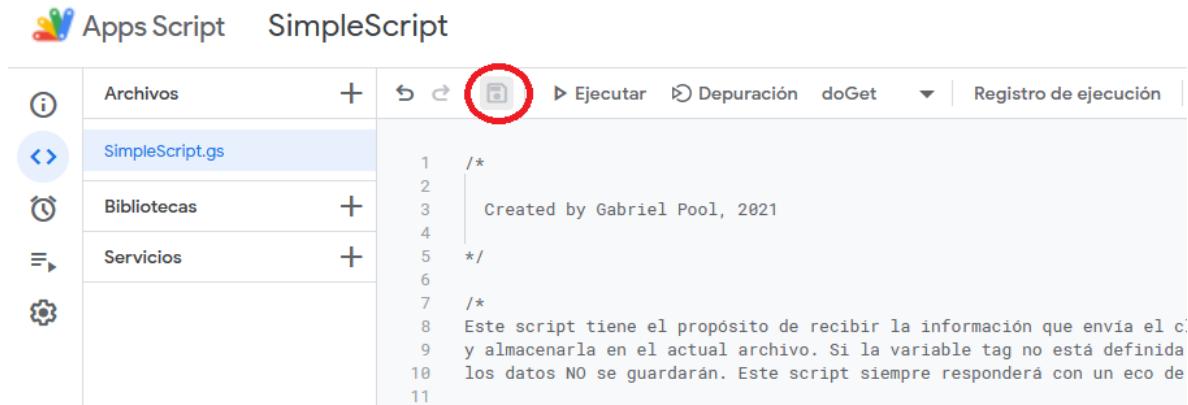


The screenshot shows the Google Apps Script editor interface with the following details:

- Left sidebar: Archivos, SimpleScript.gs (selected), Bibliotecas, Servicios.
- Right pane: Script Editor showing the pasted code:

```
1  /*
2
3     Created by Gabriel Pool, 2021
4
5 */
6
7 /*
8 Este script tiene el propósito de recibir la información que envía el cli
9 y almacenarla en el actual archivo. Si la variable tag no está definida c
10 los datos NO se guardarán. Este script siempre responderá con un eco de l
11
```

Y presione el botón en forma de diskette para guardar los cambios:



Apps Script    SimpleScript

Archivos    +

SimpleScript.gs

Bibliotecas    +

Servicios    +

```

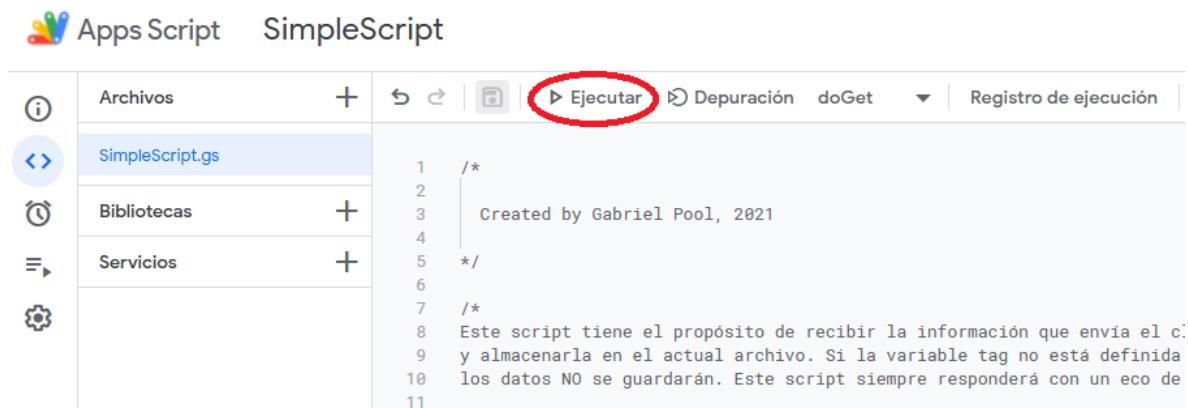
1  /*
2
3   Created by Gabriel Pool, 2021
4
5 */
6
7 /*
8 Este script tiene el propósito de recibir la información que envia el c:
9 y almacenarla en el actual archivo. Si la variable tag no está definida
10 los datos NO se guardarán. Este script siempre responderá con un eco de
11

```

... seguidamente, se requieren dos pasos para gestionar los permisos de ejecución:

### 1er Paso:

Seleccione el archivo SimpleScript.gs y presione el botón de ejecutar, así como se muestra:



Apps Script    SimpleScript

Archivos    +

SimpleScript.gs

Bibliotecas    +

Servicios    +

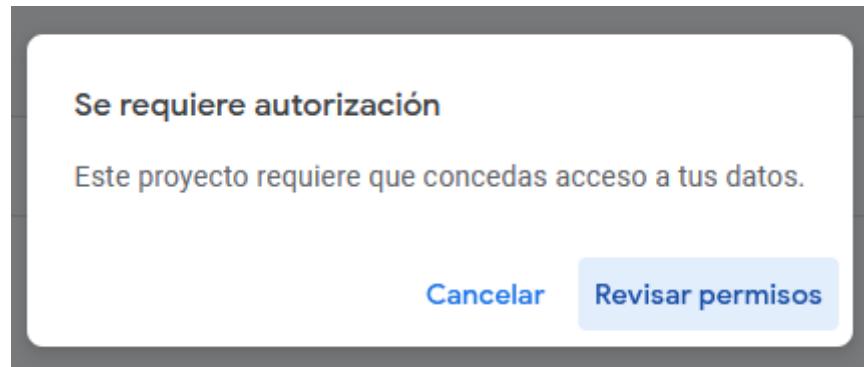
Ejecutar    Depuración    doGet    Registro de ejecución

```

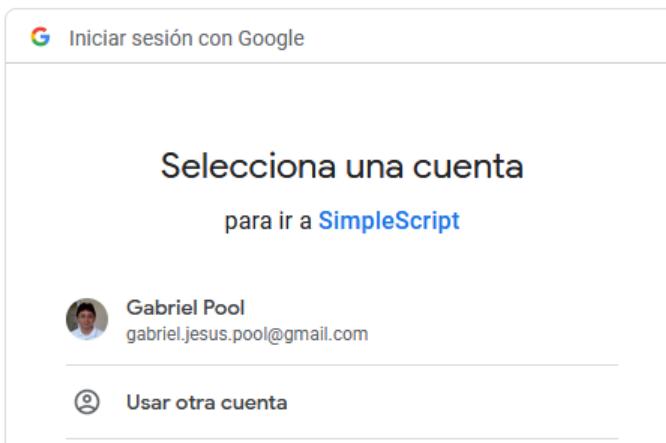
1  /*
2
3   Created by Gabriel Pool, 2021
4
5 */
6
7 /*
8 Este script tiene el propósito de recibir la información que envia el c:
9 y almacenarla en el actual archivo. Si la variable tag no está definida
10 los datos NO se guardarán. Este script siempre responderá con un eco de
11

```

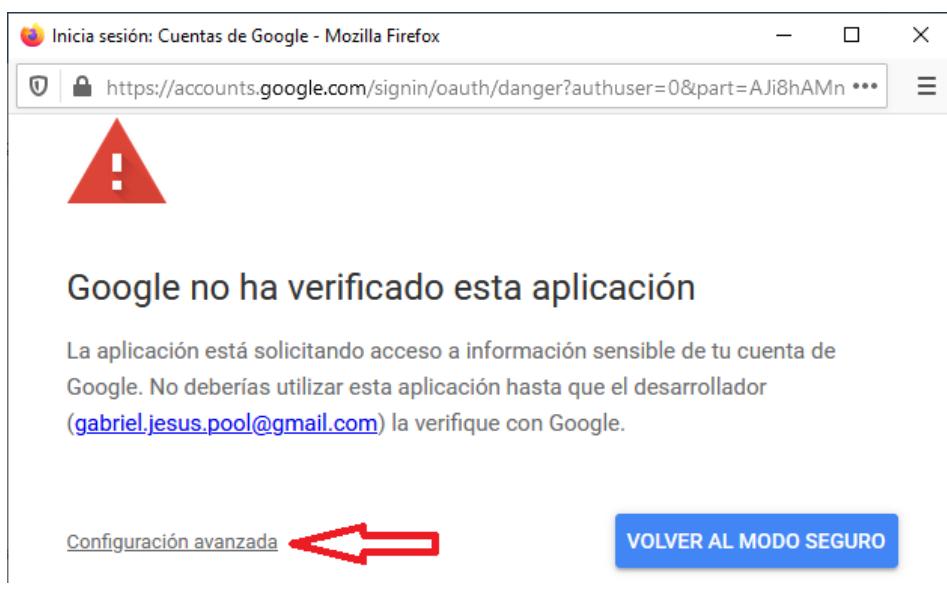
... e inmediatamente se oscurecerá la pantalla y saldrá un mensaje como éste:



... presione el botón “Revisar Permisos”:

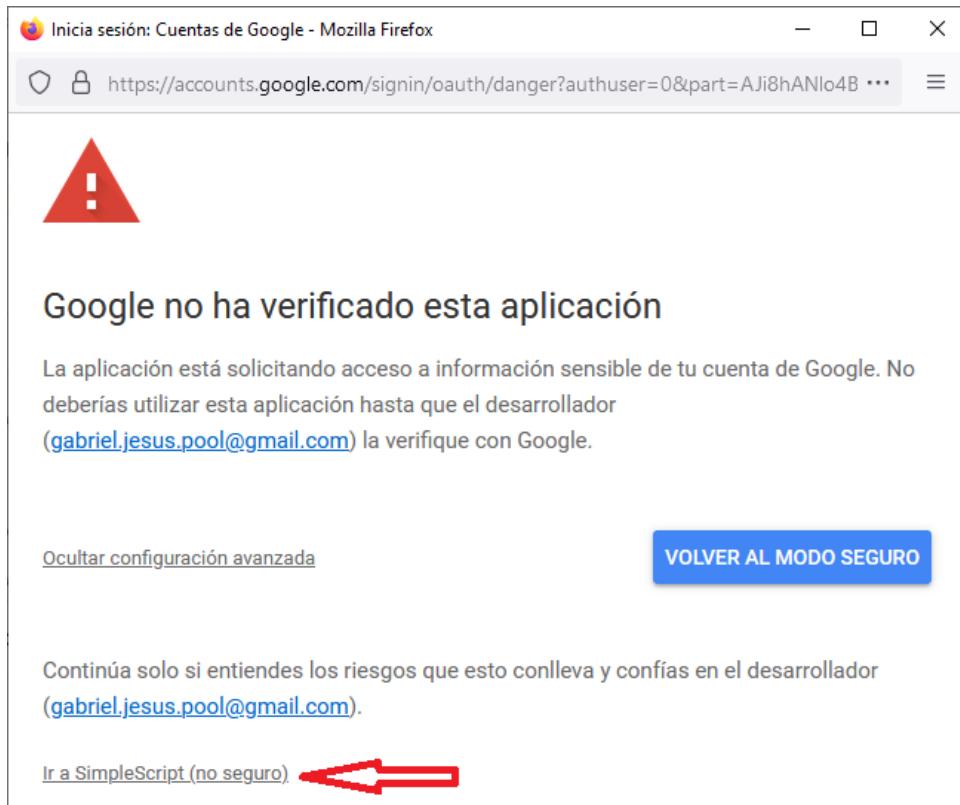


... presione en el área donde se encuentra su logo y correo e inmediatamente saldrá una ventana como ésta:

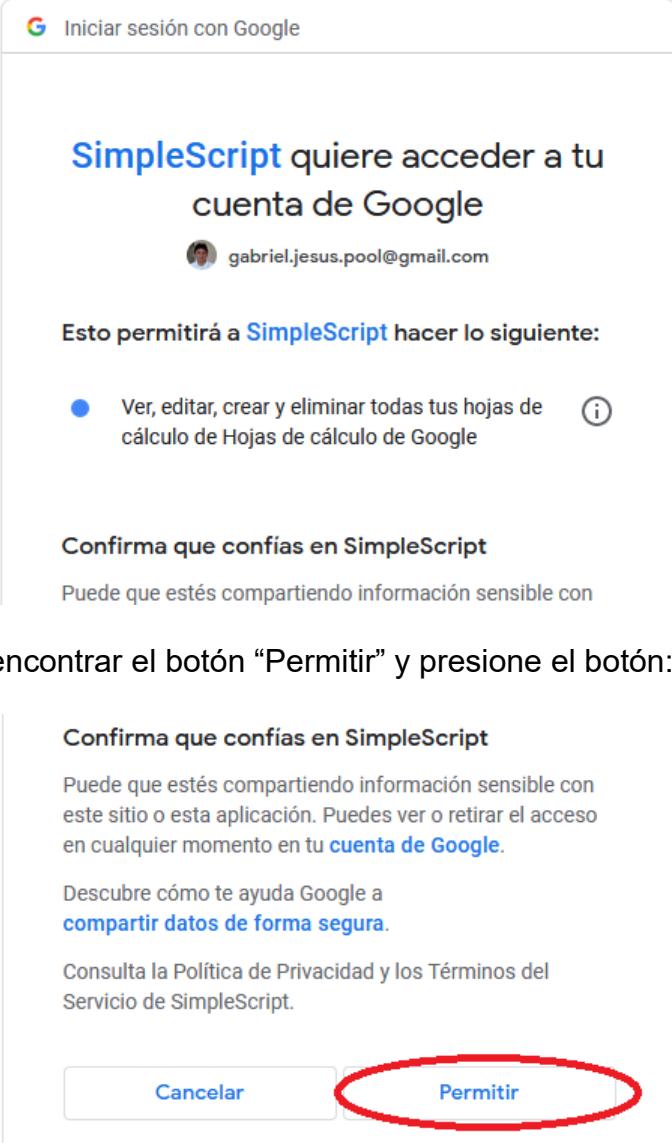


149/224

... presione el hipertexto que se encuentra marcado con una flecha en la figura anterior y saldrá otra ventana como ésta:



... nuevamente, presione el hipertexto marcado en la figura anterior y saldrá otra ventana como ésta:

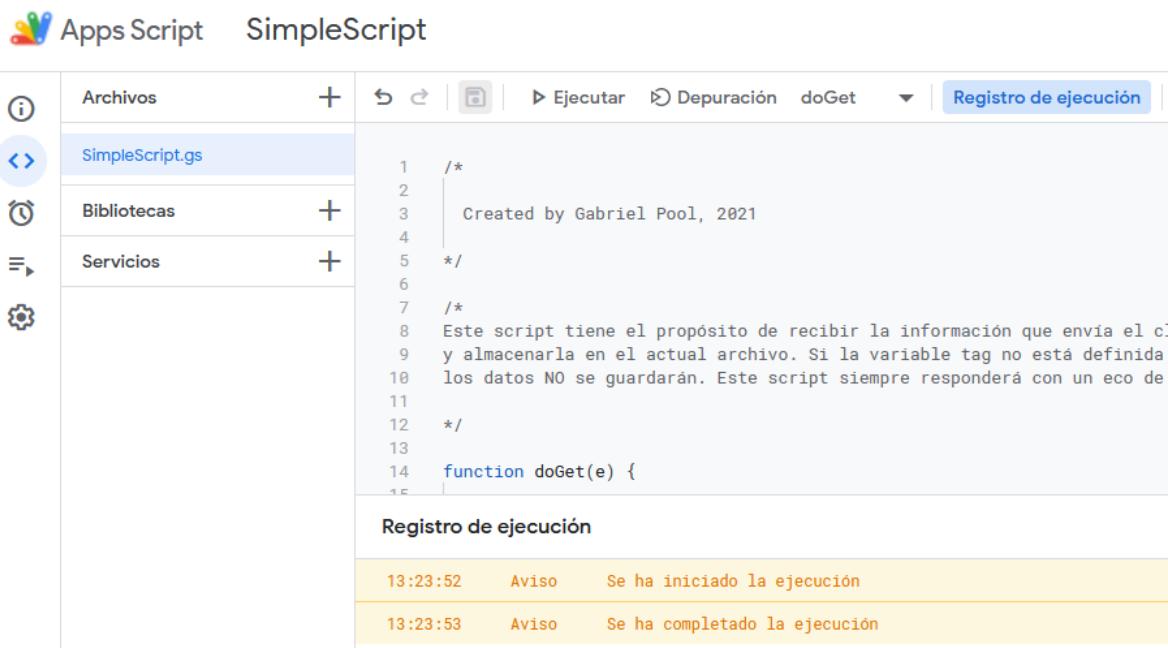


The screenshot shows a Google OAuth consent screen. At the top, it says "Iniciar sesión con Google". The main heading reads "SimpleScript quiere acceder a tu cuenta de Google". Below that is the user's email address "gabriel.jesus.pool@gmail.com". A section titled "Esto permitirá a SimpleScript hacer lo siguiente:" lists one permission: "Ver, editar, crear y eliminar todas tus hojas de cálculo de Hojas de cálculo de Google". A blue circular icon with a white question mark is next to the text. Below this, a section titled "Confirma que confías en SimpleScript" contains the text "Puede que estés compartiendo información sensible con". At the bottom, there are two buttons: "Cancelar" and "Permitir", with "Permitir" circled in red.

... deslice hasta encontrar el botón “Permitir” y presione el botón:

La ventana anterior se cerrará, pero si se toma demasiado tiempo en seguir estos pasos, en el código de Google mostrará un mensaje indicando que requiere permisos de ejecución, basta con presionar el botón de “ejecutar” y el código se ejecutará sin abrir la ventana para la gestión de permisos. Al ejecutar el código se mostrará un mensaje:

Apps Script    SimpleScript

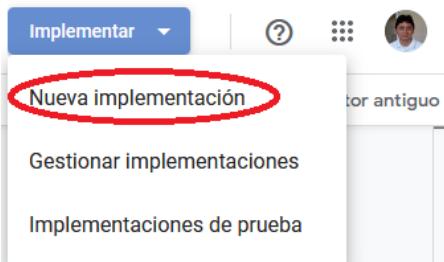


The screenshot shows the Google Apps Script interface. On the left, there's a sidebar with icons for Archivos, Bibliotecas, and Servicios. The main area shows a script named "SimpleScript.gs" with the following code:

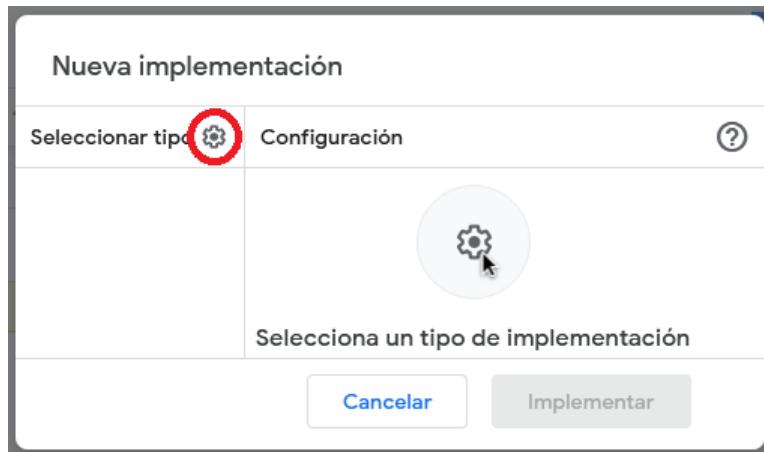
```

1  /*
2   | Created by Gabriel Pool, 2021
3   */
4
5  /*
6   Este script tiene el propósito de recibir la información que envía el cliente
7   y almacenarla en el actual archivo. Si la variable tag no está definida con el valor "Update",
8   los datos NO se guardarán. Este script siempre responderá con un eco de los datos enviados.
9
10 */
11
12 function doGet(e) {
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
159
160
161
162
163
164
165
166
167
168
169
169
170
171
172
173
174
175
176
177
178
179
179
180
181
182
183
184
185
186
187
187
188
189
189
190
191
192
193
194
195
196
197
198
199
199
200
201
202
203
204
205
206
207
208
209
209
210
211
212
213
214
215
216
217
217
218
219
219
220
221
222
223
224
225
226
227
227
228
229
229
230
231
232
233
234
235
236
237
237
238
239
239
240
241
242
243
244
245
245
246
247
247
248
249
249
250
251
252
253
254
255
255
256
257
257
258
259
259
260
261
262
263
264
265
265
266
267
267
268
269
269
270
271
272
273
274
275
275
276
277
277
278
279
279
280
281
282
283
284
285
285
286
287
287
288
289
289
290
291
292
293
294
295
295
296
297
297
298
299
299
300
301
302
303
304
305
305
306
307
307
308
309
309
310
311
312
313
314
314
315
316
316
317
318
318
319
319
320
321
322
323
323
324
325
325
326
327
327
328
329
329
330
331
332
333
333
334
335
335
336
337
337
338
339
339
340
341
342
343
343
344
345
345
346
347
347
348
349
349
350
351
352
353
353
354
355
355
356
357
357
358
359
359
360
361
362
363
363
364
365
365
366
367
367
368
369
369
370
371
371
372
373
373
374
375
375
376
377
377
378
379
379
380
381
381
382
383
383
384
385
385
386
387
387
388
389
389
390
391
391
392
393
393
394
395
395
396
397
397
398
399
399
400
401
401
402
403
403
404
405
405
406
407
407
408
409
409
410
411
411
412
413
413
414
415
415
416
417
417
418
419
419
420
421
421
422
423
423
424
425
425
426
427
427
428
429
429
430
431
431
432
433
433
434
435
435
436
437
437
438
439
439
440
441
441
442
443
443
444
445
445
446
447
447
448
449
449
450
451
451
452
453
453
454
455
455
456
457
457
458
459
459
460
461
461
462
463
463
464
465
465
466
467
467
468
469
469
470
471
471
472
473
473
474
475
475
476
477
477
478
479
479
480
481
481
482
483
483
484
485
485
486
487
487
488
489
489
490
491
491
492
493
493
494
495
495
496
497
497
498
499
499
500
501
501
502
503
503
504
505
505
506
507
507
508
509
509
510
511
511
512
513
513
514
515
515
516
517
517
518
519
519
520
521
521
522
523
523
524
525
525
526
527
527
528
529
529
530
531
531
532
533
533
534
535
535
536
537
537
538
539
539
540
541
541
542
543
543
544
545
545
546
547
547
548
549
549
550
551
551
552
553
553
554
555
555
556
557
557
558
559
559
560
561
561
562
563
563
564
565
565
566
567
567
568
569
569
570
571
571
572
573
573
574
575
575
576
577
577
578
579
579
580
581
581
582
583
583
584
585
585
586
587
587
588
589
589
590
591
591
592
593
593
594
595
595
596
597
597
598
599
599
600
601
601
602
603
603
604
605
605
606
607
607
608
609
609
610
611
611
612
613
613
614
615
615
616
617
617
618
619
619
620
621
621
622
623
623
624
625
625
626
627
627
628
629
629
630
631
631
632
633
633
634
635
635
636
637
637
638
639
639
640
641
641
642
643
643
644
645
645
646
647
647
648
649
649
650
651
651
652
653
653
654
655
655
656
657
657
658
659
659
660
661
661
662
663
663
664
665
665
666
667
667
668
669
669
670
671
671
672
673
673
674
675
675
676
677
677
678
679
679
680
681
681
682
683
683
684
685
685
686
687
687
688
689
689
690
691
691
692
693
693
694
695
695
696
697
697
698
699
699
700
701
701
702
703
703
704
705
705
706
707
707
708
709
709
710
711
711
712
713
713
714
715
715
716
717
717
718
719
719
720
721
721
722
723
723
724
725
725
726
727
727
728
729
729
730
731
731
732
733
733
734
735
735
736
737
737
738
739
739
740
741
741
742
743
743
744
745
745
746
747
747
748
749
749
750
751
751
752
753
753
754
755
755
756
757
757
758
759
759
760
761
761
762
763
763
764
765
765
766
767
767
768
769
769
770
771
771
772
773
773
774
775
775
776
777
777
778
779
779
780
781
781
782
783
783
784
785
785
786
787
787
788
789
789
790
791
791
792
793
793
794
795
795
796
797
797
798
799
799
800
801
801
802
803
803
804
805
805
806
807
807
808
809
809
810
811
811
812
813
813
814
815
815
816
817
817
818
819
819
820
821
821
822
823
823
824
825
825
826
827
827
828
829
829
830
831
831
832
833
833
834
835
835
836
837
837
838
839
839
840
841
841
842
843
843
844
845
845
846
847
847
848
849
849
850
851
851
852
853
853
854
855
855
856
857
857
858
859
859
860
861
861
862
863
863
864
865
865
866
867
867
868
869
869
870
871
871
872
873
873
874
875
875
876
877
877
878
879
879
880
881
881
882
883
883
884
885
885
886
887
887
888
889
889
890
891
891
892
893
893
894
895
895
896
897
897
898
899
899
900
901
901
902
903
903
904
905
905
906
907
907
908
909
909
910
911
911
912
913
913
914
915
915
916
917
917
918
919
919
920
921
921
922
923
923
924
925
925
926
927
927
928
929
929
930
931
931
932
933
933
934
935
935
936
937
937
938
939
939
940
941
941
942
943
943
944
945
945
946
947
947
948
949
949
950
951
951
952
953
953
954
955
955
956
957
957
958
959
959
960
961
961
962
963
963
964
965
965
966
967
967
968
969
969
970
971
971
972
973
973
974
975
975
976
977
977
978
979
979
980
981
981
982
983
983
984
985
985
986
987
987
988
989
989
990
991
991
992
993
993
994
995
995
996
997
997
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564

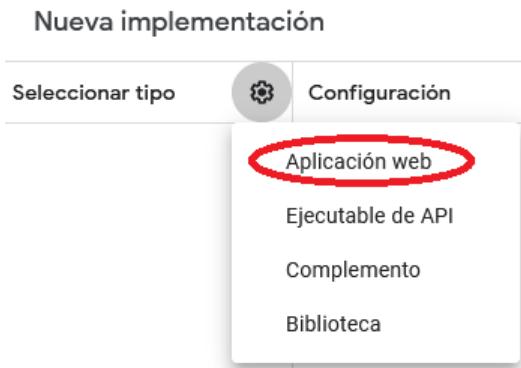
```



Presione la opción “Nueva Implementación”, y saldrá una ventana, así como se muestra:



... presione el botón con forma de engrane y saldrá un menú emergente:



... seleccione la opción “Aplicación Web” y se mostrará lo siguiente:

Nueva implementación

Seleccionar tipo	Configuración	(?)
Aplicación web	<p>Descripción</p> <p>Nueva descripción</p> <p>Aplicación web</p> <p>Ejecutar como</p> <p>Yo (gabriel.jesus.pool@gmail.com)</p> <p>Quién tiene acceso</p> <p>Solo yo</p> <p>También se puede usar como biblioteca. <a href="#">Más información</a></p>	
		<p>Cancelar</p> <p>Implementar</p>

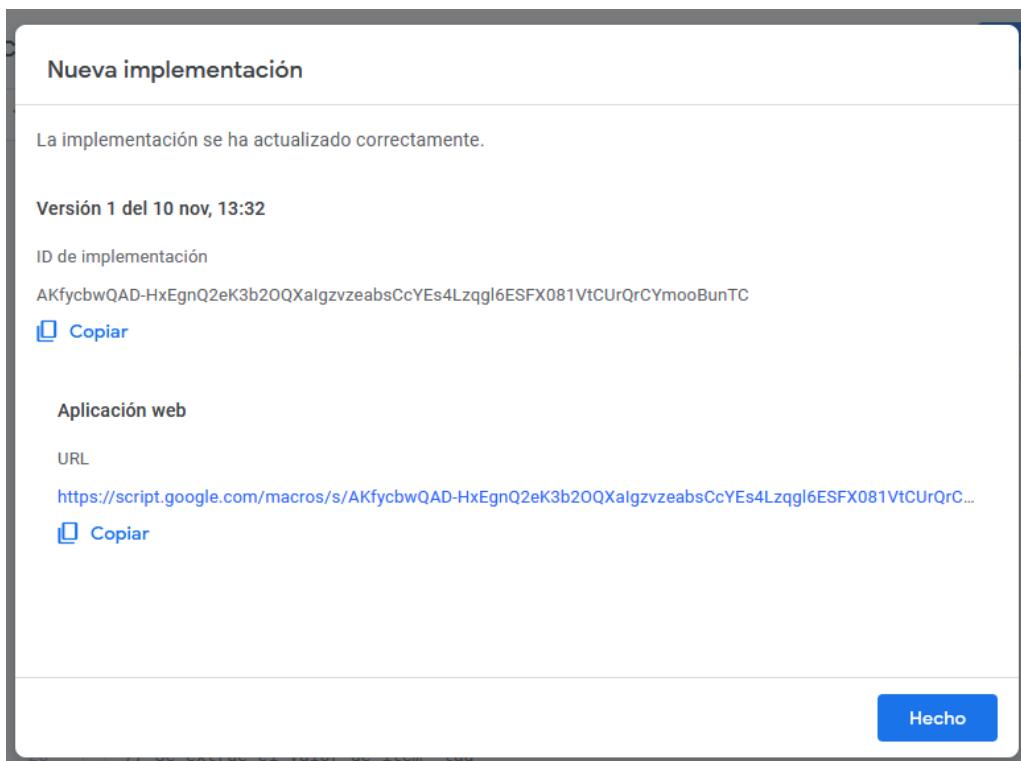
... en el campo nueva descripción ponga la versión de su proyecto, en el campo aplicación web, no haga cambios, y en el último campo llamado “Quien tiene acceso”, seleccione “Cualquier usuario”, los campos deben quedar, así como se muestra:

Nueva implementación

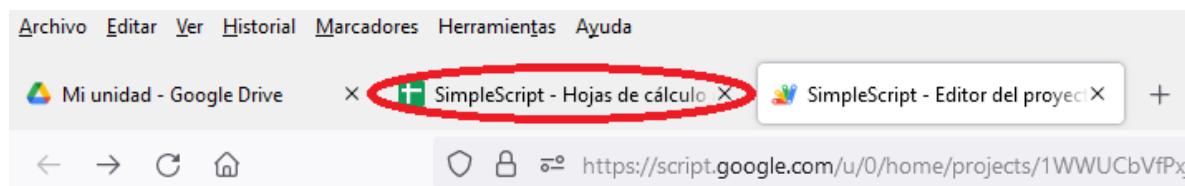
Seleccionar tipo	Configuración	?
Aplicación web	<p>Descripción</p> <p>Nueva descripción</p> <p>version 1</p> <p>Aplicación web</p> <p>Ejecutar como</p> <p>Yo (gabriel.jesus.pool@gmail.com)</p> <p>Quién tiene acceso</p> <p>Cualquier usuario</p> <p>También se puede usar como biblioteca.<a href="#">Más información</a></p>	

[Cancelar](#) [Implementar](#)

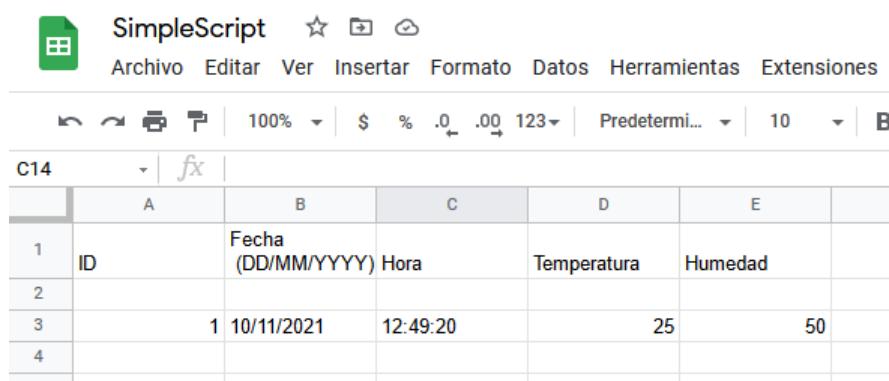
... seguidamente, presione el botón azul llamado “Implementar” y le saldrá una ventana como esta:



Si todo salió bien, seleccione la pestaña marcada en el explorador web, así como se muestra:



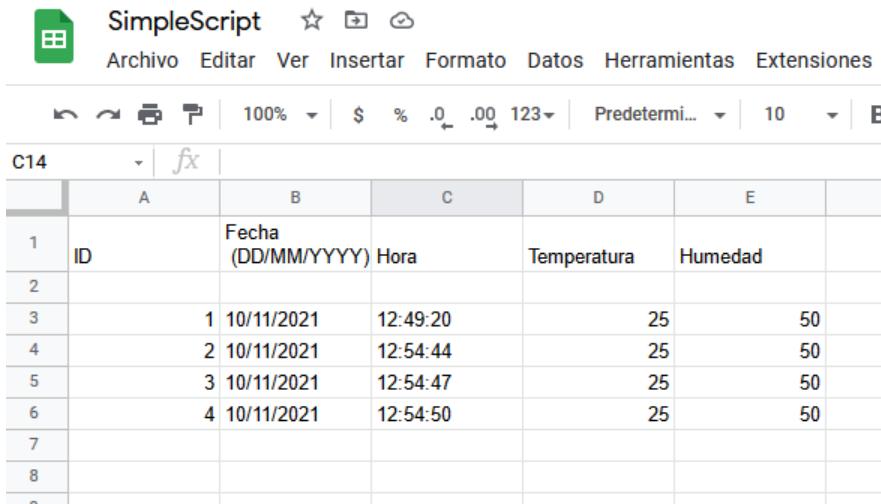
... se deberá apreciar lo siguiente:



The screenshot shows a Google Sheets document titled "SimpleScript". The menu bar includes Archivo, Editar, Ver, Insertar, Formato, Datos, Herramientas, and Extensiones. The toolbar includes standard options like back, forward, print, and zoom. The spreadsheet has a header row with columns labeled "ID", "Fecha (DD/MM/YYYY)", "Hora", "Temperatura", and "Humedad". Data rows show entries for 10/11/2021 at 12:49:20 with values 25 and 50 respectively.

	ID	Fecha (DD/MM/YYYY)	Hora	Temperatura	Humedad
1		10/11/2021	12:49:20	25	50
2					
3					
4					

Cada vez que presione el botón ejecutar, se agregará una nueva línea en el documento de Google:



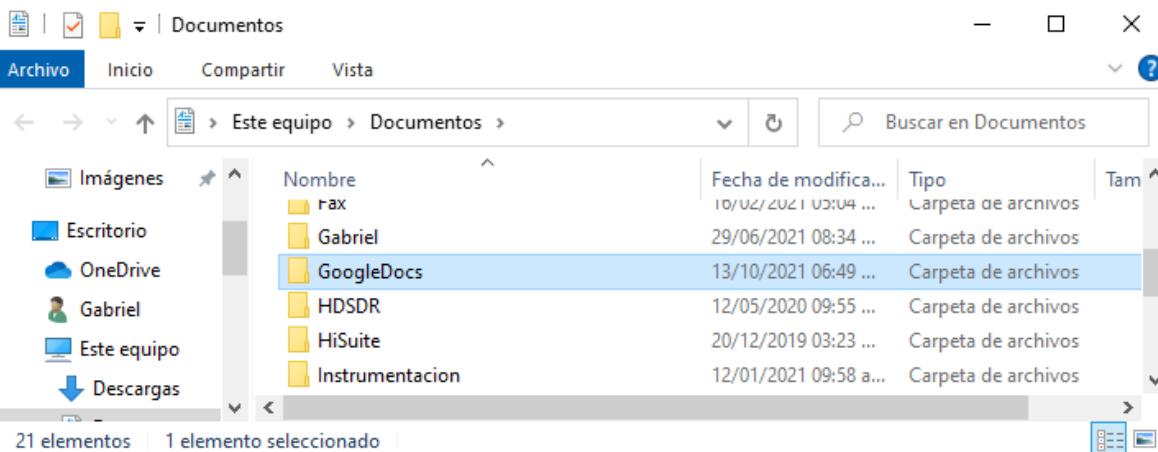
	A	B	C	D	E	
1	ID	Fecha (DD/MM/YYYY)	Hora	Temperatura	Humedad	
2						
3	1	10/11/2021	12:49:20	25	50	
4	2	10/11/2021	12:54:44	25	50	
5	3	10/11/2021	12:54:47	25	50	
6	4	10/11/2021	12:54:50	25	50	
7						
8						
~						

Con esto habremos concluido la implementación del script en Google. Este script se limita a guardar los datos de temperatura y humedad enviados por el dispositivo IoT. Para salir, simplemente cierre las ventanas del explorador web. En el último punto de éste capítulo titulado: “El Cliente para Google Apps Script implementado en NodeMCU-32S” se implementará el código del dispositivo IoT.

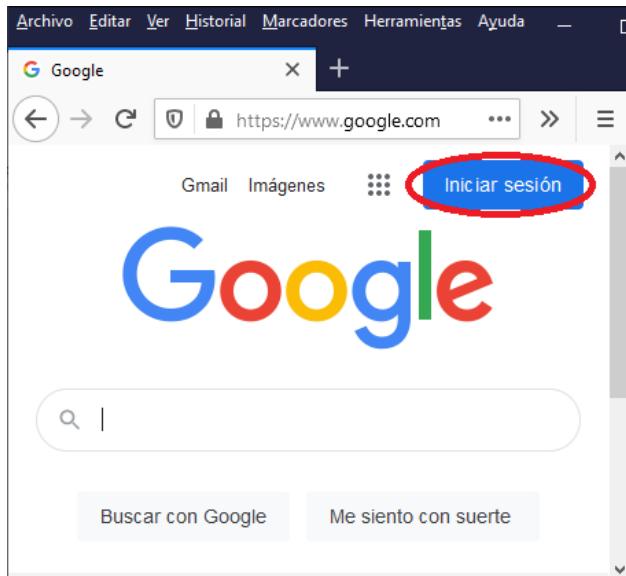
### 3.3.2 Simple página web dinámica en Google Apps Script

**Objetivo específico:** Configurará e implementará los servicios necesarios para mostrar una página web dinámica usando el lenguaje JavaScript.

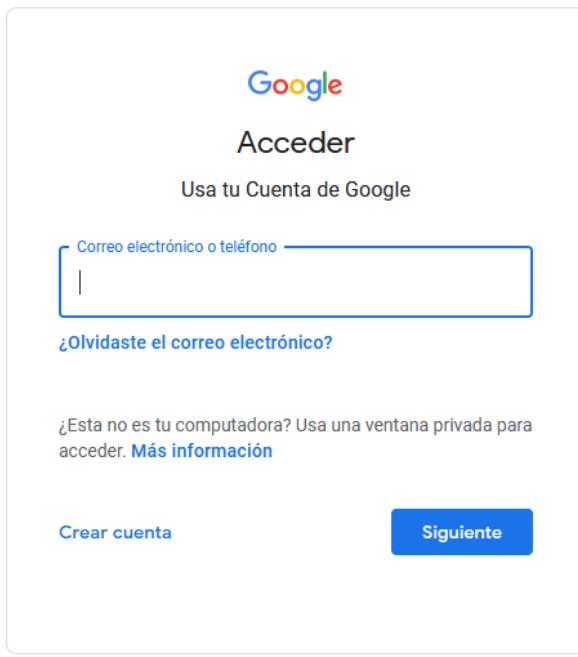
En la actualidad, las herramientas de Google han ganado popularidad y con el crecimiento de los dispositivos IoT, Google no se queda atrás, ha implementado una herramienta llamada Google Apps Script el cual, está basado en el lenguaje JavaScript. Dicho lenguaje es de alto nivel y permite la implementación rápida de soluciones. El JavaScript es ampliamente usado en las páginas web para gestionar servicios en segundo plano. En el sitio <https://developers.google.com/apps-script> se puede encontrar la ayuda necesaria para la implementación de soluciones, este proyecto se enfocará a las WebApps y puede encontrar la documentación en este sitio: <https://developers.google.com/apps-script/guides/web>. Como dice un dicho muy popular “Para muestra basta un botón”. Para implementar este proyecto se requiere de la descarga de los archivos del proyecto ubicado en <https://github.com/gpoolb/ESP32>. Descargue los archivos ubique la carpeta llamada “GoogleDocs” y copie la carpeta en Mis Documentos, así como se muestra:



Seguidamente, se parte del hecho que Ud ya tiene una cuenta en Google. En la página principal de Google, presione el botón “Iniciar sesión”:



Introduzca su nombre de usuario:



Google

Acceder

Usa tu Cuenta de Google

Correo electrónico o teléfono \_\_\_\_\_

[¿Olvidaste el correo electrónico?](#)

¿Esta no es tu computadora? Usa una ventana privada para acceder. [Más información](#)

[Crear cuenta](#) [Siguiente](#)

Español (Latinoamérica) ▾

Ayuda

Privacidad

Condiciones

y contraseña:

Google

Gabriel Pool

 gabriel.jesus.pool@gmail.com ▾

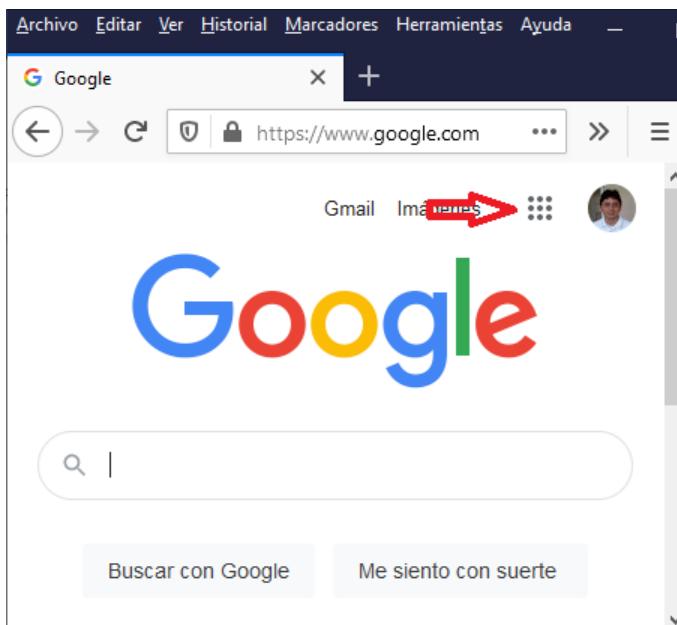
Ingresá tu contraseña \_\_\_\_\_

Mostrar contraseña

[¿Olvidaste la contraseña?](#) [Siguiente](#)

Español (Latinoamérica) ▾ Ayuda Privacidad Condiciones

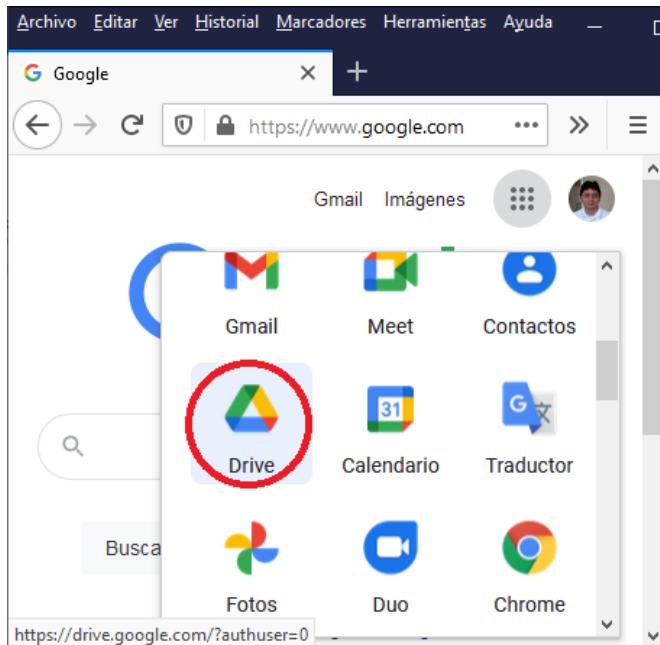
... e inmediatamente se mostrará una ventana como ésta:



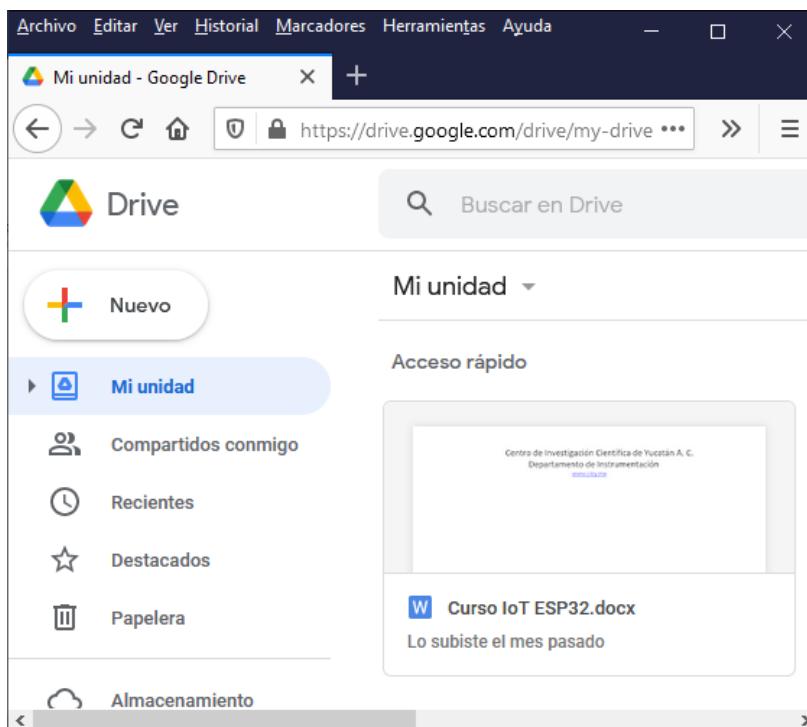
Presione el botón con forma de puntitos que se señala en la figura anterior con

160/224

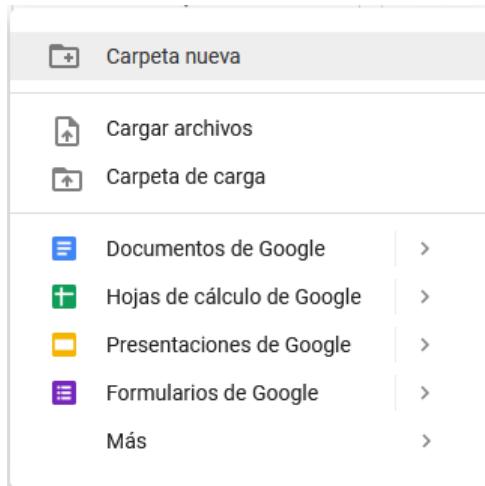
una flecha roja y observará lo siguiente:



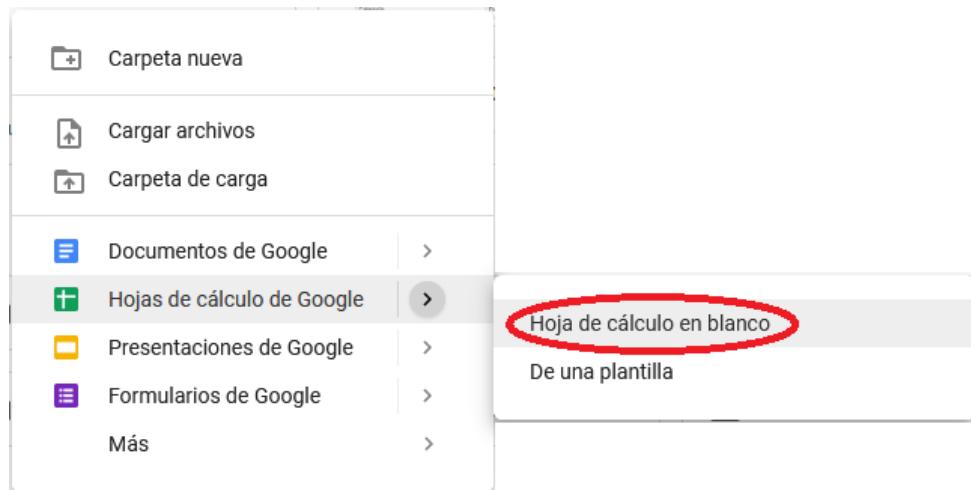
Presione el botón Drive, así como se muestra en la figura anterior y se abrirá otra ventana:



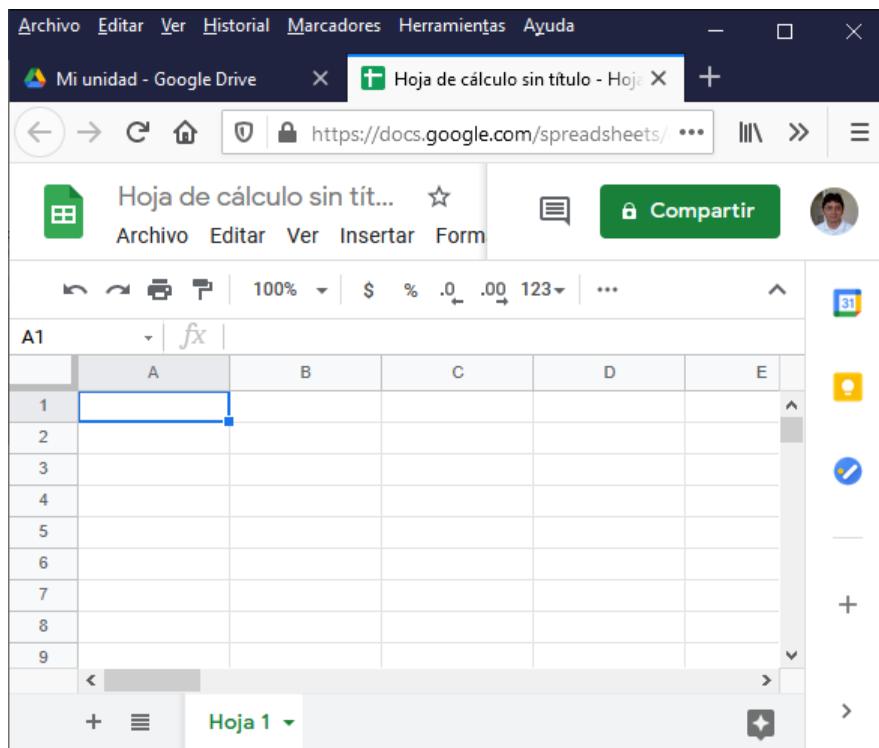
Ubique un área en blanco del lado derecho de la página web (en la sección de archivos o carpetas, NO ponga el ratón encima de las carpetas o archivos) y presione el botón derecho del ratón y le saldrá una ventana como ésta:



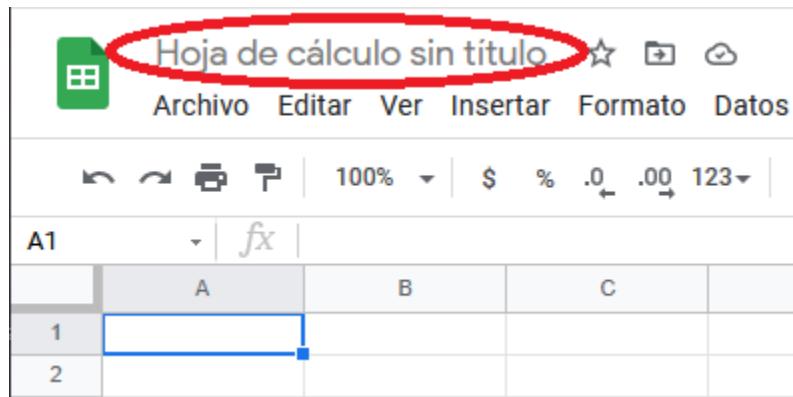
Ubíquese en el área de “Hojas de cálculo de Google” y saldrá un menú, así como se muestra a continuación:



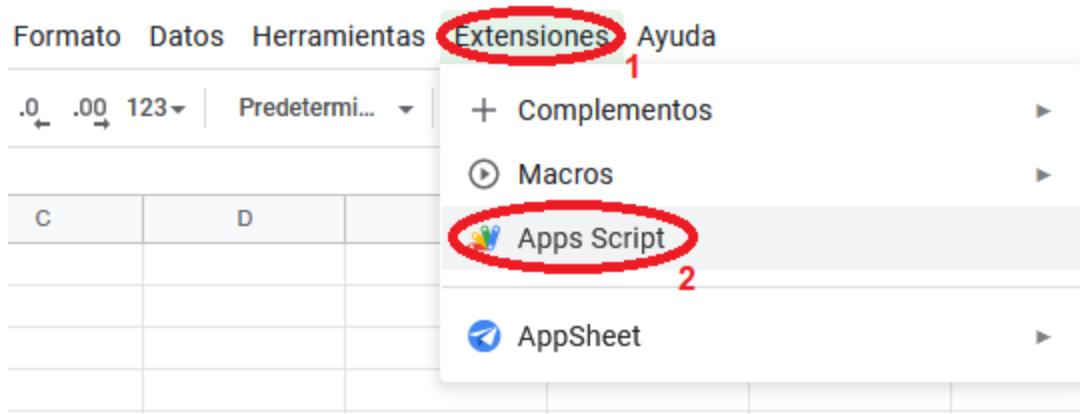
... y seleccione Hoja de cálculo en blanco:



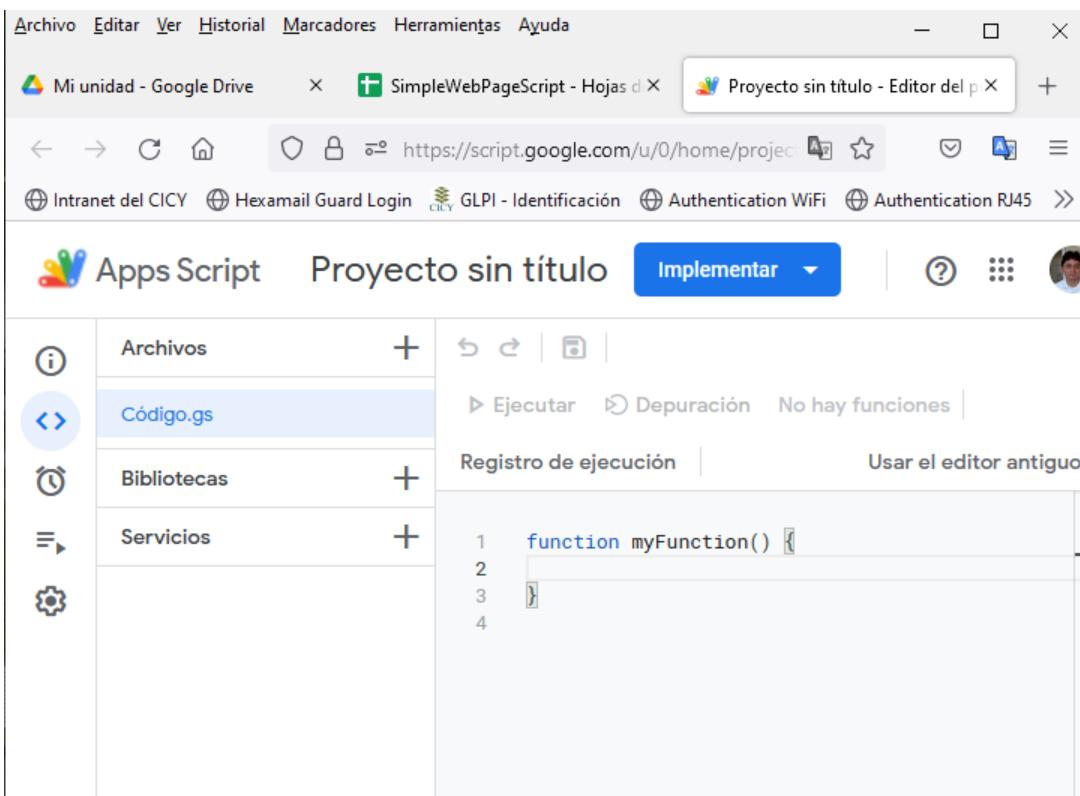
... e inmediatamente se mostrará la hoja de cálculo lista para ser usada. Asigne un nombre presionando en el área marcada, así como se muestra en la siguiente figura:



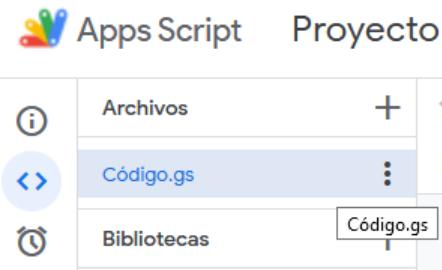
Asigne el nombre SimpleWebPageScript y presione la tecla “ENTER” para guardar los cambios. Seguidamente, ubique la barra de herramientas en la parte superior de la hoja de cálculo y seleccione en secuencia el menú, así como se muestra a continuación:



... y se abrirá una nueva pestaña, así como se muestra a continuación:

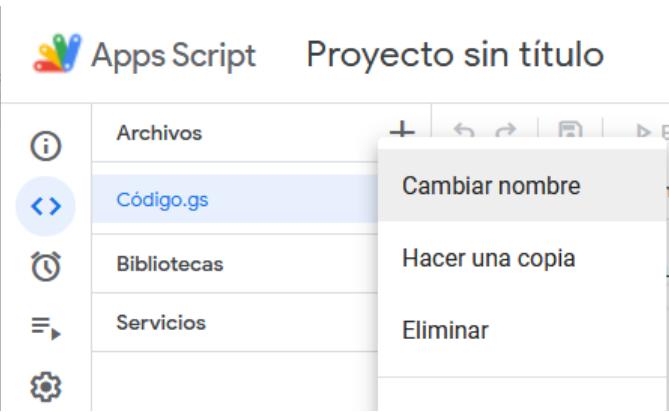


Ubíquese en el campo donde se encuentra el nombre del código y se activará un botón con tres puntitos:


  
**Apps Script Proyecto**

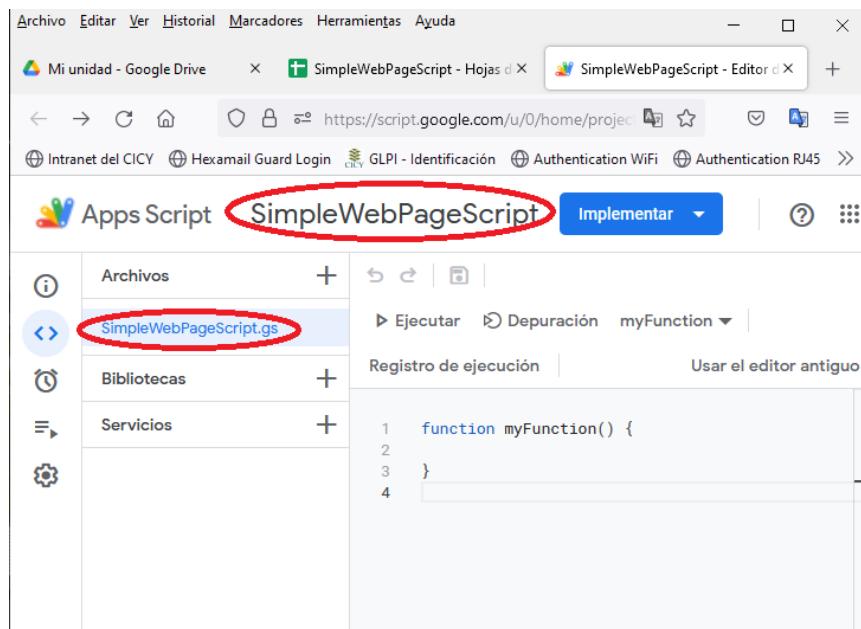
	Archivos	
	Código.gs	
	Bibliotecas	Código.gs

... presione el botón para activar el menú emergente:


  
**Apps Script Proyecto sin título**

- Archivos
  - Código.gs
  - Bibliotecas
  - Servicios
  -
- Cambiar nombre
  - Hacer una copia
  - Eliminar

... y cambie el nombre del código a SimpleWebPageScript, haga lo mismo en la sección del nombre de proyecto y cambie el nombre a SimpleWebPageScript, así como se muestra:



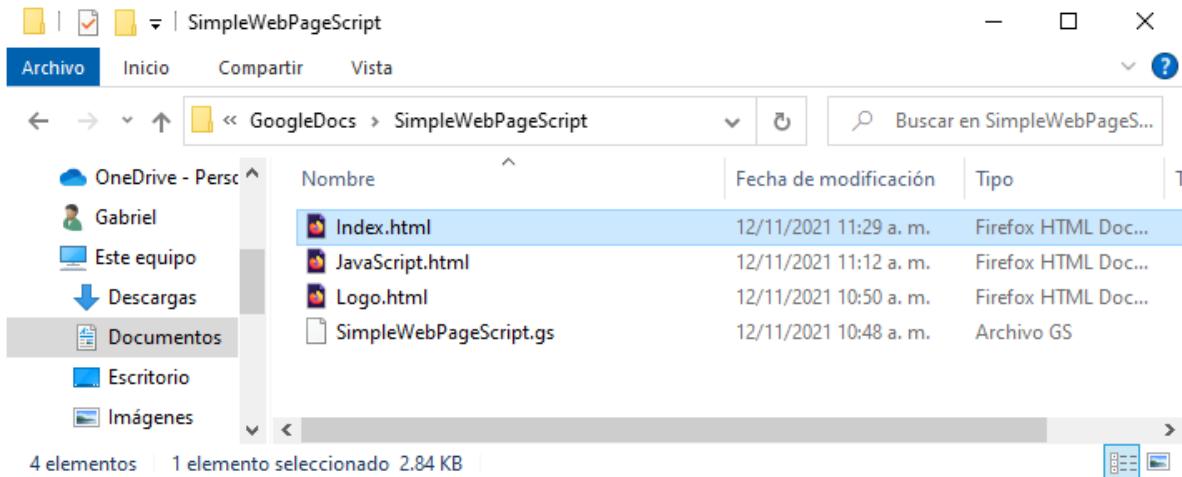
The screenshot shows the Google Apps Script Editor interface. The title bar says "SimpleWebPageScript - Editor". The project name "SimpleWebPageScript" is circled in red at the top. In the left sidebar, the file "SimpleWebPageScript.gs" is also circled in red. The code editor contains the following function:

```

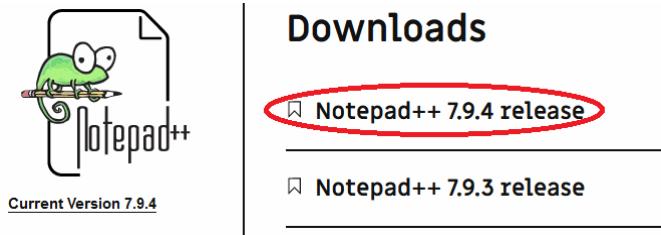
function myFunction() {
}

```

Ahora, regresemos a la carpeta recién descargada y copiada a la carpeta de Documentos, entre a la carpeta GoogleDocs -> SimpleWebPageScript, así como se muestra a continuación:



Por facilidad, se requiere la instalación del programa llamado Notepad++ y lo puede obtener en <https://notepad-plus-plus.org/downloads/>:

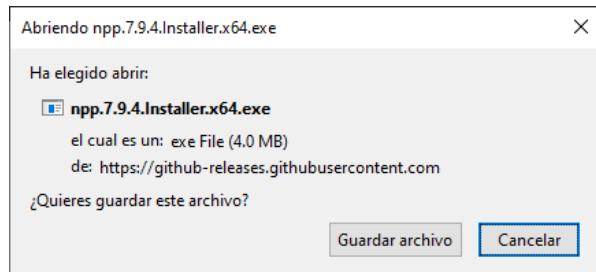


Descargue la versión correspondiente a la versión de su sistema operativo, si desconoce la versión, prueba instalando primero la versión de 64 bits, si falla la instalación, pruebe con la versión de 32 bits:

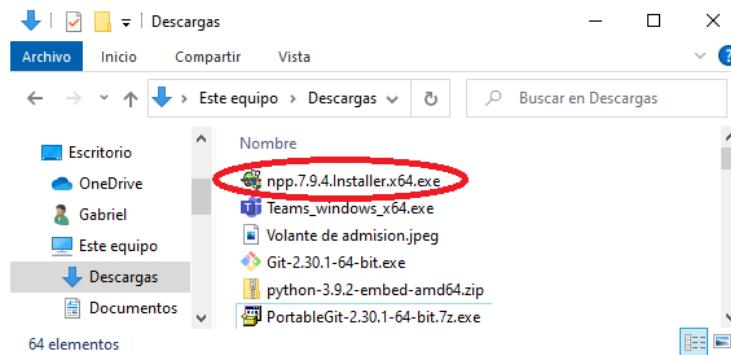
#### Download 64-bit x64

- [Installer | GPG Signature](#)
- [Portable \(zip\) | GPG Signature](#)
- [Portable \(7z\) | GPG Signature](#)
- [Mini-portable \(7z\) | GPG Signature](#)

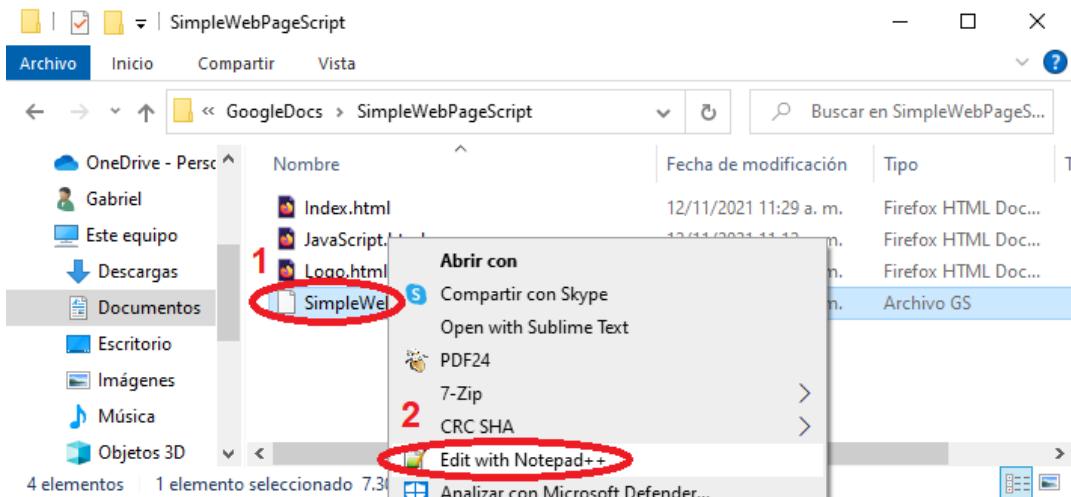
Descargue el archivo:



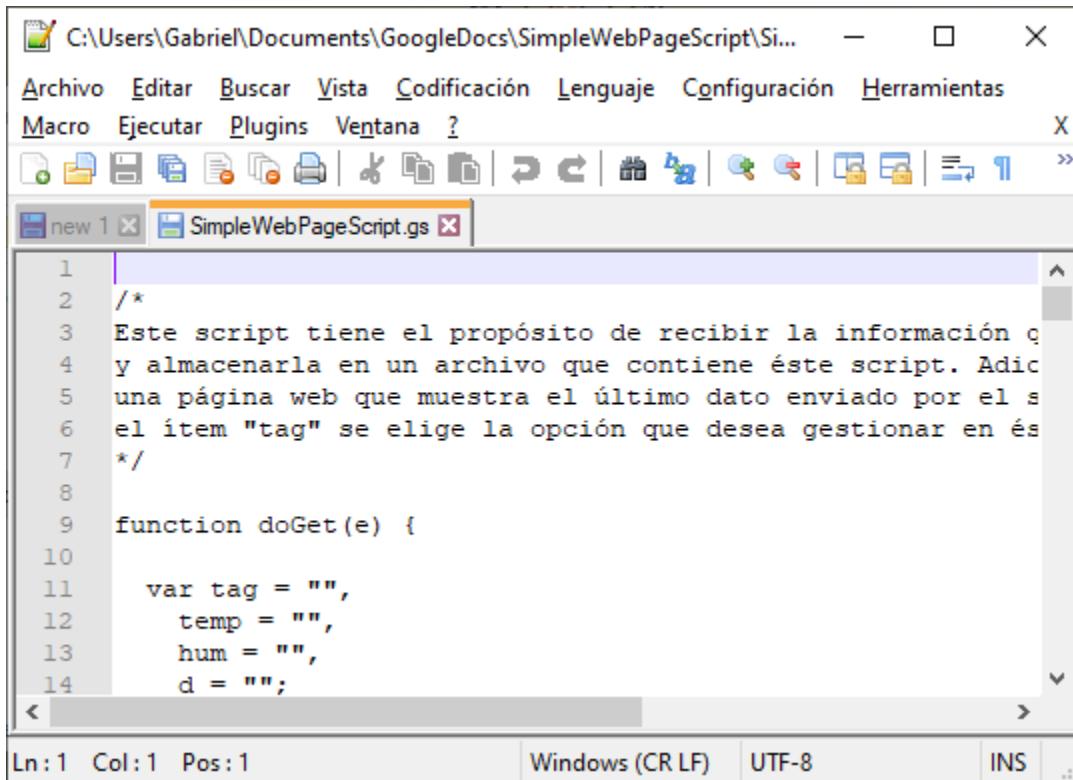
... e instale en su PC con las opciones que trae predeterminadas (sólo presione el botón “siguiente” hasta que el programa se instale):



Una vez instalado el Notepad++, regresamos a la ventana que se muestra donde se ubica el Script de Google, ubique el archivo SimpleWebPageScript y abra con el programa recién instalado. Con el botón derecho del mouse, dé un click sobre el archivo “SimpleWebPageScript.gs” y seleccione la opción que se muestra a continuación:



... e inmediatamente se abrirá una ventana como ésta:



The screenshot shows a code editor window with the following details:

- Title bar: C:\Users\Gabriel\Documents\GoogleDocs\SimpleWebPageScript\Si...
- Menu bar: Archivo, Editar, Buscar, Vista, Codificación, Lenguaje, Configuración, Herramientas, Macro, Ejecutar, Plugins, Ventana, ?
- Toolbar: Includes icons for file operations like Open, Save, Print, and a large set of developer tools.
- Tab bar: new 1 (highlighted) and SimpleWebPageScript.gs
- Code area:

```
1 /*
2  * Este script tiene el propósito de recibir la información q
3  * y almacenarla en un archivo que contiene éste script. Adic
4  * una página web que muestra el último dato enviado por el s
5  * el ítem "tag" se elige la opción que desea gestionar en é
6  */
7
8  function doGet(e) {
9
10    var tag = "",
11        temp = "",
12        hum = "",
13        d = "";
14 }
```
- Status bar: Ln:1 Col:1 Pos:1, Windows (CR LF), UTF-8, INS

... presione las teclas Ctrl + a (Seleccionar todo el texto) y posteriormente Ctrl + c para copiar el texto, ubíquese en el navegador web, borre el contenido SimpleWebPageScript.gs en el navegador web y pegue el contenido copiado, así como se muestra:



The screenshot shows the Google Apps Script editor interface with the following details:

- Left sidebar: Archivos (SimpleWebPageScript.gs selected), Bibliotecas, Servicios.
- Right pane: Script Editor showing the code:

```
1 /*
2  * Este script tiene el propósito de recibir la información que envia el cli
3  * y almacenarla en un archivo que contiene éste script. Adicionalmente, el
4  * una página web que muestra el último dato enviado por el sensor remoto (c
5  * el ítem "tag" se elige la opción que desea gestionar en éste script.
6  */
7
8  function doGet(e) {
9
10    var tag = "",
```

Y presione el botón en forma de diskette para guardar los cambios:

### Apps Script    SimpleWebPageScript



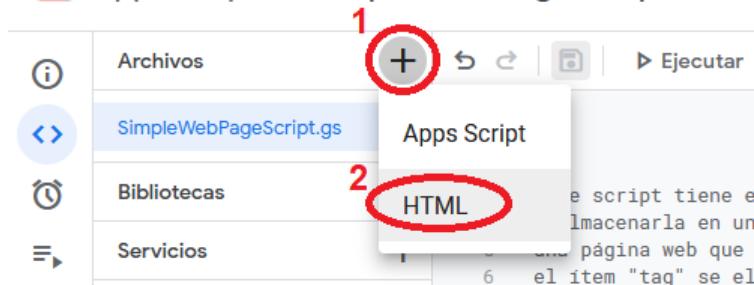
```

1  /*
2   * Este script tiene el propósito de recibir la información que envía el cliente
3   * y almacenarla en un archivo que contiene este script. Adicionalmente, el script
4   * genera una página web que muestra el último dato enviado por el sensor remoto (el
5   * item "tag" se elige la opción que desea gestionar en este script).
6   */
7
8
9  function doGet(e) {
10
11    var tag = "";

```

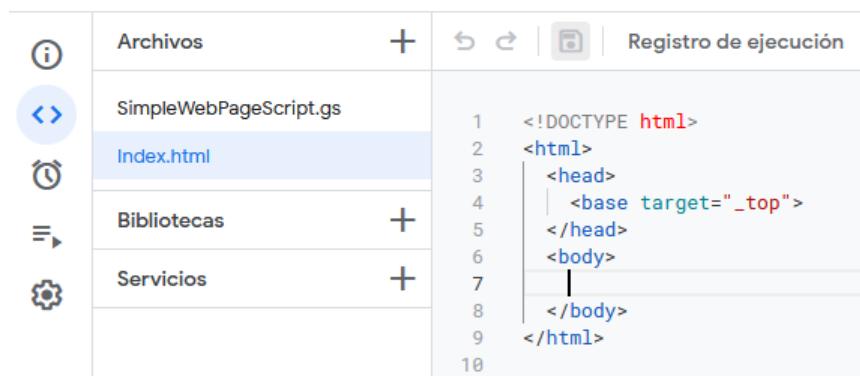
... seguidamente, presione el botón "+" para agregar el siguiente archivo:

### Apps Script    SimpleWebPageScript



... ponga el nombre "Index" (sin .html) con la primera letra mayúscula y presione la tecla "Enter" para guardar el nombre (el .html se agrega automáticamente):

### Apps Script    SimpleWebPageScript



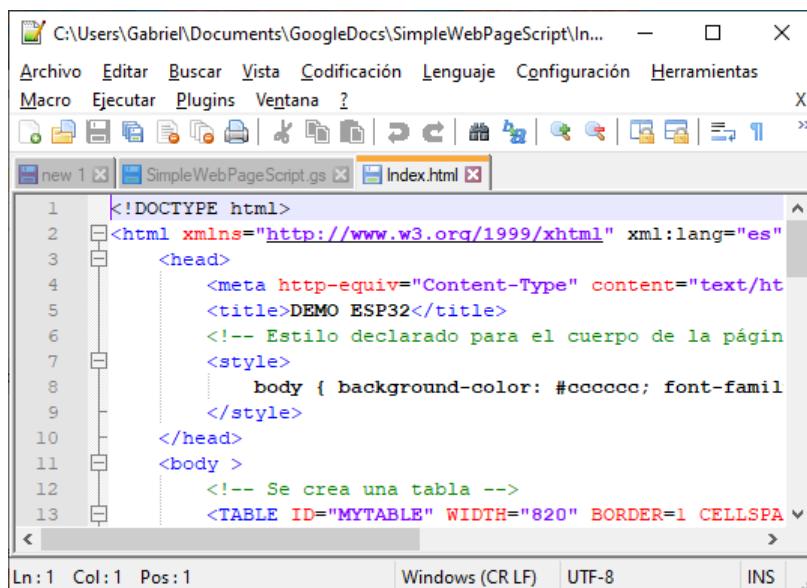
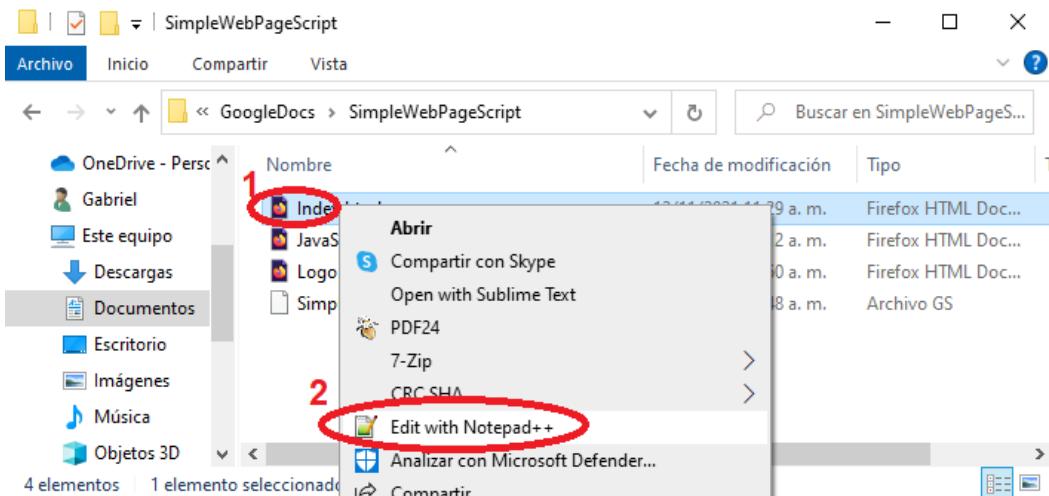
```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <base target="_top">
5    </head>
6    <body>
7
8    </body>
9
10   </html>

```

... al igual que en el caso anterior, borre el contenido de Index.html en su página

web, abra el archivo Index.html con el Notepad++ ubicado en su carpeta recién descargada, así como se muestra:



... presione las teclas Ctrl + a (Seleccionar todo el texto) y posteriormente Ctrl + c para copiar el texto, ubíquese en el archivo Index.html en el navegador web (NO olvide borrar el contenido previo de Index.html en el navegador web) y pegue el contenido copiado, así como se muestra:

### Apps Script    SimpleWebPageScript

	Archivos	+	1   <!DOCTYPE html> 2   <html xmlns="http://www.w3.org/1999/xhtml"> 3       <head> 4           <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" /> 5           <title>DEMO ESP32</title> 6           <!-- Estilo declarado para el documento --> 7           <style> 8               body { background-color: #f0f0f0; } 9       </style>	Registro de ejecución
<span style="font-size: 1.5em;">i</span> <span style="font-size: 1.5em;">&lt;&gt;</span> <span style="font-size: 1.5em;">🕒</span> <span style="font-size: 1.5em;">≡</span> <span style="font-size: 1.5em;">⚙</span>	<span style="font-size: 1.5em;">i</span> SimpleWebPageScript.gs <span style="font-size: 1.5em;">&lt;&gt;</span> Index.html <span style="font-size: 1.5em;">🕒</span> <span style="font-size: 1.5em;">≡</span> <span style="font-size: 1.5em;">⚙</span>	+    +		

Y presione el botón en forma de diskette para guardar los cambios:

### Apps Script    SimpleWebPageScript

	Archivos	+	1   <!DOCTYPE html> 2   <html xmlns="http://www.w3.org/1999/xhtml"> 3       <head> 4           <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" /> 5           <title>DEMO ESP32</title> 6           <!-- Estilo declarado para el documento --> 7           <style> 8               body { background-color: #f0f0f0; } 9       </style>	Registro de ejecución
<span style="font-size: 1.5em;">i</span> <span style="font-size: 1.5em;">&lt;&gt;</span> <span style="font-size: 1.5em;">🕒</span> <span style="font-size: 1.5em;">≡</span> <span style="font-size: 1.5em;">⚙</span>	<span style="font-size: 1.5em;">i</span> SimpleWebPageScript.gs <span style="font-size: 1.5em;">&lt;&gt;</span> Index.html <span style="font-size: 1.5em;">🕒</span> <span style="font-size: 1.5em;">≡</span> <span style="font-size: 1.5em;">⚙</span>	+    +		

Repita estos pasos hasta subir todos los archivos al proyecto. Debe quedar, así como se muestra:

### Apps Script    SimpleWebPageScript

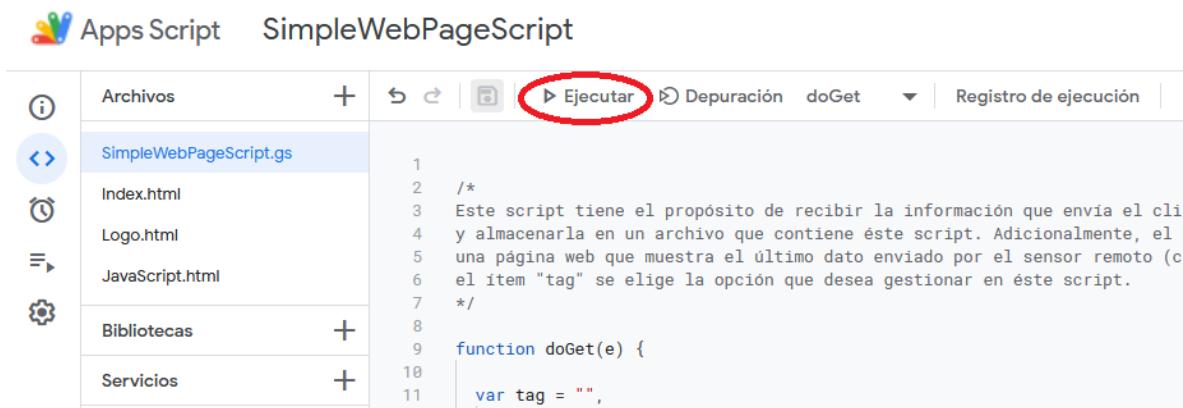
	Archivos	+	1   /* 2   * Este script tiene el propósito de recibir la información que envía el cliente y almacenarla en un archivo que contiene este script. Adicionalmente, el script genera una página web que muestra el último dato enviado por el sensor remoto (el item "tag" se elige la opción que desea gestionar en este script). 3   */ 4   function doGet(e) { 5     var tag = ""; 6   }	Registro de ejecución
<span style="font-size: 1.5em;">i</span> <span style="font-size: 1.5em;">&lt;&gt;</span> <span style="font-size: 1.5em;">🕒</span> <span style="font-size: 1.5em;">≡</span> <span style="font-size: 1.5em;">⚙</span>	<span style="font-size: 1.5em;">i</span> SimpleWebPageScript.gs <span style="font-size: 1.5em;">&lt;&gt;</span> Index.html <span style="font-size: 1.5em;">🕒</span> <span style="font-size: 1.5em;">≡</span> <span style="font-size: 1.5em;">⚙</span>	+    +		

En todos los casos, cuando se crea un nuevo archivo html en la Apps Script de

Google ya incluye un contenido previo. Ese contenido previo se requiere borrarlo antes de insertar el código recién copiado del Notepad++. **NO importa el orden en que se crean los archivos, lo importante es que todos los nombres de los archivos se escriban tal como se muestran (respetando mayúsculas y minúsculas).** La extensión de archivo (.html) es agregado automáticamente al momento de presionar la tecla “ENTER” para guardar el nombre. Con los archivos ya completos, se requieren dos pasos para gestionar los permisos de ejecución:

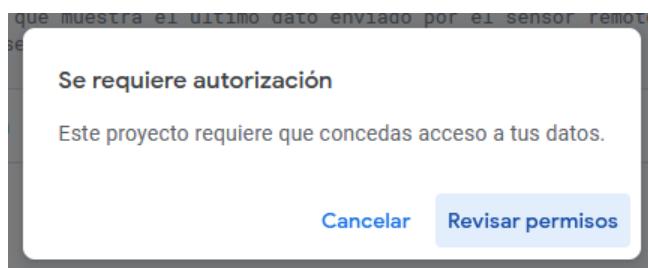
**1er Paso:**

Seleccione el archivo SimpleWebPage.gs y presione el botón de ejecutar, así como se muestra:

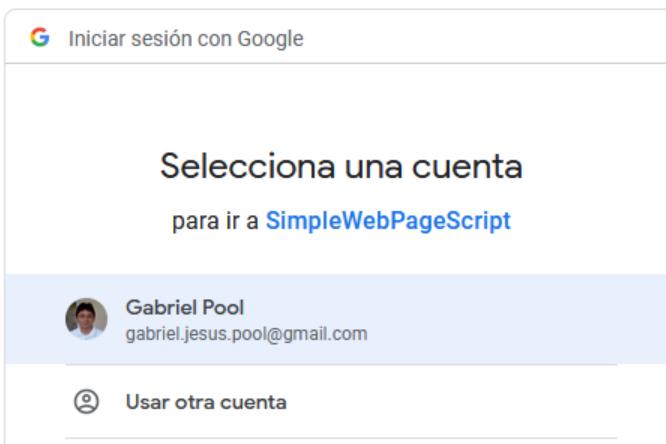


```
1  /*
2   * Este script tiene el propósito de recibir la información que envía el cliente
3   * y almacenarla en un archivo que contiene este script. Adicionalmente, el script
4   * genera una página web que muestra el último dato enviado por el sensor remoto (cada
5   * vez que el item "tag" se elige la opción que desea gestionar en este script).
6   */
7
8
9  function doGet(e) {
10
11    var tag = "",
```

... e inmediatamente se oscurecerá la pantalla y saldrá un mensaje como éste:



... presione el botón “Revisar Permisos”:



... presione en el área donde se encuentra su logo y correo e inmediatamente saldrá una ventana como ésta:

Inicia sesión: Cuentas de Google - Mozilla Firefox

https://accounts.google.com/signin/oauth/danger?authuser=0&part=AJi8hAMn \*\*\*

**!**

**Google no ha verificado esta aplicación**

La aplicación está solicitando acceso a información sensible de tu cuenta de Google. No deberías utilizar esta aplicación hasta que el desarrollador ([gabriel.jesus.pool@gmail.com](mailto:gabriel.jesus.pool@gmail.com)) la verifique con Google.

[Configuración avanzada](#)  [VOLVER AL MODO SEGURO](#)

... presione el hipertexto que se encuentra marcado con una flecha en la figura anterior y saldrá otra ventana como ésta:



**Google no ha verificado esta aplicación**

La aplicación está solicitando acceso a información sensible de tu cuenta de Google. No deberías utilizar esta aplicación hasta que el desarrollador ([gabriel.jesus.pool@gmail.com](mailto:gabriel.jesus.pool@gmail.com)) la verifique con Google.

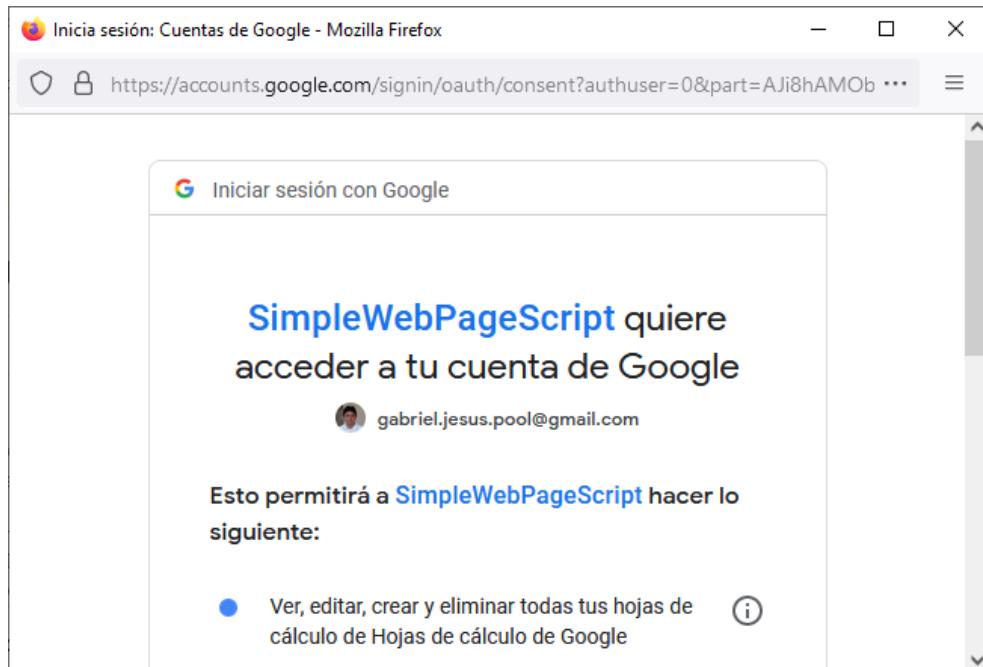
[Ocultar configuración avanzada](#)

[VOLVER AL MODO SEGURO](#)

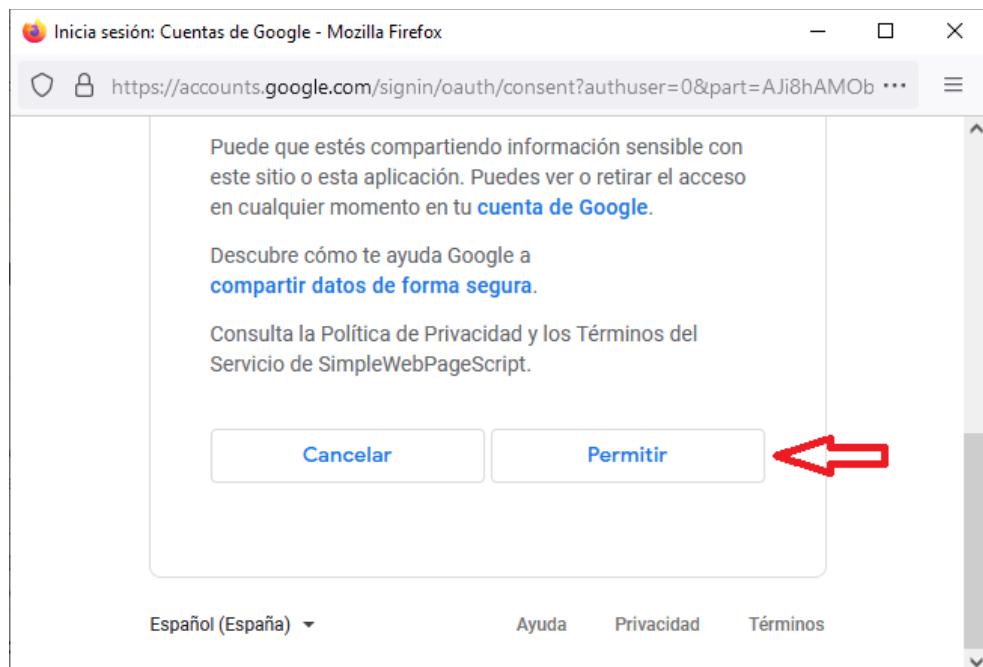
Continúa solo si entiendes los riesgos que esto conlleva y confías en el desarrollador ([gabriel.jesus.pool@gmail.com](mailto:gabriel.jesus.pool@gmail.com)).

[Ir a SimpleWebPageScript \(no seguro\)](#) 

... nuevamente, presione el hipertexto marcado en la figura anterior y saldrá otra ventana como ésta:

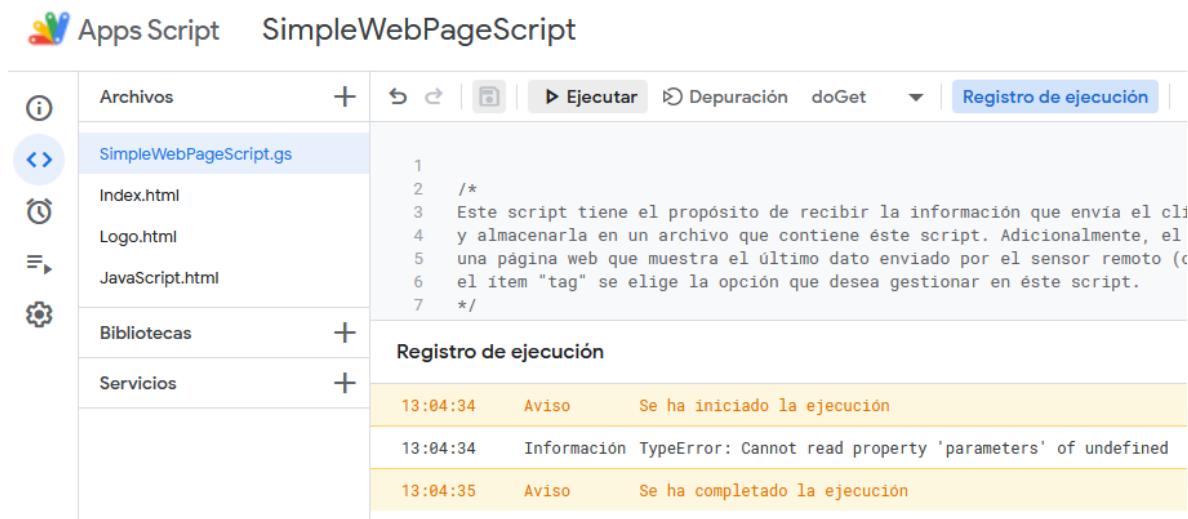


... deslice hasta encontrar el botón “Permitir” y presione el botón:



La ventana anterior se cerrará, pero si se toma demasiado tiempo en seguir estos pasos, en el código de Google mostrará un mensaje indicando que requiere permisos de ejecución, basta con presionar el botón de “ejecutar” y el código se ejecutará sin abrir la ventana para la gestión de permisos. Al ejecutar el código se

mostrará un mensaje:



The screenshot shows the Google Apps Script interface. On the left, there's a sidebar with icons for Archivos, Bibliotecas, and Servicios. The main area shows a script named "SimpleWebPageScript.gs" with the following code:

```

1
2  /*
3  Este script tiene el propósito de recibir la información que envia el cli
4  y almacenarla en un archivo que contiene éste script. Adicionalmente, el
5  una página web que muestra el último dato enviado por el sensor remoto (c
6  el ítem "tag" se elige la opción que desea gestionar en éste script.
7 */

```

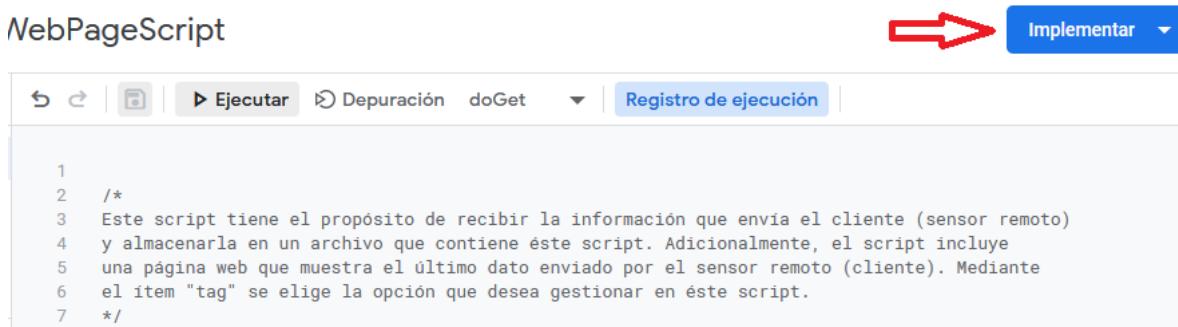
Below the code, there's a "Registro de ejecución" (Execution Log) section with the following entries:

- 13:04:34 Aviso Se ha iniciado la ejecución
- 13:04:34 Información TypeError: Cannot read property 'parameters' of undefined
- 13:04:35 Aviso Se ha completado la ejecución

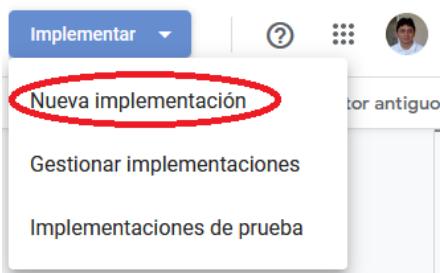
Eso indica que el código se ha ejecutado exitosamente.

## 2º Paso:

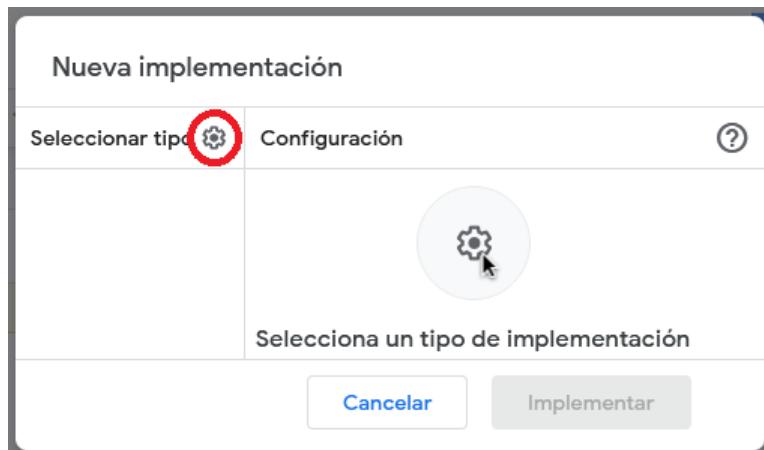
Ahora se ubica en la parte superior izquierda de su navegador web y encontrará un botón azul llamado “Implementar”, presione dicho botón, así como se muestra:



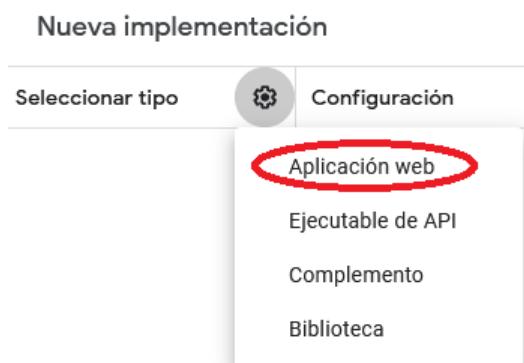
... e inmediatamente se abrirá una ventana como esta:



Presione la opción “Nueva Implementación”, y saldrá una ventana, así como se muestra:



... presione el botón con forma de engrane y saldrá un menú emergente:



... seleccione la opción “Aplicación Web” y se mostrará lo siguiente:

Nueva implementación

Seleccionar tipo	Configuración	(?)
Aplicación web	<p>Descripción</p> <p>Nueva descripción</p> <p>Aplicación web</p> <p>Ejecutar como</p> <p>Yo (gabriel.jesus.pool@gmail.com)</p> <p>Quién tiene acceso</p> <p>Solo yo</p> <p>También se puede usar como biblioteca. <a href="#">Más información</a></p>	
		<p>Cancelar</p> <p>Implementar</p>

... en el campo nueva descripción ponga la versión de su proyecto, en el campo aplicación web, no haga cambios, y en el último campo llamado “Quien tiene acceso”, seleccione “Cualquier usuario”, los campos deben quedar, así como se muestra:

Nueva implementación

Seleccionar tipo	Configuración	?
Aplicación web	<p>Descripción</p> <p>Nueva descripción</p> <p>version 1</p> <p>Aplicación web</p> <p>Ejecutar como</p> <p>Yo (gabriel.jesus.pool@gmail.com)</p> <p>Quién tiene acceso</p> <p>Cualquier usuario</p> <p>También se puede usar como biblioteca.<a href="#">Más información</a></p>	

[Cancelar](#) [Implementar](#)

... seguidamente, presione el botón azul llamado “Implementar” y le saldrá una ventana como esta:

Nueva implementación

La implementación se ha actualizado correctamente.

Versión 1 del 12 nov, 13:10

ID de implementación

AKfycbwjptz0UF7K40meD-Fknu31fukJ-I4qDs9YCTvxpsQDJw2PAIIICqgdgpaVikvdVySRHw

 Copiar

Aplicación web

URL

<https://script.google.com/macros/s/AKfycbwjptz0UF7K40meD-Fknu31fukJ-I4qDs9YCTvxpsQDJw2PAIIICqgdgpaVi...>

 Copiar

Hecho

Presione el botón marcado en la figura anterior para copiar el Id del proyecto, este es necesario obtenerlo para actualizar el archivo JavaScript y posteriormente implementarlo en el módulo NodeMCU. Presione el botón “Hecho” para cerrar la ventana anterior, y ahora se requiere actualizar el archivo JavaScript como sigue:

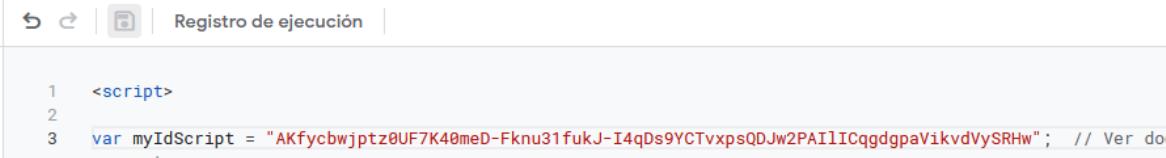
Apps Script    SimpleWebPageScript

	Archivos		  	Registro de ejecución
	SimpleWebPageScript.gs	1    <script> 2 3    var myIdScript = "myIdScript"; // Ver documentación del proyecto 4		
	Index.html			
	Logo.html			

... ubique la línea 3 y sustituya el texto “myIdScript” por el Id del proyecto que se acaba de copiar. No olvide que el Id del proyecto también va entre comillas:

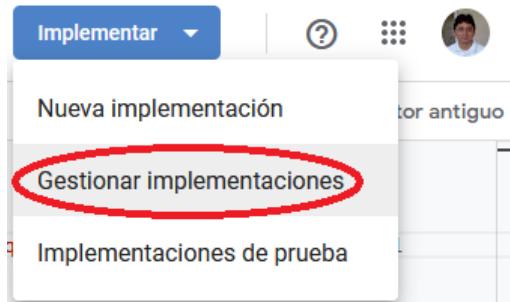
WebPageScript

Implementar ▾



```
1 <script>
2
3 var myIdScript = "AKfyccbwpjt0UF7K40meD-Fknu31fukJ-I4qDs9YCTvxpsQDJw2PAI1ICqgdgpaVikvdVySRHw"; // Ver doc
proyecto
```

Asegúrese de guardar los cambios presionando el botón del diskette y vuelva a presionar el botón azul de “Implementar” para crear una segunda versión del proyecto, ya que, la primera versión NO va a funcionar ya que, es imprescindible el Id del proyecto:



... y se abrirá una ventana como esta:

Gestionar implementaciones

Activa	Configuración
version 1	<small>versión</small> Versión 1 del 12 nov, 13:29
Archivadas	<small>Descripción</small> version 1
No hay implementaciones archivadas	
	<small>ID de implementación</small> AKfyccb... <span style="color: blue;">Copiar</span>
	<small>Aplicación web</small> <small>URL</small> <a href="https://script.google.com/macros/s/AKfyccb...">https://script.google.com/macros/s/AKfyccb...</a> <span style="color: blue;">Copiar</span>

Cancelar
Implementar

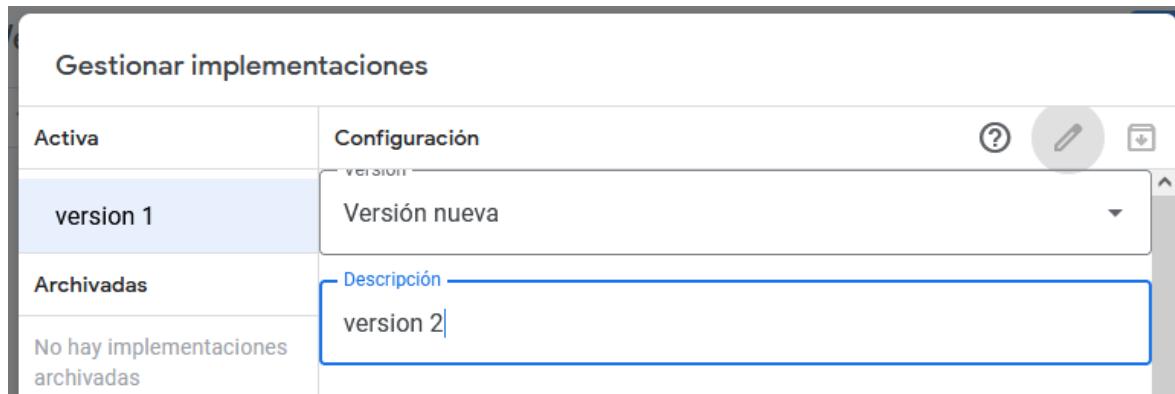
Presione el botón marcado en la siguiente imagen, así como se muestra:

Gestionar implementaciones

Activa	Configuración
version 1	<small>versión</small> Versión 1 del 12 nov, 13:29

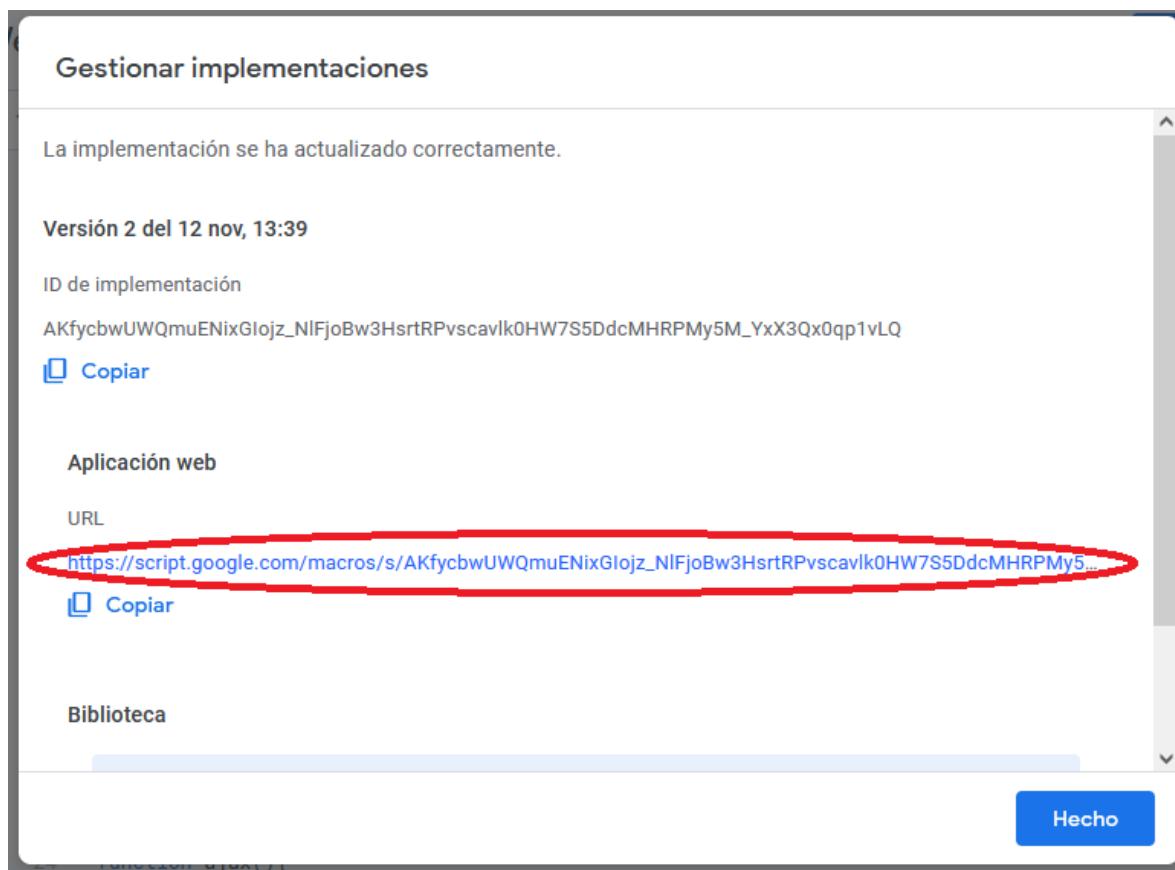
Editar
↓

... e inmediatamente se desbloquean los campos configuración y descripción, seleccione la opción y llene el segundo campo, así como se muestra:



Activa	Configuración
version 1	<div style="border: 1px solid #ccc; padding: 5px;"> <span>Versión</span>             Versión nueva         </div>
Archivadas	<div style="border: 1px solid #ccc; padding: 5px;"> <span>Descripción</span>             version 2         </div>
No hay implementaciones archivadas	

... presione el botón azul llamado “Implementar” que se encuentra en la parte inferior derecha y la ventana mostrará lo siguiente:



Gestionar implementaciones

La implementación se ha actualizado correctamente.

Versión 2 del 12 nov, 13:39

ID de implementación  
AKfycbwUWQmuENixGlojz\_NlFjoBw3HsrtRPvscavlk0HW7S5DdcMHRPMy5M\_YxX3Qx0qp1vLQ

Aplicación web

URL  
[https://script.google.com/macros/s/AKfycbwUWQmuENixGlojz\\_NlFjoBw3HsrtRPvscavlk0HW7S5DdcMHRPMy5...](https://script.google.com/macros/s/AKfycbwUWQmuENixGlojz_NlFjoBw3HsrtRPvscavlk0HW7S5DdcMHRPMy5...)

Biblioteca

... presione el vínculo marcado en la figura anterior y si todo salió bien, en una nueva pestaña, observará lo siguiente:



# Centro de Investigación Científica de Yucatán A. C.

## Departamento de Instrumentación

[www.cicy.mx](http://www.cicy.mx)

The screenshot shows a web browser window with the URL <https://script.google.com/macros/s/AKfycbwUWQmuE>. The page content is as follows:

**Centro de Investigación Científica de Yucatán A.C.  
Departamento de Instrumentación 2021**

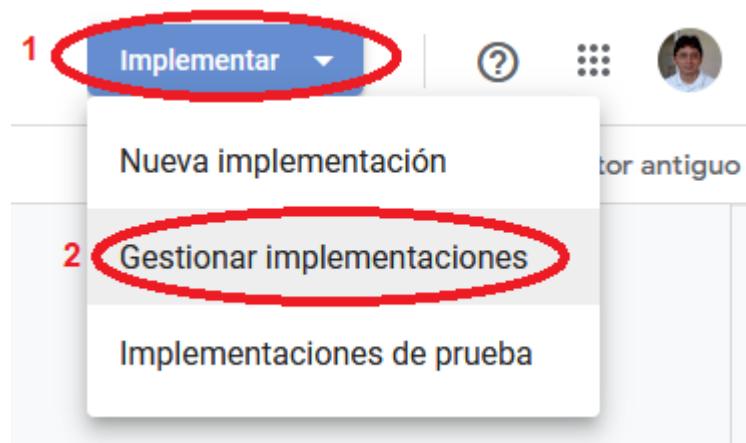
Temperatura:	Humedad:
Último dato en:	
Hora Local <b>12/11/2021 12:40:47</b>	

No se podrá observar dato alguno, puesto que el sensor no ha sido implementado. En el último punto de este capítulo titulado: “El Cliente para Google Apps Script implementado en NodeMCU-32S” se implementará el código del dispositivo IoT.

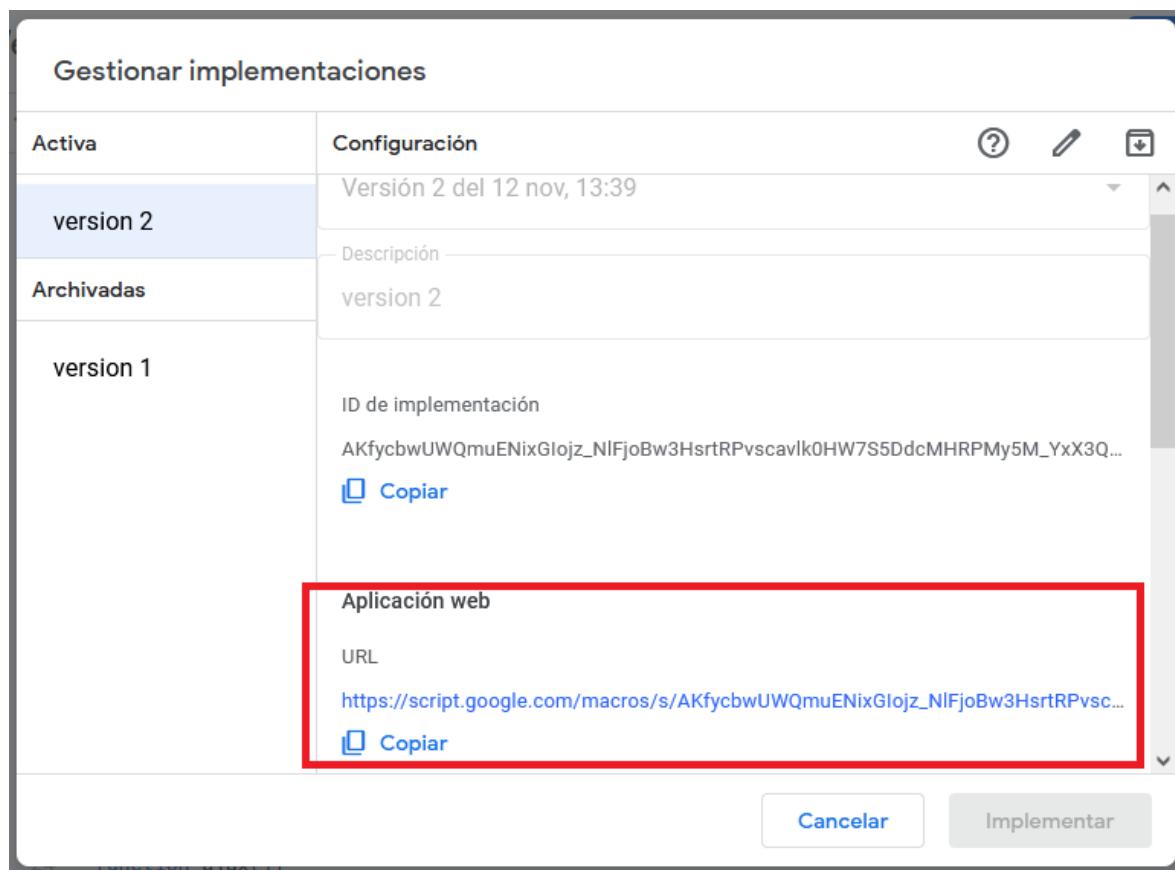
Si no puede observar todo lo mostrado anteriormente en la página web (incluyendo la hora del servidor) indica que ocurrió un problema en la implementación. No es necesario borrar el proyecto, trate de verificar los pasos anteriores y de encontrar el error, solamente siga los pasos a partir de donde se copió el Id del proyecto para hacer una nueva implementación.

**IMPORTANTE:** Por cada cambio realizado en el proyecto es necesario gestionar la implementación y realizar una nueva implementación, ya que, de no hacerlo (sólo guarda los cambios) el último proyecto implementado es el que seguirá funcionando.

Para obtener el link donde se ubica la página web, se parte del proyecto ya implementado y ubicando el botón azul llamado “Implementar”, siga la siguiente secuencia:



Al presionar la opción anterior, inmediatamente se abrirá una ventana como ésta:



Gestionar implementaciones	
Activa	Configuración
version 2	Versión 2 del 12 nov, 13:39
Archivadas	Descripción version 2
version 1	ID de implementación AKfycbwUWQmuENixGlojz_NlFjoBw3HsrtRPvscavlk0HW7S5DdcMHRPMy5M_YxX3Q... <a href="#" style="color: blue;">Copiar</a>
<b>Aplicación web</b> URL <a href="https://script.google.com/macros/s/AKfycbwUWQmuENixGlojz_NlFjoBw3HsrtRPvsc..." style="color: blue;">Copiar</a>	

Si usted presiona el hipertexto “Copiar” el link quedará en el portapapeles y Ud podrá compartirlo mediante mensajería instantánea o correo electrónico.

Otro detalle adicional que es muy notable es el encabezado que Google le agrega al proyecto con el fin de deslinde de responsabilidades. Hay que destacar que este proyecto utiliza intercambio de datos entre el servidor y el cliente, así que, Google sólo permite el envío de archivos que se agrega al proyecto, en este caso: index.html, JavaScript.html y Logo.html en formato html, svg y JavaScript (todos embebidos en el archivo html). Para el response admite los formatos xml, JSON y texto plano. Google NO permite el intercambio de fotos, pdf, ni cualquier otro archivo almacenado en el Drive de Google (se activa la directiva CORS y las bloquea).

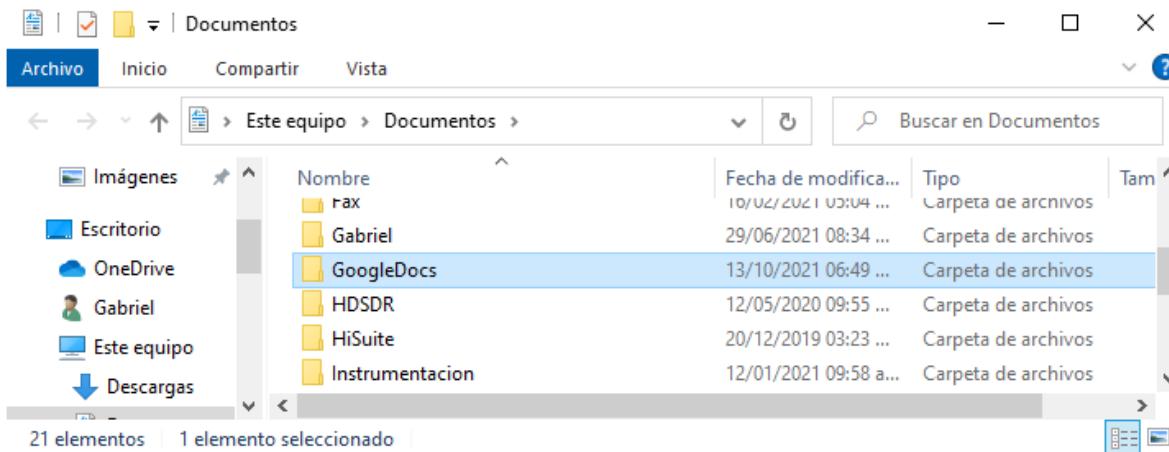
En la red existe una amplia ayuda para desarrollar en esta plataforma, adicionalmente que JavaScript es un lenguaje muy conocido.

En el último punto de este capítulo titulado: “El Cliente para Google Apps Script implementado en NodeMCU-32S” se implementará el código del dispositivo IoT.

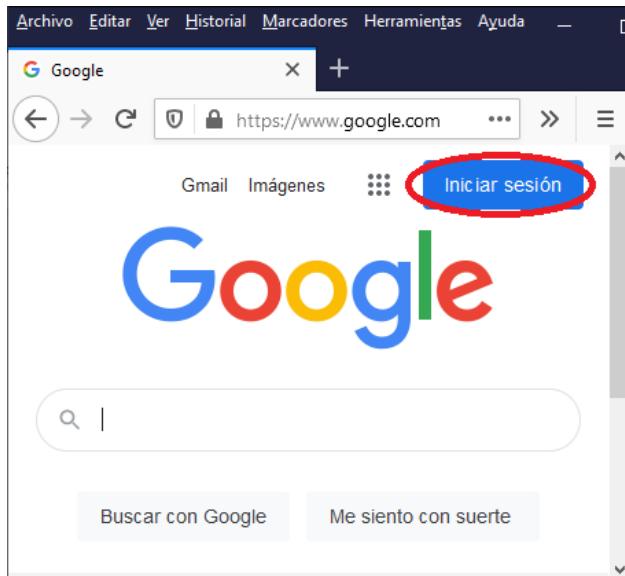
### 3.3.3 Pagina web con gráfica y descarga de históricos en Google Apps Script

**Objetivo específico:** Configurará e implementará los servicios necesarios para mostrar una página web dinámica con gráficos y descarga de históricos usando el lenguaje JavaScript.

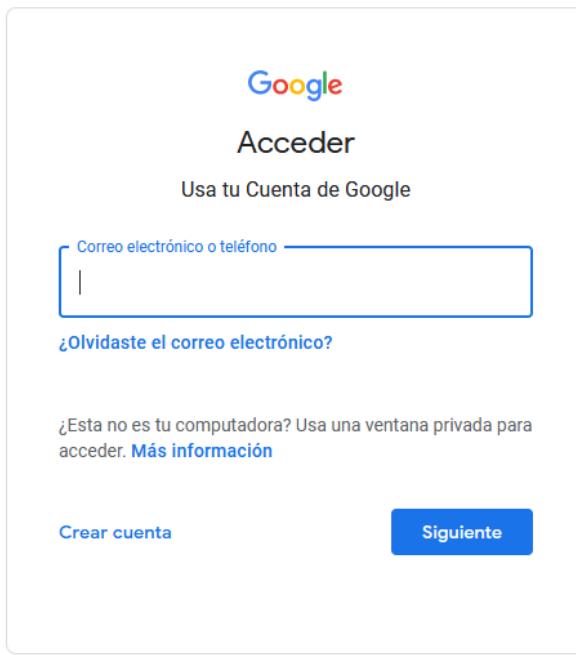
En la actualidad, las herramientas de Google han ganado popularidad y con el crecimiento de los dispositivos IoT, Google no se queda atrás, ha implementado una herramienta llamada Google Apps Script el cual, está basado en el lenguaje JavaScript. Dicho lenguaje es de alto nivel y permite la implementación rápida de soluciones. El JavaScript es ampliamente usado en las páginas web para gestionar servicios en segundo plano. En el sitio <https://developers.google.com/apps-script> se puede encontrar la ayuda necesaria para la implementación de soluciones, este proyecto se enfocará a las WebApps y puede encontrar la documentación en este sitio: <https://developers.google.com/apps-script/guides/web>. Como dice un dicho muy popular “Para muestra basta un botón”. Para implementar este proyecto se requiere de la descarga de los archivos del proyecto ubicado en <https://github.com/gpoolb/ESP32>. Descargue los archivos ubique la carpeta llamada “GoogleDocs” y copie la carpeta en Mis Documentos, así como se muestra:



Seguidamente, se parte del hecho que Ud ya tiene una cuenta en Google. En la página principal de Google, presione el botón “Iniciar sesión”:



Introduzca su nombre de usuario:



Google

Acceder

Usa tu Cuenta de Google

Correo electrónico o teléfono \_\_\_\_\_

[¿Olvidaste el correo electrónico?](#)

¿Esta no es tu computadora? Usa una ventana privada para acceder. [Más información](#)

[Crear cuenta](#) [Siguiente](#)

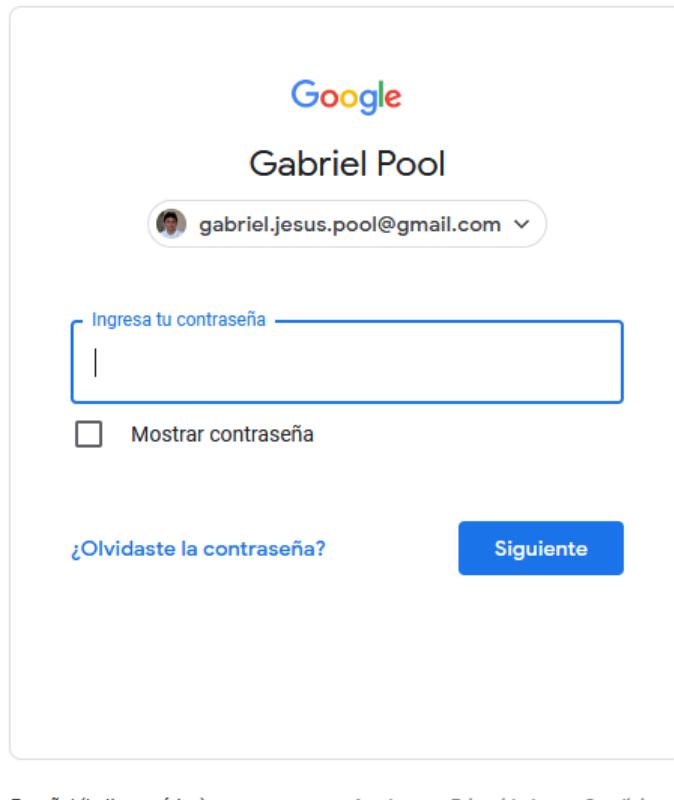
Español (Latinoamérica) ▾

Ayuda

Privacidad

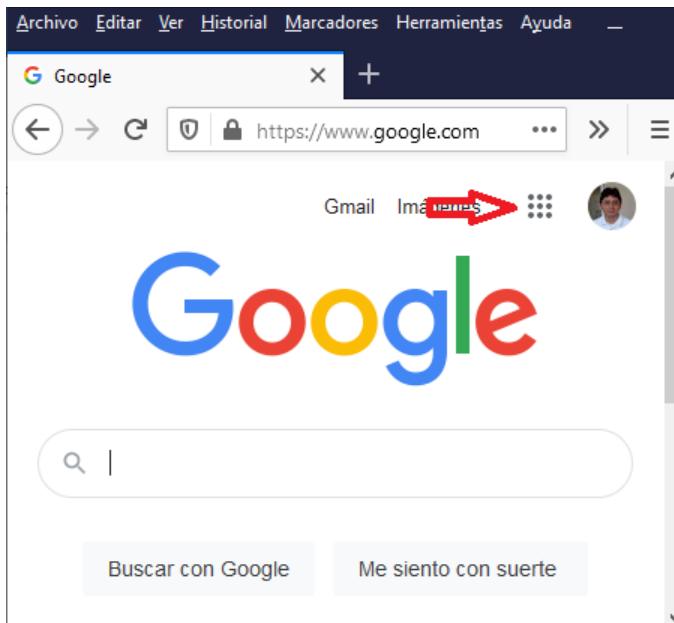
Condiciones

y contraseña:



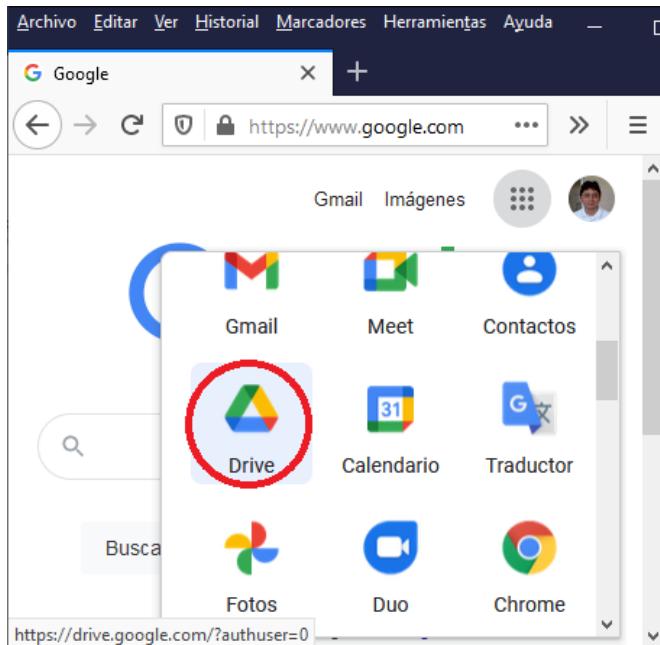
Español (Latinoamérica) ▾ Ayuda Privacidad Condiciones

... e inmediatamente se mostrará una ventana como ésta:

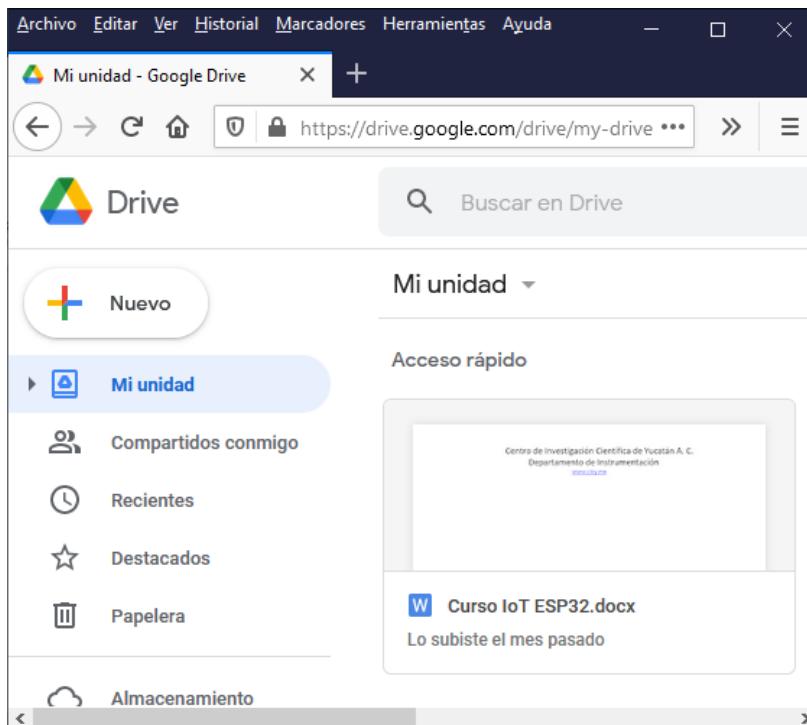


Presione el botón con forma de puntitos que se señala en la figura anterior con

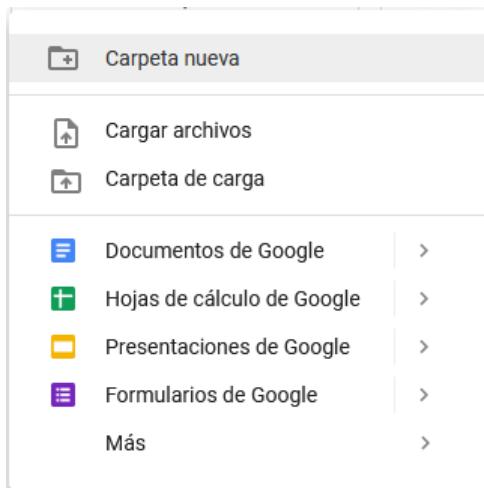
una flecha roja y observará lo siguiente:



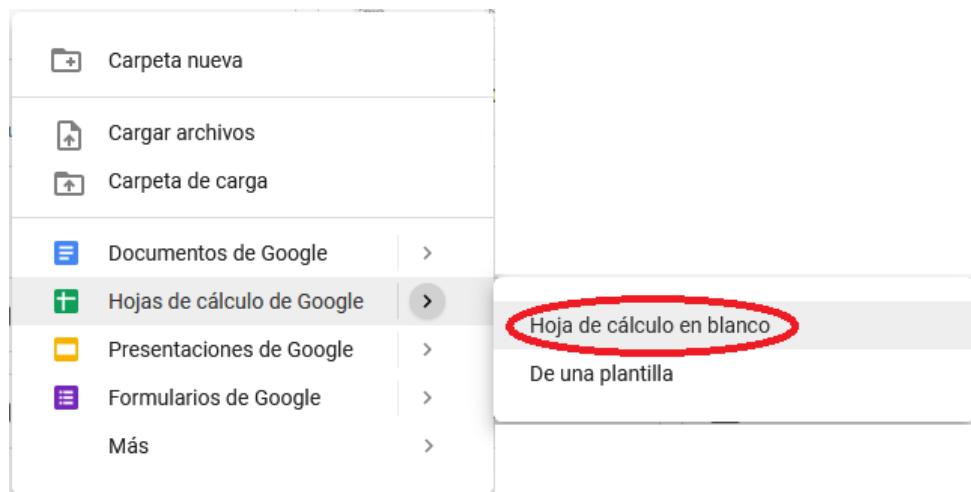
Presione el botón Drive, así como se muestra en la figura anterior y se abrirá otra ventana:



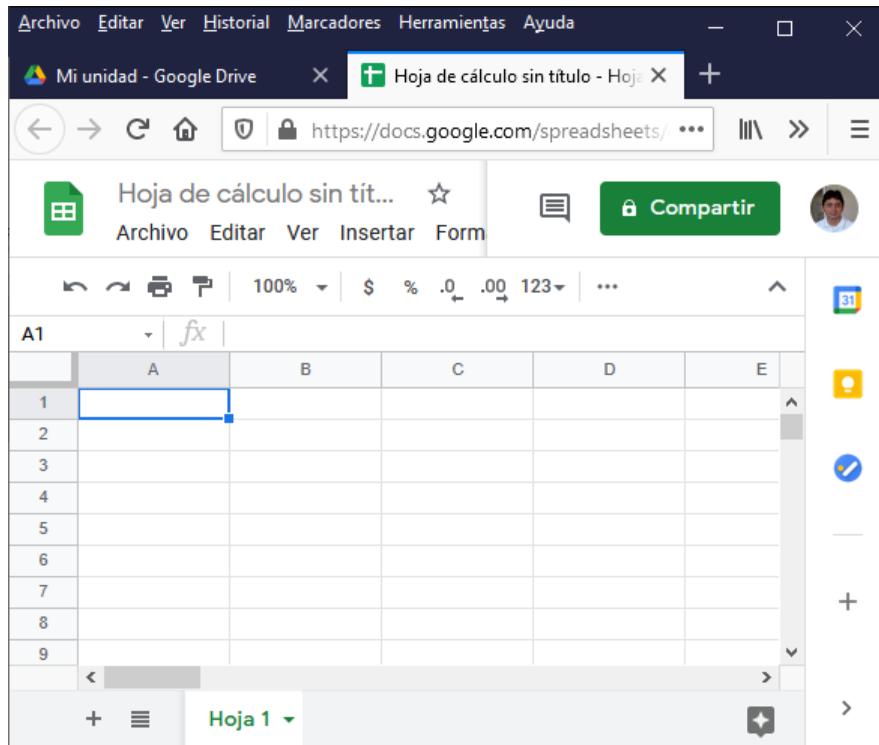
Ubique un área en blanco del lado derecho de la página web (en la sección de archivos o carpetas, NO ponga el ratón encima de las carpetas o archivos) y presione el botón derecho del ratón y le saldrá una ventana como ésta:



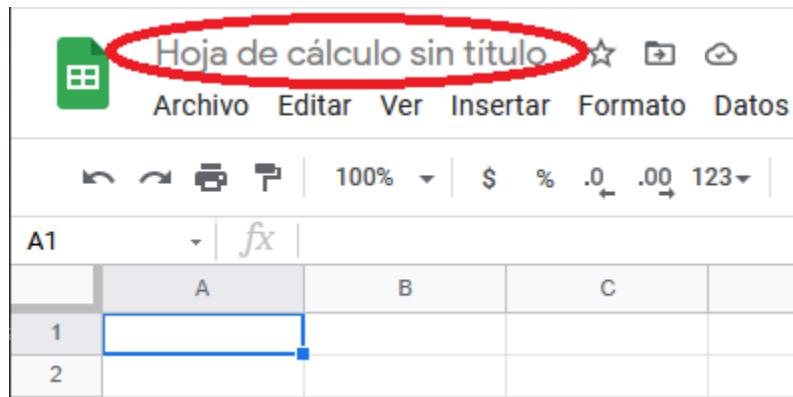
Ubíquese en el área de “Hojas de cálculo de Google” y saldrá un menú, así como se muestra a continuación:



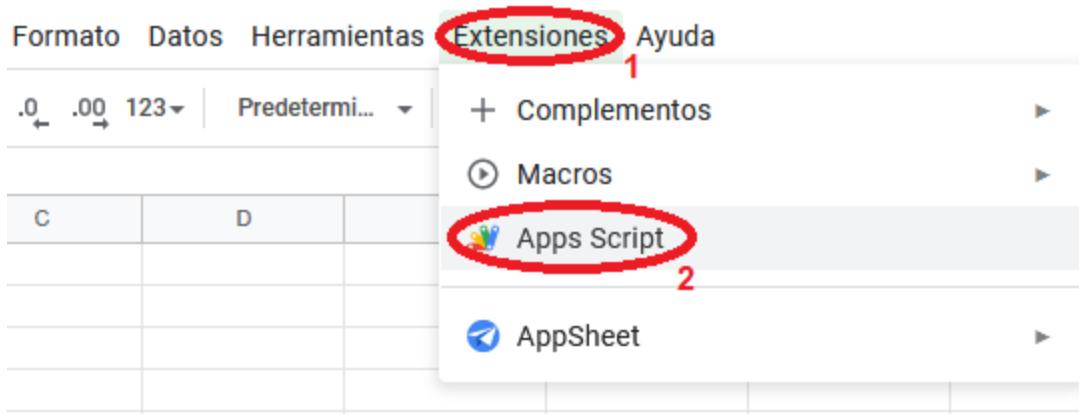
... y seleccione Hoja de cálculo en blanco:



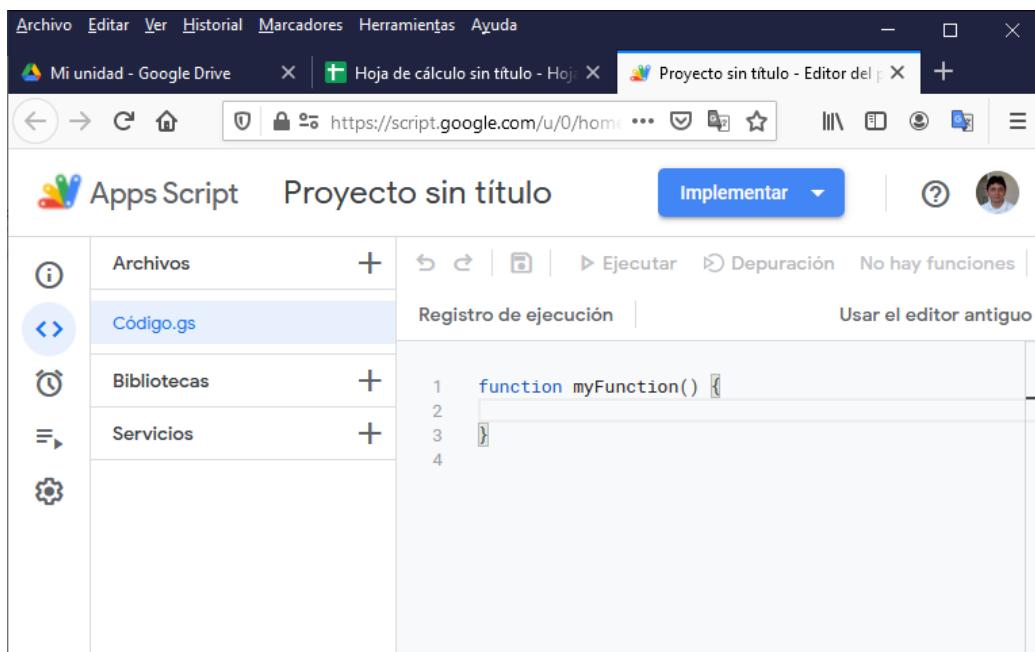
... e inmediatamente se mostrará la hoja de cálculo lista para ser usada. Asigne un nombre presionando en el área marcada, así como se muestra en la siguiente figura:



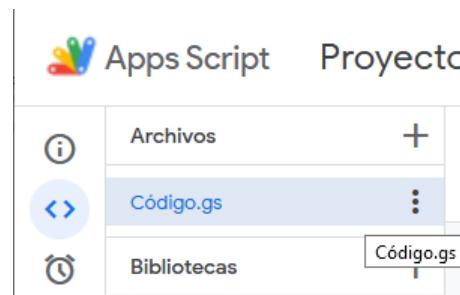
Asigne el nombre DHT22 y presione la tecla "ENTER" para guardar los cambios. Seguidamente, ubique la barra de herramientas en la parte superior de la hoja de cálculo y seleccione en secuencia el menú, así como se muestra a continuación:



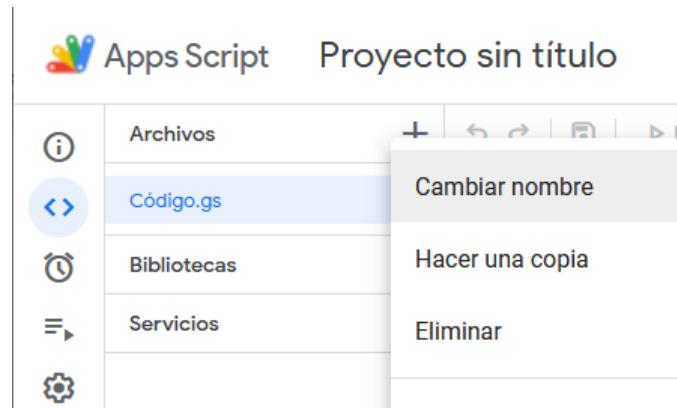
... y se abrirá una nueva pestaña, así como se muestra a continuación:



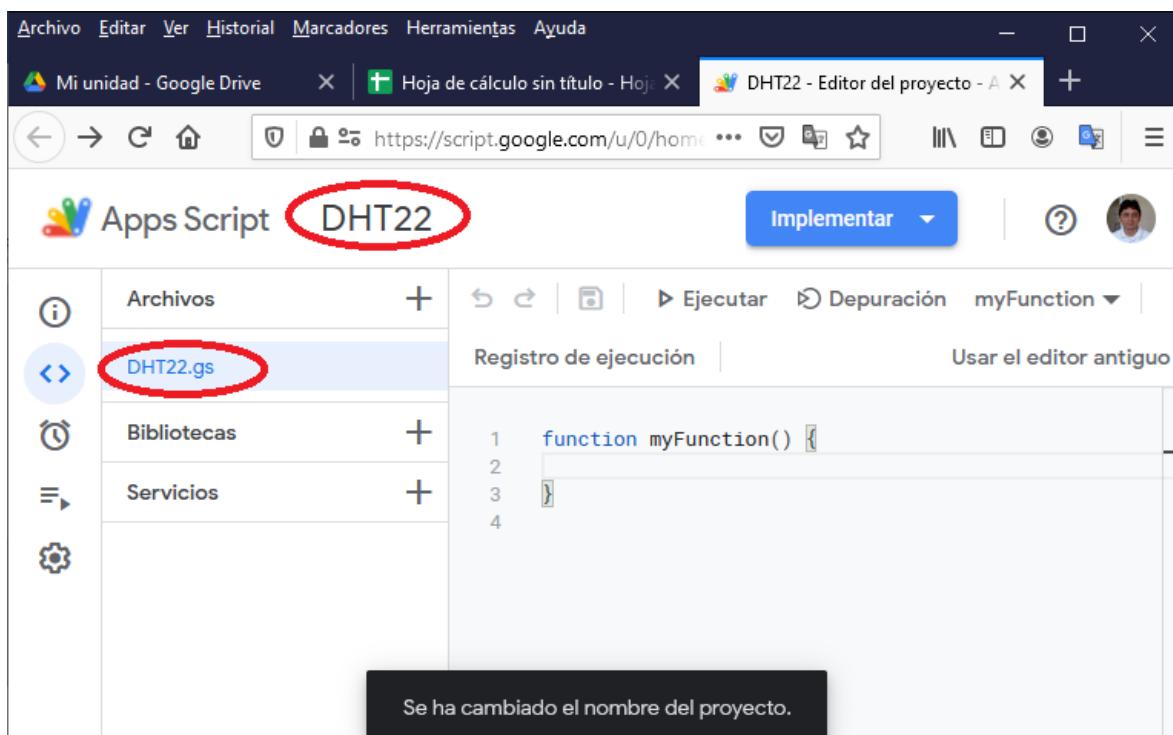
Ubíquese en el campo donde se encuentra el nombre del código y se activará un botón con tres puntitos:



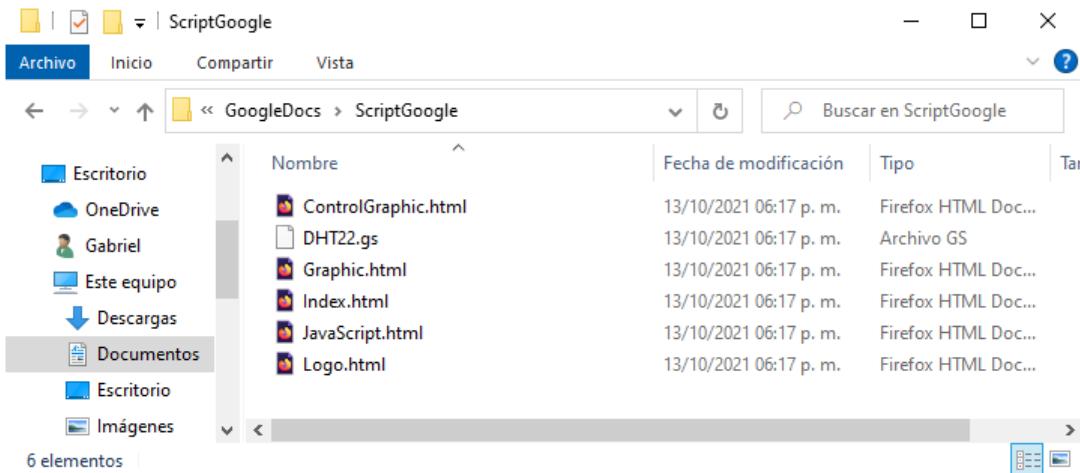
... presione el botón para activar el menú emergente:



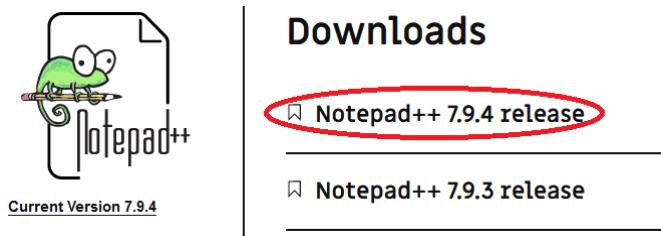
... y cambie el nombre del código a DHT22, haga lo mismo en la sección del nombre de proyecto y cambie el nombre a DHT22, así como se muestra:



Ahora, regresemos a la carpeta recién descargada y copiada a la carpeta de Documentos, entre a la carpeta GoogleDocs -> ScriptGoogle, así como se muestra a continuación:



Por facilidad, se requiere la instalación del programa llamado Notepad++ y lo puede obtener en <https://notepad-plus-plus.org/downloads/> :

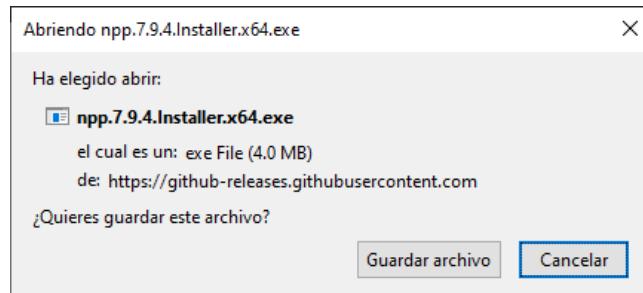


Descargue la versión correspondiente a la versión de su sistema operativo, si desconoce la versión, prueba instalando primero la versión de 64 bits, si falla la instalación, pruebe con la versión de 32 bits:

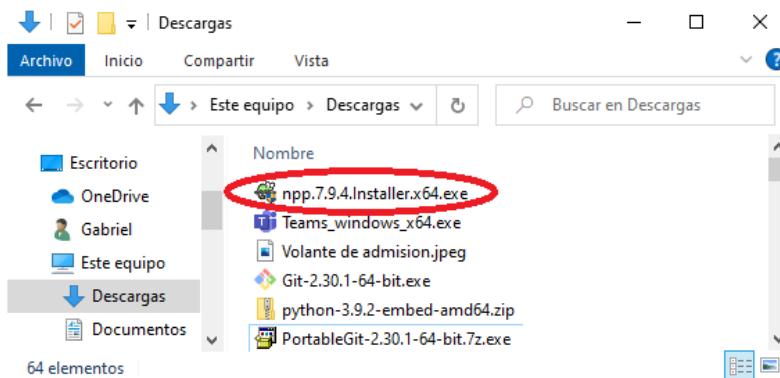
#### Download 64-bit x64

- [Installer | GPG Signature](#)
- [Portable \(zip\) | GPG Signature](#)
- [Portable \(7z\) | GPG Signature](#)
- [Mini-portable \(7z\) | GPG Signature](#)

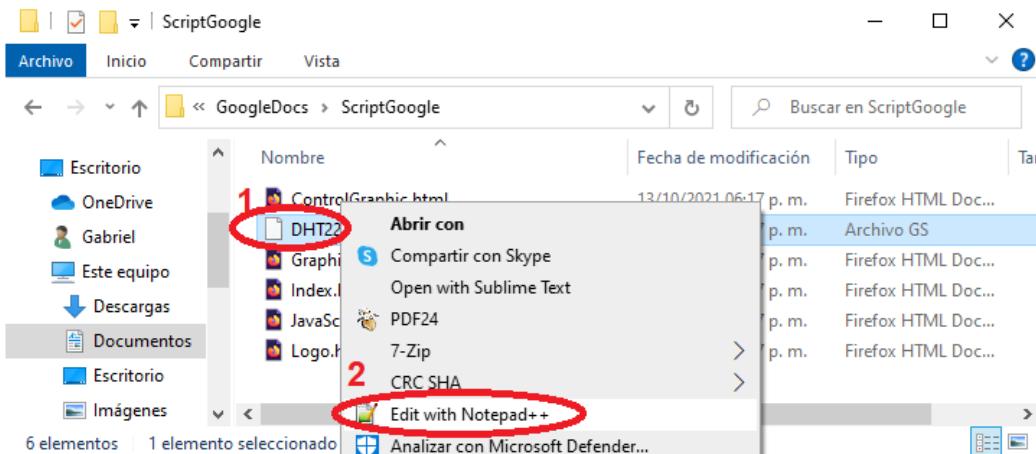
Descargue el archivo:



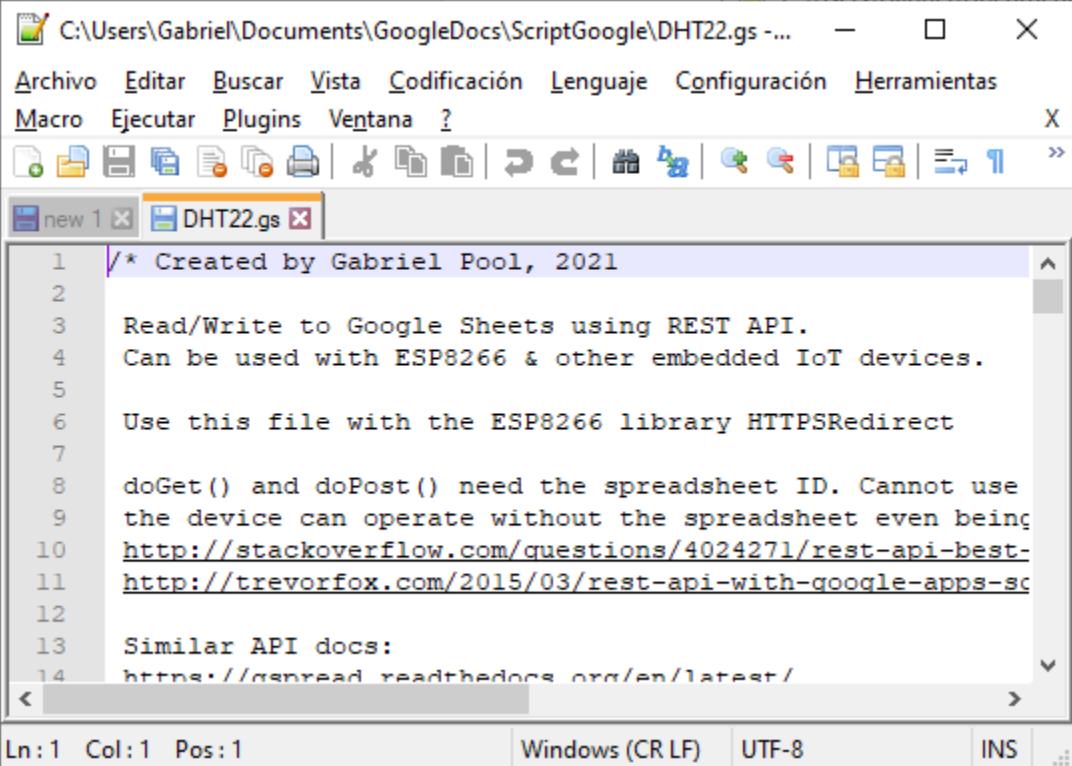
... e instale en su PC con las opciones que trae predeterminadas (sólo presione el botón “siguiente” hasta que el programa se instale):



Una vez instalado el Notepad++, regresamos a la ventana que se muestra donde se ubica el Script de Google, ubique el archivo DHT22 y abra con el programa recién instalado, con el botón derecho del mouse, dé un click sobre el archivo “DHT22.gs” y seleccione la opción que se muestra a continuación:



... e inmediatamente se abrirá una ventana como ésta:

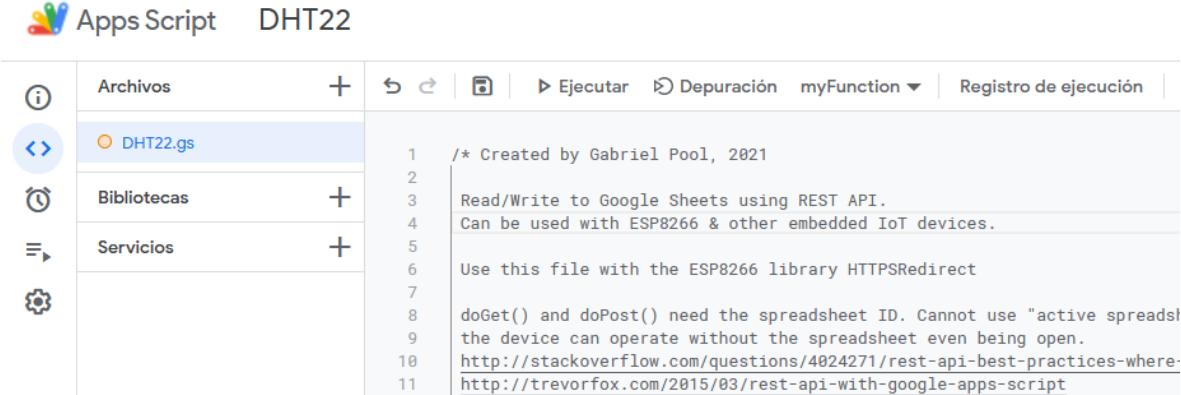


```

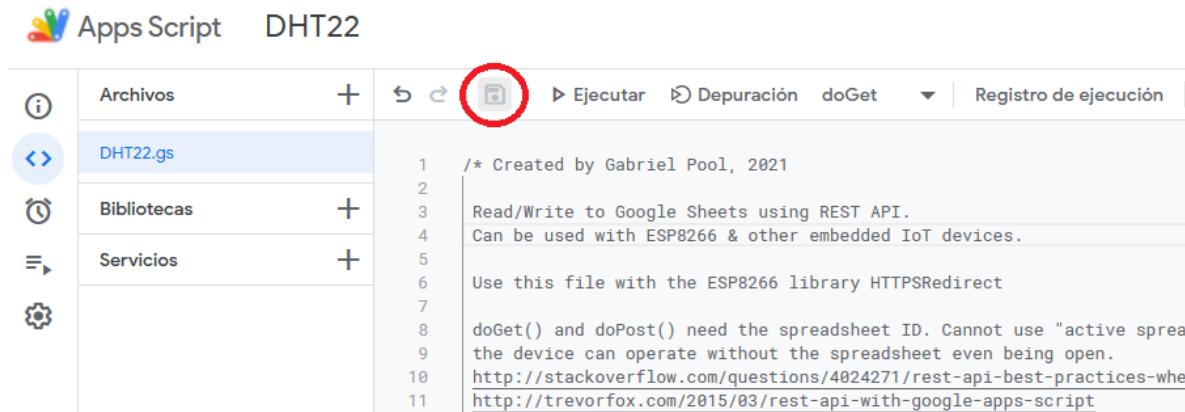
 1 /* Created by Gabriel Pool, 2021
 2
 3 Read/Write to Google Sheets using REST API.
 4 Can be used with ESP8266 & other embedded IoT devices.
 5
 6 Use this file with the ESP8266 library HTTPSRedirect
 7
 8 doGet() and doPost() need the spreadsheet ID. Cannot use
 9 the device can operate without the spreadsheet even being
10 http://stackoverflow.com/questions/4024271/rest-api-best-
11 http://trevorfox.com/2015/03/rest-api-with-google-apps-script
12
13 Similar API docs:
14 https://developers.google.com/apps-script/guides/rest-api

```

... presione las teclas Ctrl + a (Seleccionar todo el texto) y posteriormente Ctrl + c para copiar el texto, ubíquese en el navegador web, borre el contenido DHT22.gs en el navegador web y pegue el contenido copiado, así como se muestra:



Y presione el botón en forma de diskette para guardar los cambios:



Apps Script DHT22

Archivos +

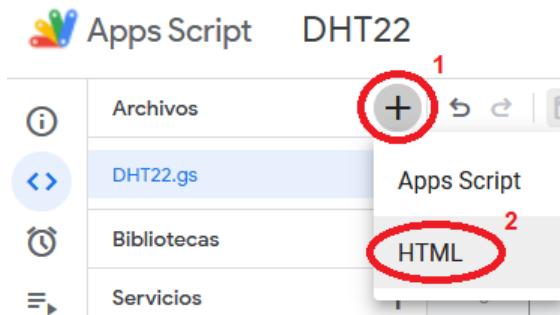
DHT22.gs

```

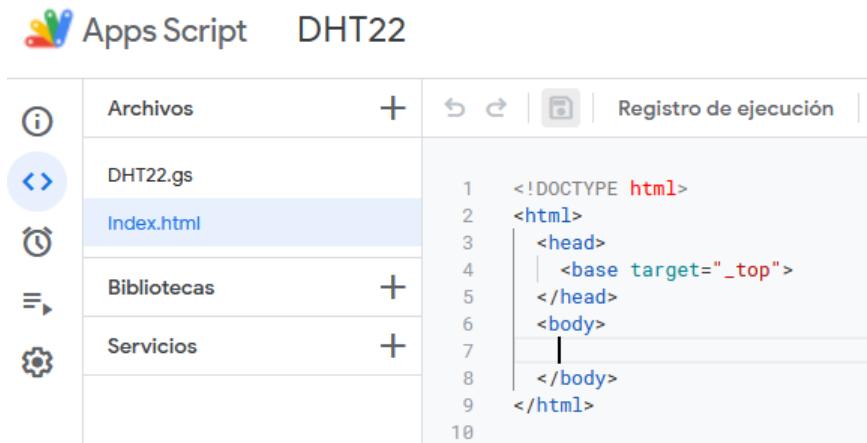
1 /* Created by Gabriel Pool, 2021
2
3 Read/Write to Google Sheets using REST API.
4 Can be used with ESP8266 & other embedded IoT devices.
5
6 Use this file with the ESP8266 library HTTPSRedirect
7
8 doGet() and doPost() need the spreadsheet ID. Cannot use "active spre
9 the device can operate without the spreadsheet even being open.
10 http://stackoverflow.com/questions/4024271/rest-api-best-practices-whe
11 http://trevorfox.com/2015/03/rest-api-with-google-apps-script

```

... seguidamente, presione el botón “+” para agregar el siguiente archivo:



... ponga el nombre “Index” (sin .html) con la primera letra mayúscula y presione la tecla “Enter” para guardar el nombre (el .html se agrega automáticamente):



Apps Script DHT22

Archivos +

DHT22.gs

Index.html

Bibliotecas +

Servicios +

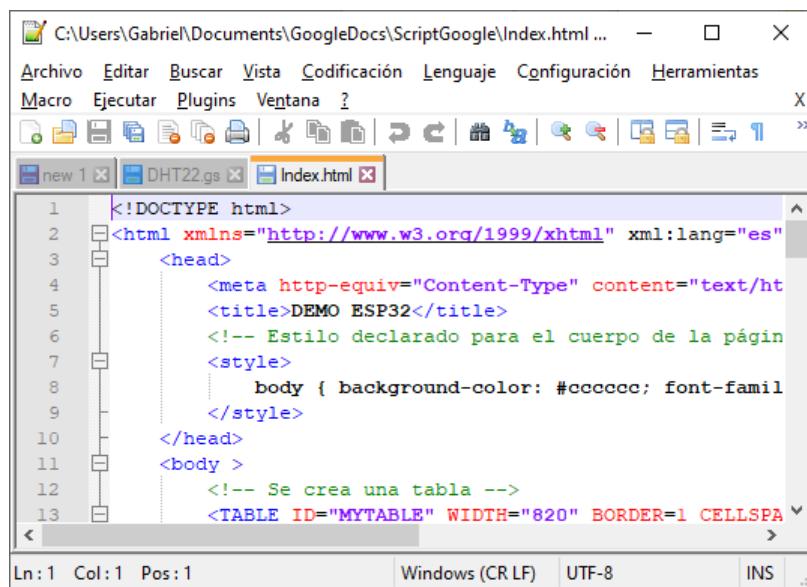
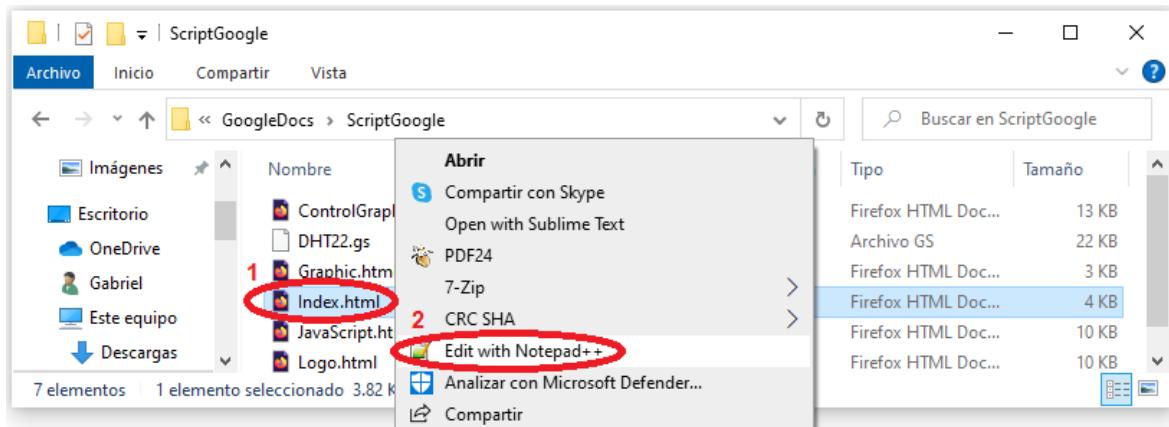
```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <base target="_top">
5   </head>
6   <body>
7     </body>
8   </html>
9
10

```

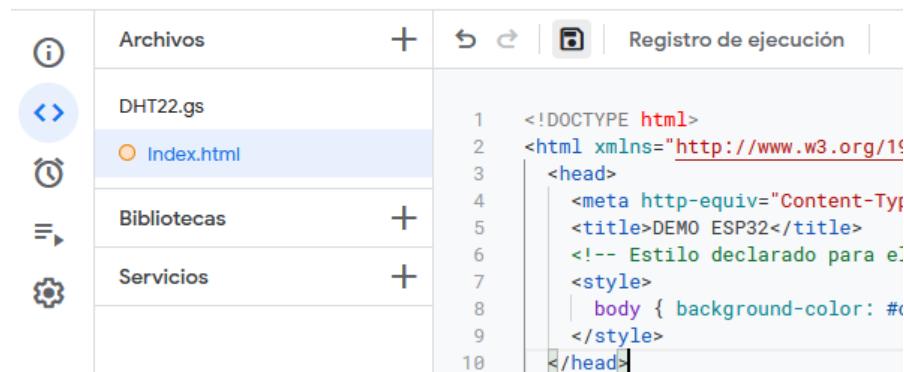
... al igual que en el caso anterior, borre el contenido de Index.html en su página

web, abra el archivo Index.html con el Notepad++ ubicado en su carpeta recién descargada, así como se muestra:



... presione las teclas Ctrl + a (Seleccionar todo el texto) y posteriormente Ctrl + c para copiar el texto, ubíquese en el archivo Index.html en el navegador web (NO olvide borrar el contenido previo de Index.html en el navegador web) y pegue el contenido copiado, así como se muestra:

### Apps Script DHT22



The screenshot shows the Google Apps Script interface for a project named "DHT22". On the left sidebar, there are icons for Archivos, DHT22.gs, Index.html, Bibliotecas, and Servicios. The "Index.html" file is selected. The main area displays the following HTML code:

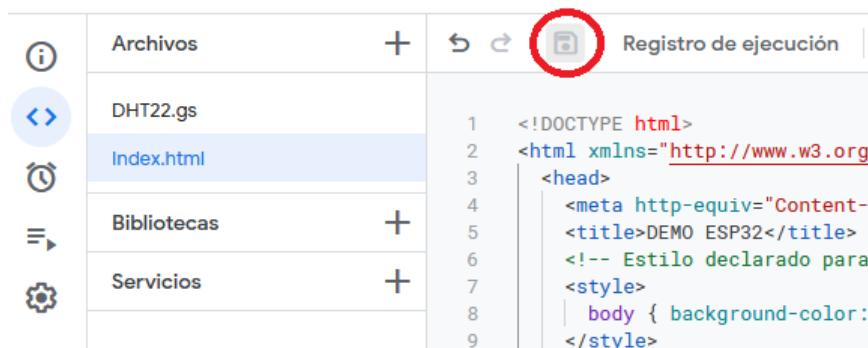
```

1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml">
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
5     <title>DEMO ESP32</title>
6     <!-- Estilo declarado para el fondo de la página -->
7     <style>
8       body { background-color: #00FFFF; }
9     </style>
10   </head>

```

Y presione el botón en forma de diskette para guardar los cambios:

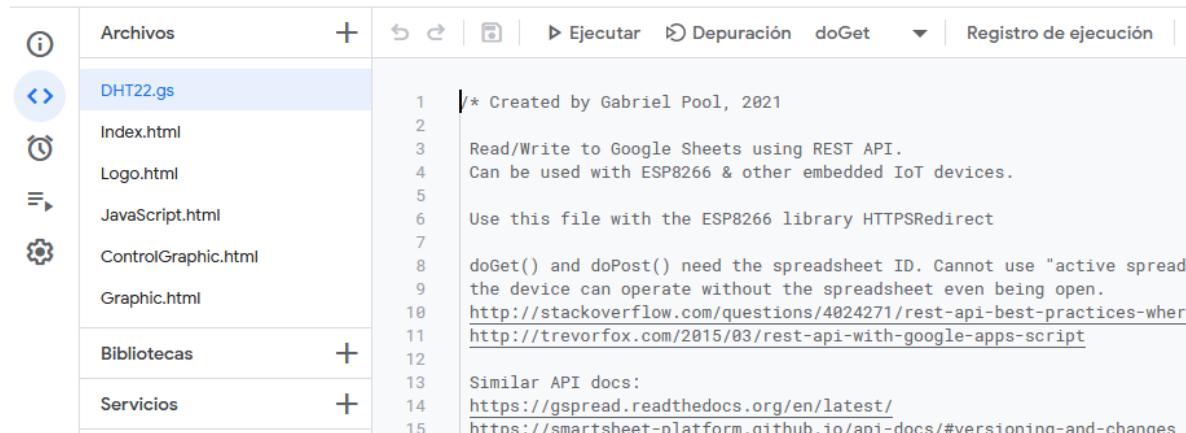
### Apps Script DHT22



The screenshot shows the Google Apps Script interface for the same project. The "Index.html" file is selected. The main area displays the same HTML code as before. The save icon (a floppy disk) in the toolbar is circled in red, indicating it should be clicked to save changes.

Repita estos pasos hasta subir todos los archivos el proyecto. Debe quedar, así como se muestra:

### Apps Script DHT22

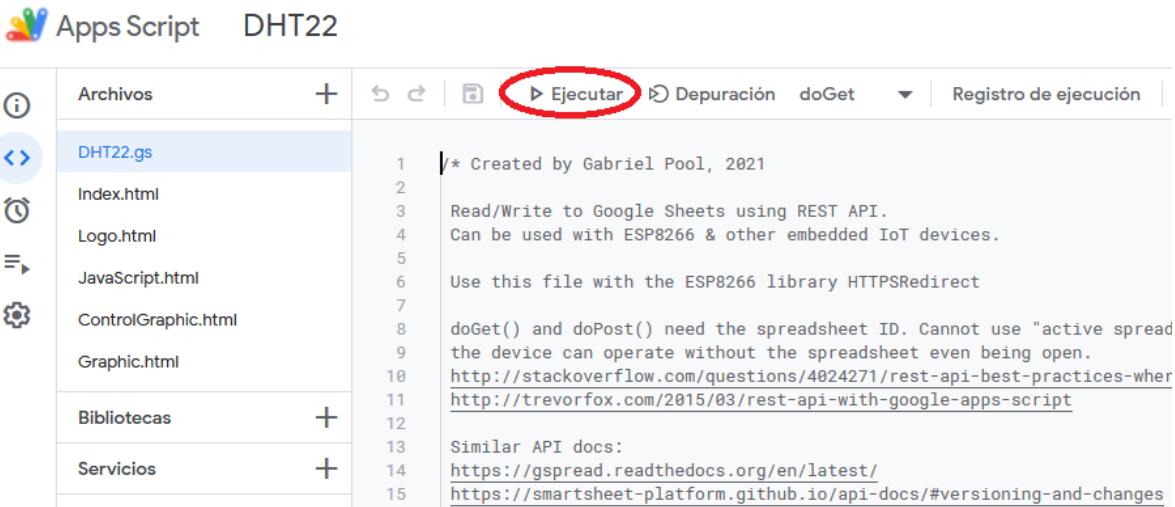


The screenshot shows the Google Apps Script interface with multiple files listed in the sidebar: DHT22.gs, Index.html, Logo.html, JavaScript.html, ControlGraphic.html, Graphic.html, Bibliotecas, and Servicios. The "DHT22.gs" file is selected. The main area shows the content of "DHT22.gs", which includes comments about using the REST API with ESP8266 devices and links to Stack Overflow and Trevor Fox's websites for more information. The save icon in the toolbar is circled in red.

En todos los casos, cuando se crea un nuevo archivo html en la Apps Script de Google ya incluye un contenido previo. Ese contenido previo se requiere borrarlo antes de insertar el código recién copiado del Notepad++. **NO importa el orden en que se crean los archivos, lo importante es que todos los nombres de los archivos se escriban tal como se muestran (respetando mayúsculas y minúsculas).** La extensión de archivo (.html) es agregado automáticamente al momento de presionar la tecla “ENTER” para guardar el nombre. Con los archivos ya completos, se requieren dos pasos para gestionar los permisos de ejecución:

**1er Paso:**

Seleccione el archivo DHT22.gs y presione el botón de ejecutar, así como se muestra:



Apps Script    DHT22

Archivos

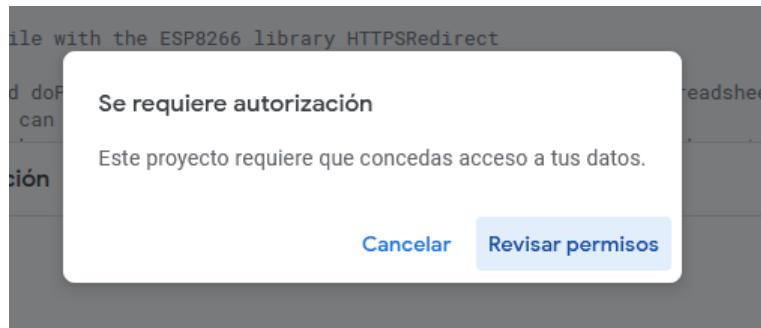
- DHT22.gs
- Index.html
- Logo.html
- JavaScript.html
- ControlGraphic.html
- Graphic.html

Bibliotecas

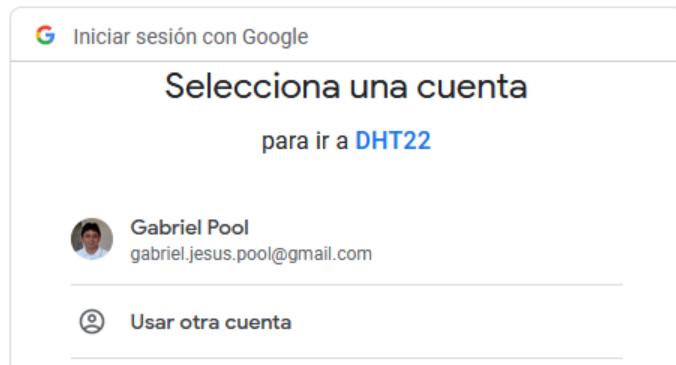
Servicios

```
/* Created by Gabriel Pool, 2021
Read/Write to Google Sheets using REST API.
Can be used with ESP8266 & other embedded IoT devices.
Use this file with the ESP8266 library HTTPSRedirect
doGet() and doPost() need the spreadsheet ID. Cannot use "active spread
the device can operate without the spreadsheet even being open.
http://stackoverflow.com/questions/4024271/rest-api-best-practices-wher
http://trevorfox.com/2015/03/rest-api-with-google-apps-script
Similar API docs:
https://gspread.readthedocs.org/en/latest/
https://smartsheet-platform.github.io/api-docs/#versioning-and-changes
```

... e inmediatamente se oscurecerá la pantalla y saldrá un mensaje como éste:



... presione el botón “Revisar Permisos”:



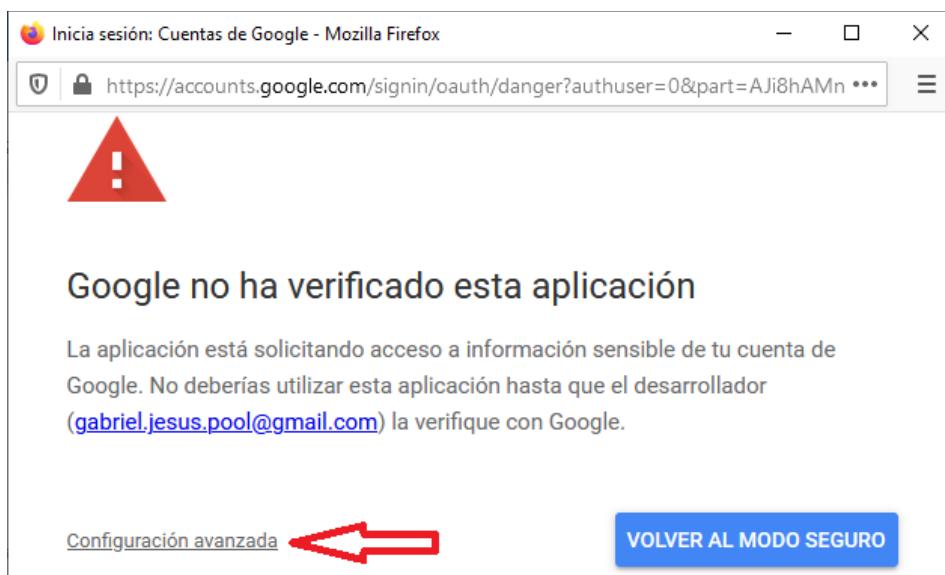
Español (España) ▾

Ayuda

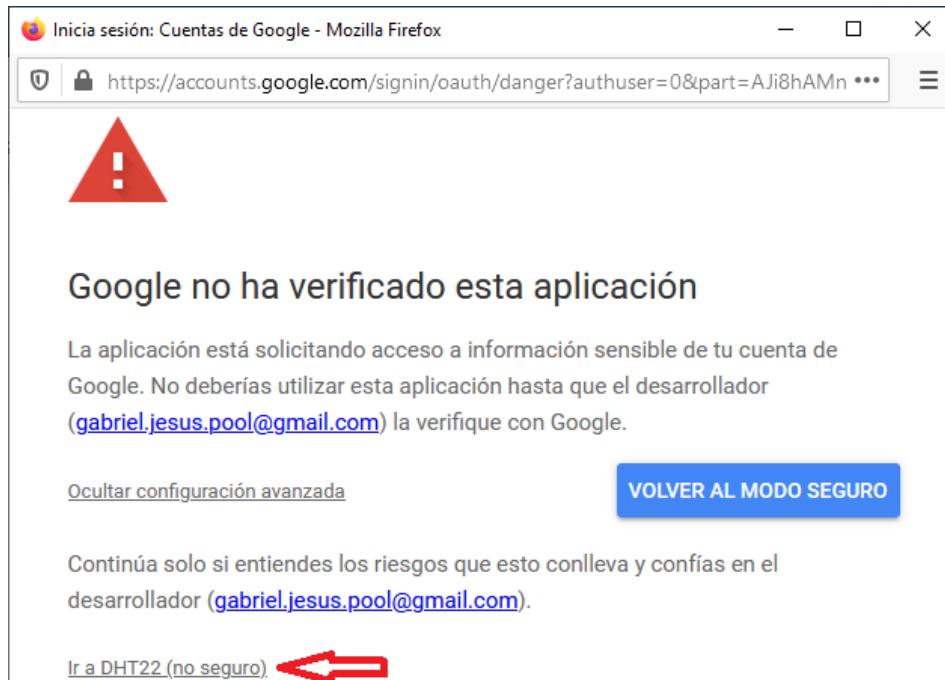
Privacidad

Términos

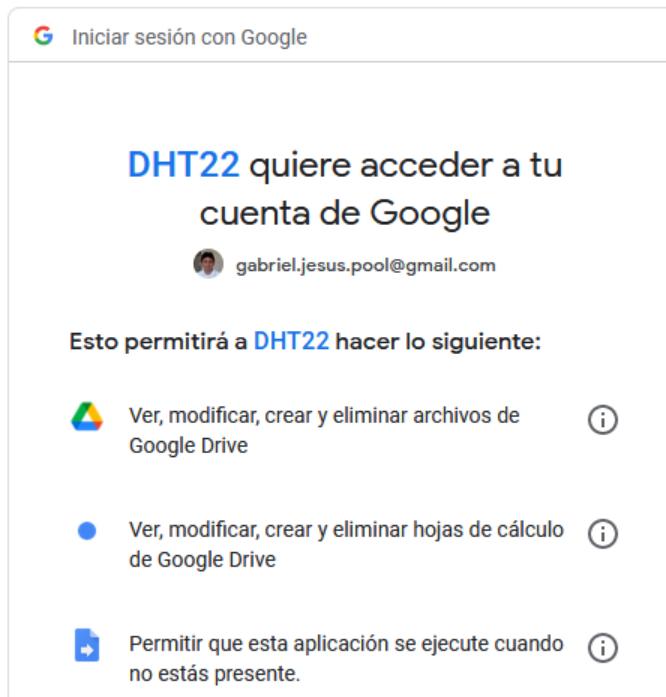
... presione en el área donde se encuentra su logo y correo e inmediatamente saldrá una ventana como ésta:



... presione el hipertexto que se encuentra marcado con una flecha en la figura anterior y saldrá otra ventana como ésta:

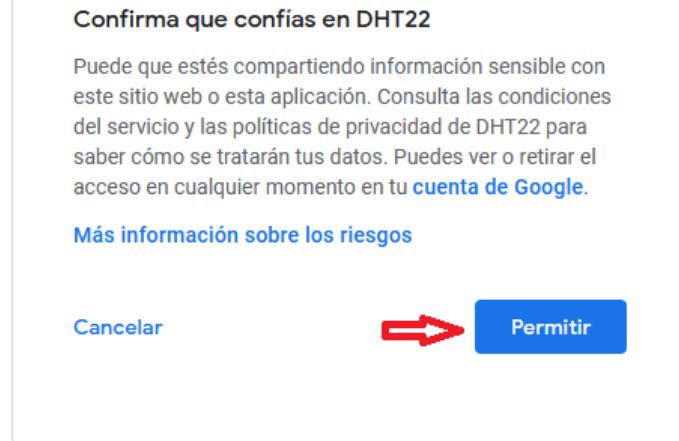


... nuevamente, presione el hipertexto marcado en la figura anterior y saldrá otra ventana como ésta:



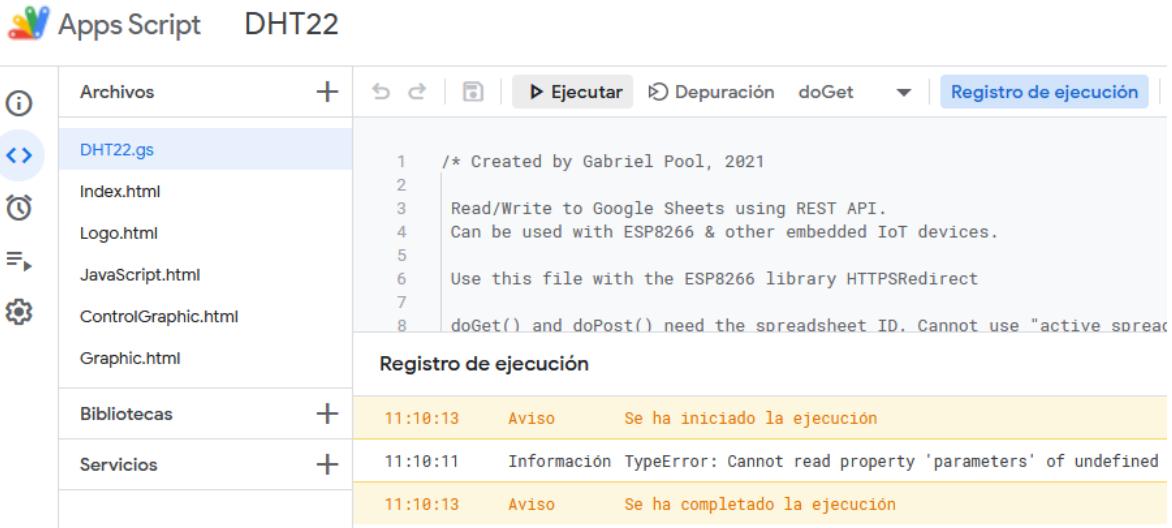
... deslice hasta encontrar el botón “Permitir” y presione el botón:

204/224



Español (España) ▾ Ayuda Privacidad Términos

La ventana anterior se cerrará, pero si se toma demasiado tiempo en seguir estos pasos, en el código de Google mostrará un mensaje indicando que requiere permisos de ejecución, basta con presionar el botón de "ejecutar" y el código se ejecutará sin abrir la ventana para la gestión de permisos. Al ejecutar el código se mostrará un mensaje:

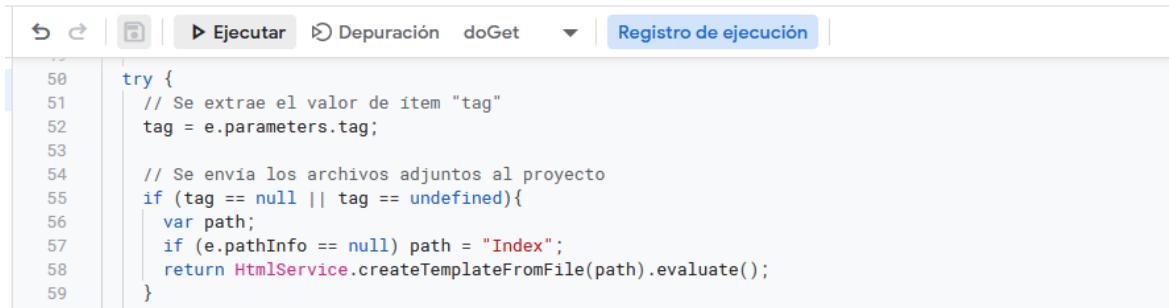


Timestamp	Severity	Message
11:10:13	Aviso	Se ha iniciado la ejecución
11:10:11	Información	TypeError: Cannot read property 'parameters' of undefined
11:10:13	Aviso	Se ha completado la ejecución

Eso indica que el código se ha ejecutado exitosamente.

## 2º Paso:

Ahora se ubica en la parte superior izquierda de su navegador web y encontrará un botón azul llamado "Implementar", presione dicho botón, así como se muestra:

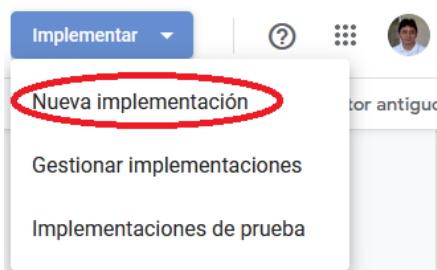



```

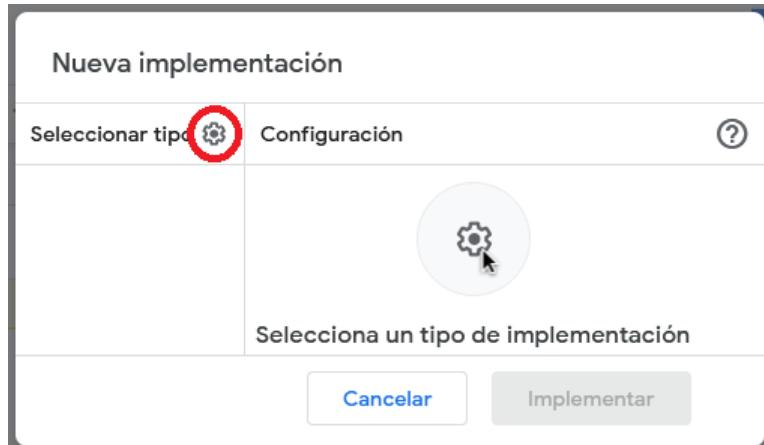
50 try {
51     // Se extrae el valor de item "tag"
52     tag = e.parameters.tag;
53
54     // Se envia los archivos adjuntos al proyecto
55     if (tag == null || tag == undefined){
56         var path;
57         if (e.pathInfo == null) path = "Index";
58         return HtmlService.createTemplateFromFile(path).evaluate();
59     }

```

... e inmediatamente se abrirá una ventana como esta:

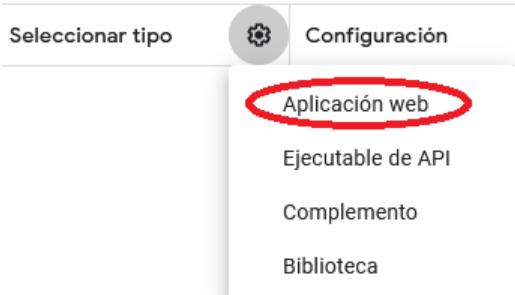


Presione la opción “Nueva Implementación”, y saldrá una ventana, así como se muestra:



... presione el botón con forma de engrane y saldrá un menú emergente:

Nueva implementación



... seleccione la opción “Aplicación Web” y se mostrará lo siguiente:

Nueva implementación

Seleccionar tipo	Configuración
Aplicación web	<p>Descripción</p> <p>Nueva descripción</p> <p>Aplicación web</p> <p>Ejecutar como</p> <p>Yo (gabriel.jesus.pool@gmail.com)</p> <p>La ejecución de la aplicación web se autorizará con los datos de tu cuenta.</p> <p>Quién tiene acceso</p> <p>Solo yo</p> <p>También se puede usar como biblioteca. <a href="#">Más información</a></p>

**Cancelar** **Implementar**

... en el campo nueva descripción ponga la versión de su proyecto, en el campo aplicación web, no haga cambios, y en el último campo llamado “Quien tiene acceso”, seleccione “Cualquier usuario”, los campos deben quedar, así como se muestra:

Nueva implementación

Seleccionar tipo	Configuración	?
Aplicación web	<p>Descripción</p> <p>Nueva descripción</p> <p>version 1</p> <p>Aplicación web</p> <p>Ejecutar como</p> <p>Yo (gabriel.jesus.pool@gmail.com)</p> <p>Quién tiene acceso</p> <p>Cualquier usuario</p> <p>También se puede usar como biblioteca.<a href="#">Más información</a></p>	

[Cancelar](#) [Implementar](#)

... seguidamente, presione el botón azul llamado “Implementar” y le saldrá una ventana como esta:

### Nueva implementación

La implementación se ha actualizado correctamente.

Versión 1 del 17 mar, 22:58

ID de implementación

AKfycbz...wMLBXnLkBmra9g4zgeY7LmgQ0jBmAG3uEUQHdb7dY8uNHZ0BXr99fQzl2TWbb

 **Copiar**

Aplicación web

URL

<https://script.google.com/macros/s/AKfycbz...wMLBXnLkBmra9g4zgeY7LmgQ0jBmAG3uEUQHdb7dY8uNHZ0BXr...>

 **Copiar**

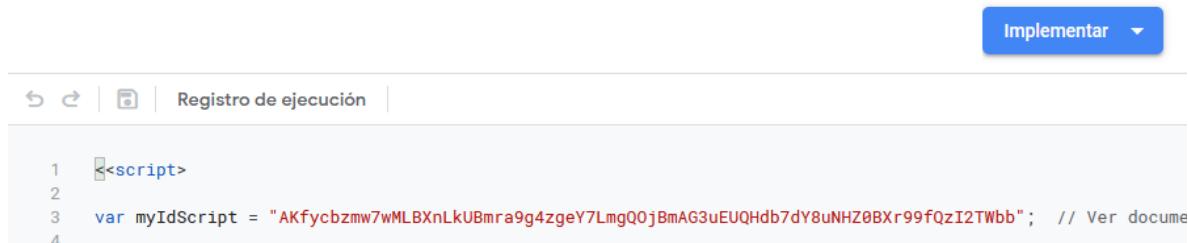
**Listo**

Presione el botón marcado en la figura anterior para copiar el Id del proyecto, este es necesario obtenerlo para actualizar el archivo JavaScript y posteriormente implementarlo en el módulo NodeMCU. Presione el botón listo para cerrar la ventana anterior, y ahora se requiere actualizar el archivo JavaScript como sigue:

Apps Script    DHT22

(i)	Archivos	+ ↻ ⌂ ☑ Registro de ejecución
	DHT22.gs	1    <<script>
	Index.html	2
	Logo.html	3    var myIdScript = "myIdScript"; // Ver documentación del proyecto
	JavaScript.html	4
	ControlGraphic.html	5
	Graphic.html	6    // Esta sección se encarga de la comunicación asincrónica entre cli
		7    // Envía, espera y recibe las peticiones hechas al servidor, por e
		8
		9
		10   // Determines when a request is considered "timed out"

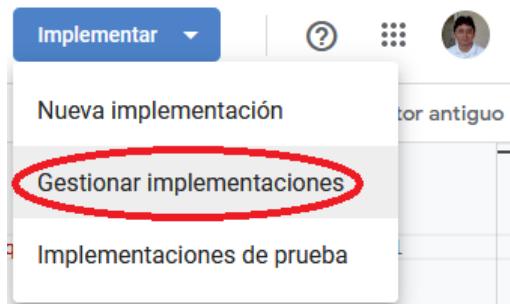
... ubique la línea 3 y sustituya el texto “myIdScript” por el Id del proyecto que se acaba de copiar. No olvide que el Id del proyecto también va entre comillas:



The screenshot shows a software interface with a toolbar at the top featuring icons for back, forward, and save, followed by a blue button labeled "Implementar". Below the toolbar is a header bar with the text "Registro de ejecución". The main area contains a code editor with the following content:

```
1 <script>
2
3 var myIdScript = "AKfyccbzmw7wMLBXnLkUBmra9g4zgeY7LmgQ0jBmAAG3uEUQHdb7dY8uNHZ0BXr99fQzI2TWbb"; // Ver docume
4
```

Asegúrese de guardar los cambios presionando el botón del diskette y vuelva a presionar el botón azul de “Implementar” para crear una segunda versión del proyecto, ya que, la primera versión NO va a funcionar ya que, es imprescindible el Id del proyecto:



... y se abrirá una ventana como esta:

Gestionar implementaciones

Activa	Configuración
version 1	<small>versión</small> Versión 1 del 12 nov, 13:29
Archivadas	<small>Descripción</small> version 1
No hay implementaciones archivadas	
	<small>ID de implementación</small> AKfyccb... <input type="button" value="Copiar"/>
	<small>Aplicación web</small> <small>URL</small> <a href="https://script.google.com/macros/s/AKfyccb...">https://script.google.com/macros/s/AKfyccb...</a> <input type="button" value="Copiar"/>

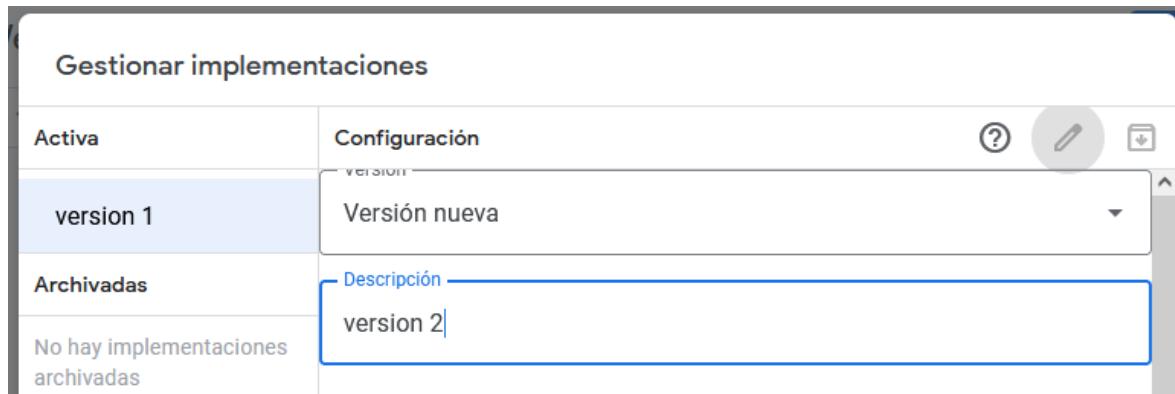
Presione el botón marcado en la siguiente imagen, así como se muestra:

Gestionar implementaciones

Activa	Configuración
version 1	<small>versión</small> Versión 1 del 12 nov, 13:29

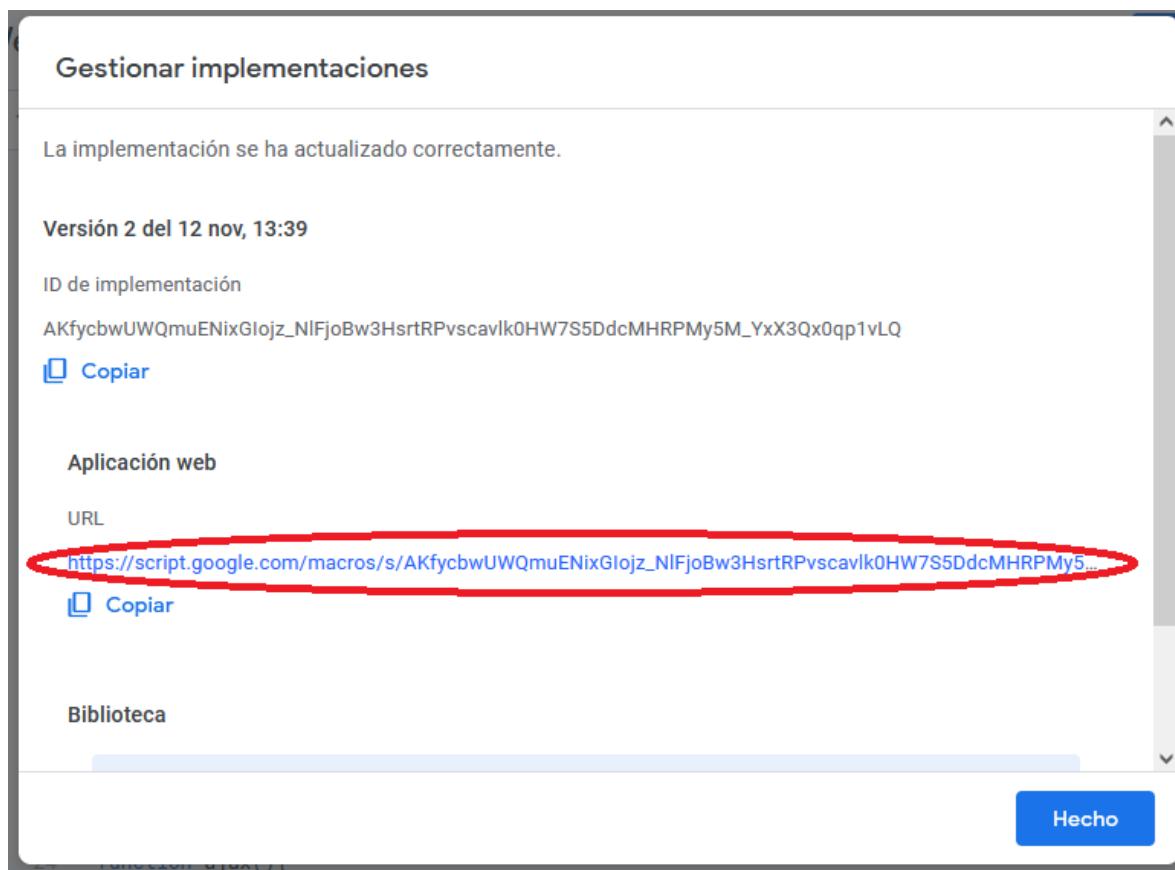
↓

... e inmediatamente se desbloquean los campos configuración y descripción, seleccione la opción y llene el segundo campo, así como se muestra:



Activa	Configuración
version 1	<div style="border: 1px solid #ccc; padding: 5px;"> <span>Versión</span>   <input type="button" value="Versión nueva"/>   <span>Descripción</span>   <input type="text" value="version 2"/> </div>
Archivadas	No hay implementaciones archivadas

... presione el botón azul llamado “Implementar” que se encuentra en la parte inferior derecha y la ventana mostrará lo siguiente:



Gestionar implementaciones

La implementación se ha actualizado correctamente.

Versión 2 del 12 nov, 13:39

ID de implementación

AKfycbwUWQmuENixGlojz\_NIFjoBw3HsrtRPvscavlk0HW7S5DdcMHRPMy5M\_YxX3Qx0qp1vLQ

Aplicación web

URL

[https://script.google.com/macros/s/AKfycbwUWQmuENixGlojz\\_NIFjoBw3HsrtRPvscavlk0HW7S5DdcMHRPMy5...](https://script.google.com/macros/s/AKfycbwUWQmuENixGlojz_NIFjoBw3HsrtRPvscavlk0HW7S5DdcMHRPMy5...)

Biblioteca

... presione el vínculo marcado en la figura anterior y si todo salió bien, en una nueva pestaña, observará lo siguiente:



Centro de Investigación Científica de Yucatán A. C.  
Departamento de Instrumentación  
[www.cicy.mx](http://www.cicy.mx)

The screenshot shows a web browser window with multiple tabs open. The active tab is a Google Apps Script macro titled 'script.google.com/macros/s/AKfycbz...' which displays a weather station interface. The interface includes the CICY logo, the title 'Centro de Investigación Científica de Yucatán A.C. Departamento de Instrumentación 2021', fields for 'Temperatura:' and 'Humedad:', a timestamp 'Hora Local' showing '18/03/2021 00:17:54', a link to 'Descargar historial' with a dropdown menu 'Elija mes', and a chart area showing temperature and humidity data for the last 24 hours. The chart has two data series: 'Temperatura °C' and 'Humedad %', both showing values of 0 at the top and 100 at the bottom.

No se podrá observar dato alguno, puesto que el sensor no ha sido implementado. En el último punto de este capítulo titulado: “El Cliente para Google Apps Script implementado en NodeMCU-32S” se implementará el código del dispositivo IoT.

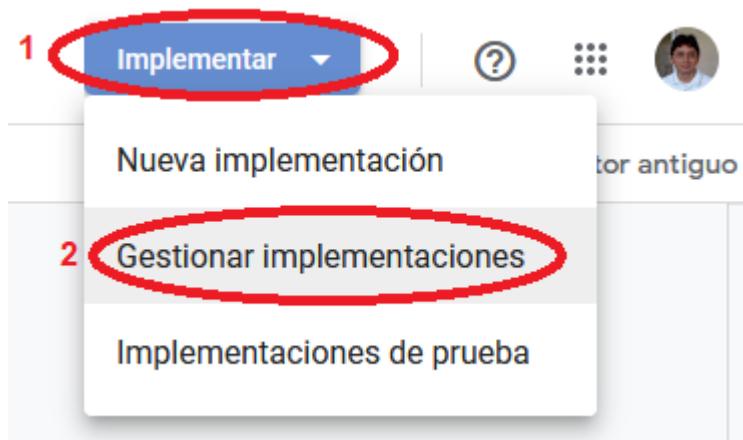
Si no puede observar todo lo mostrado anteriormente en la página web (incluyendo la hora del servidor) indica que ocurrió un problema en la implementación. No es necesario borrar el proyecto, trate de verificar los pasos anteriores y de encontrar el error, solamente siga los pasos a partir de donde se copió el Id del proyecto para hacer una nueva implementación.

**IMPORTANTE:** Por cada cambio realizado en el proyecto es necesario realizar una nueva implementación, ya que, de no hacerlo (sólo guarda los cambios) el último proyecto implementado es el que seguirá funcionando.

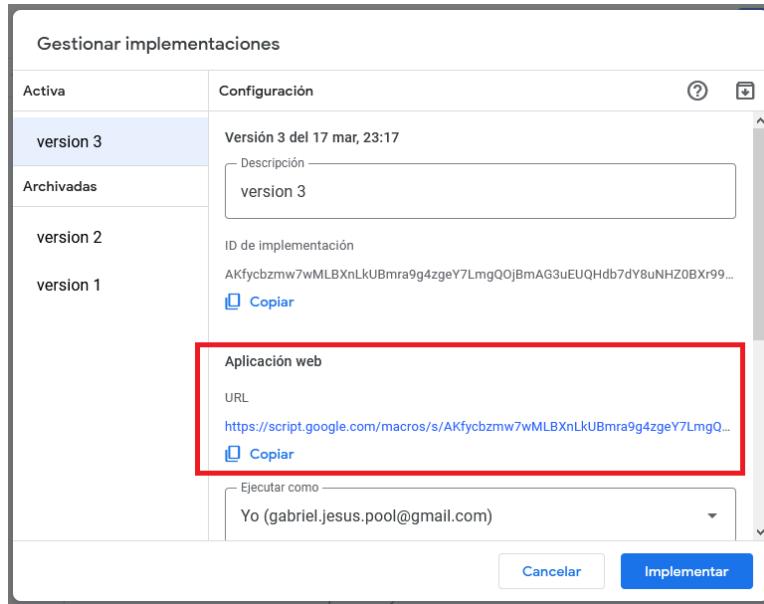
Nótese que se utilizó la misma página web implementada en el punto 3.2.2 con sólo un cambio: el control “select” para la descarga de históricos en las lecturas.

El propósito de la página web es evitar que el usuario interactúe con los archivos que guardan los históricos. Dichos archivos son visibles para el usuario y puede descargarlos directamente desde su cuenta en Google, pero hay un pequeño defecto: si realiza un cambio de manera accidental, podría perder la información. Adicionalmente, la página web contiene una gráfica donde se muestra los valores de las últimas 24 horas y podría darle una idea rápida al usuario sobre el comportamiento de dichas variables. Si Usted es un usuario experto, podría prescindir de todas las funciones de la página web y el código se volvería más sencillo, pero si planea venderlo a un cliente, sería mejor que todo lo gestione desde la página web.

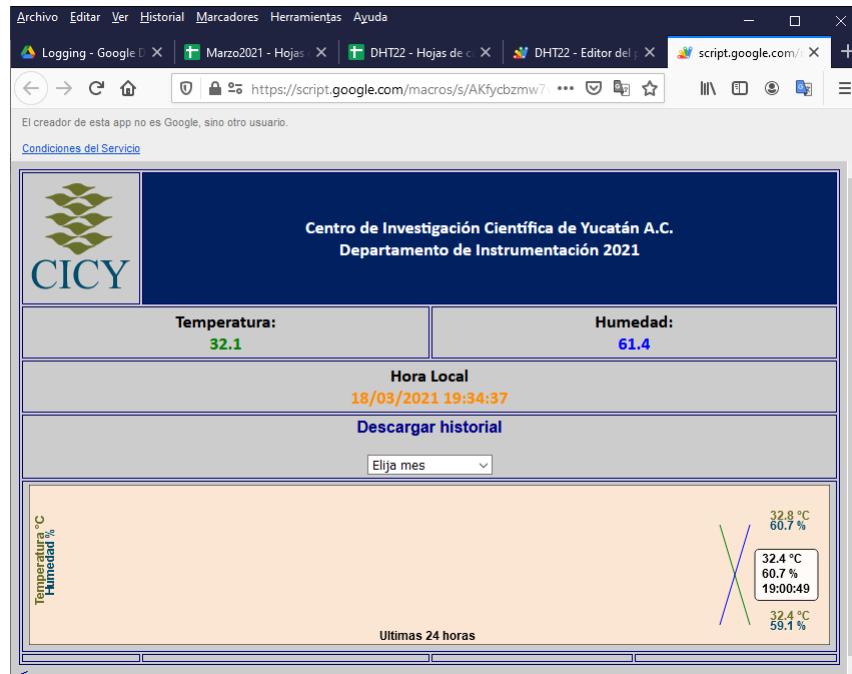
Para obtener el link donde se ubica la página web, se parte del proyecto ya implementado y ubicando el botón azul llamado “Implementar”, siga la siguiente secuencia:



Al presionar la opción anterior, inmediatamente se abrirá una ventana como ésta:



Si usted presiona el hipertexto “Copiar” el link quedará en el portapapeles y Ud podrá compartirlo mediante mensajería instantánea o correo electrónico. Si presiona el link que inicia con https inmediatamente se abrirá una ventana como ésta:

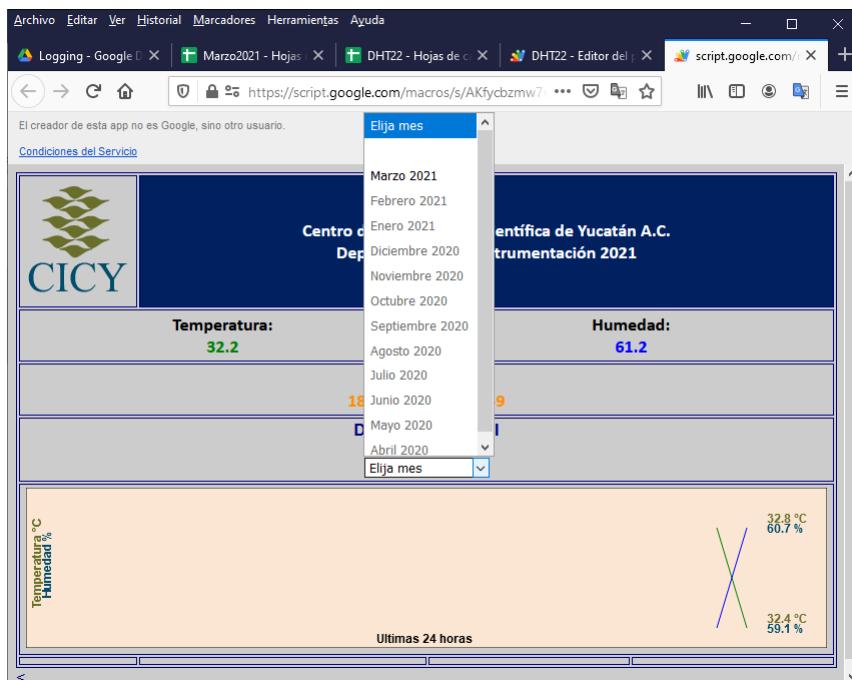


A diferencia de la vez anterior, ya se puede observar los valores en la gráfica y al término de 24 horas podrá observar la gráfica completa.

El único detalle sobre el link es que es largo y difícil de aprender, pero existen

215/224

servicios (acortadores de link) que podría resolver ese problema. Para poder descargar el histórico, simplemente presione el selector y elija el mes de descarga que desee. Si el mes se encuentra en gris, indica al usuario que ese archivo no se encuentra en el servidor (no hay registros de ese mes).



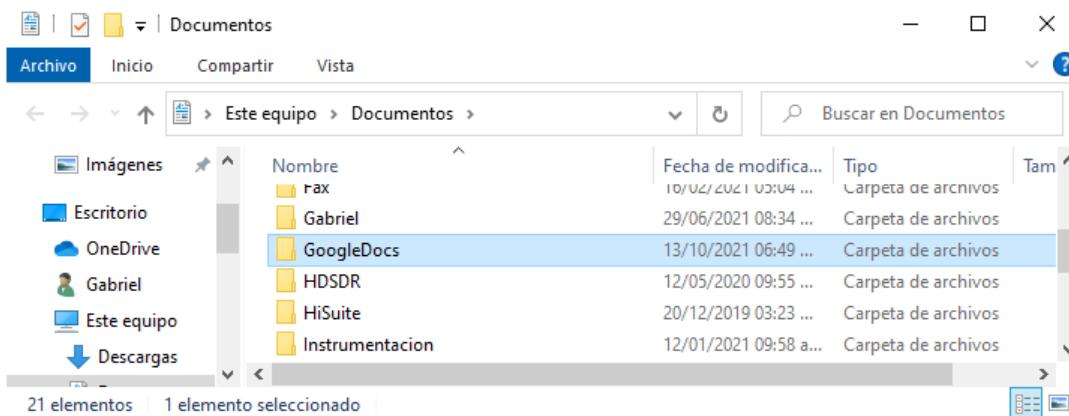
Otro detalle adicional que es muy notable es el encabezado que Google le agrega al proyecto con el fin de deslinde de responsabilidades. Hay que destacar que este proyecto utiliza intercambio de datos entre el servidor y el cliente, así que, Google sólo permite el envío de archivos que se agrega al proyecto, en este caso: index.html, JavaScript.html, ControlGraphic.html, Graphic.html y Logo.html en formato html, svg y JavaScript (todos embebidos en el archivo html). Para el response admite los formatos xml, JSON y texto plano. Google NO permite el intercambio de fotos, pdf, ni cualquier otro archivo almacenado en el Drive de Google (se activa la directiva CORS y las bloquea).

En la red existe una amplia ayuda para desarrollar en esta plataforma, adicionalmente que JavaScript es un lenguaje muy conocido.

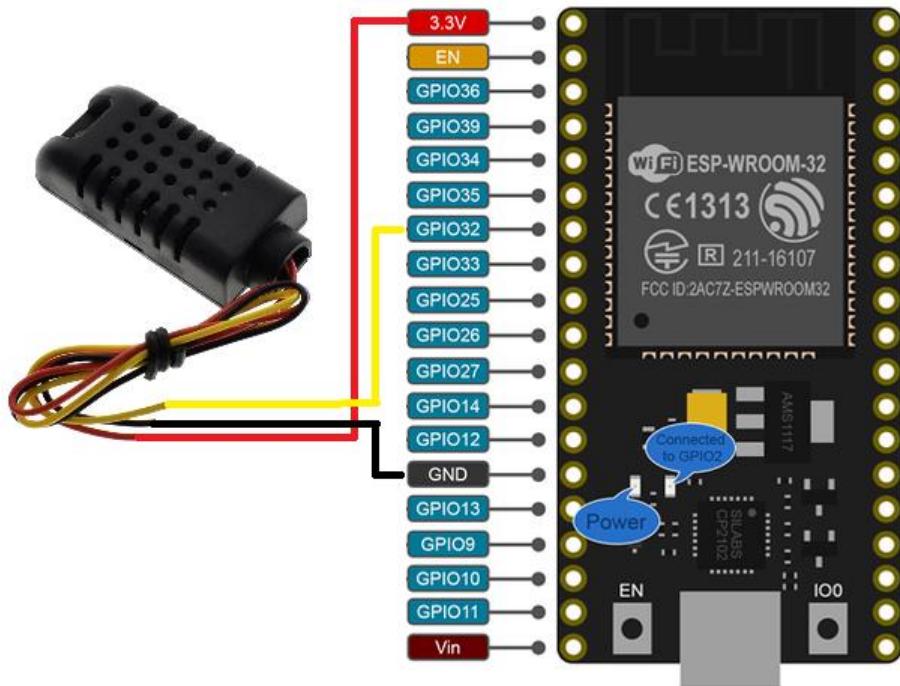
### 3.3.4 El Cliente para Google Apps Script implementado en NodeMCU-32S

**Objetivo específico:** Configurará e implementará el cliente en el NodeMCU-32S para enviar datos de Humedad y Temperatura al Google Sheet implementado en los puntos anteriores.

Ahora se configura el NodeMCU-32S como cliente para que pueda enviar los datos al servidor de Google. En el sitio <https://github.com/gpoolb/ESP32>, específicamente en la carpeta “GoogleDocs” se encuentra la implementación de este proyecto. En el punto 3.3.1 se hizo la descarga y la carpeta fue copiada dentro de Mis Documentos, así como se muestra:



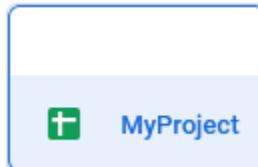
Se aclara que, en el punto 2.2.6 de este manual, se implementó un proyecto donde se utiliza el sensor DHT 22:



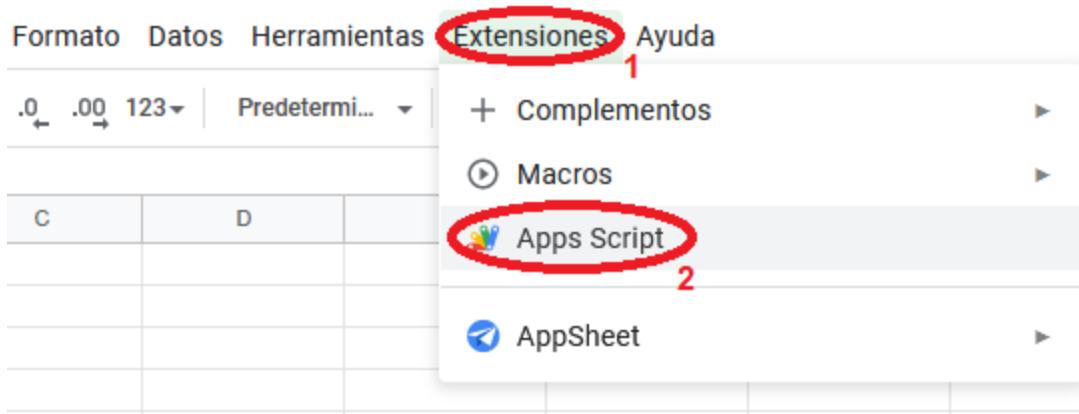
Para implementar este proyecto, se requiere que ya se hayan descargado las bibliotecas necesarias para que el DHT22 funcione, adicionalmente, se requiere conectar el DHT22 en el pin 32 de su NodeMCU-32S.

El último requisito para esta implementación es el Id del proyecto realizado en el punto 3.3.1 y lo puede obtener de esta manera:

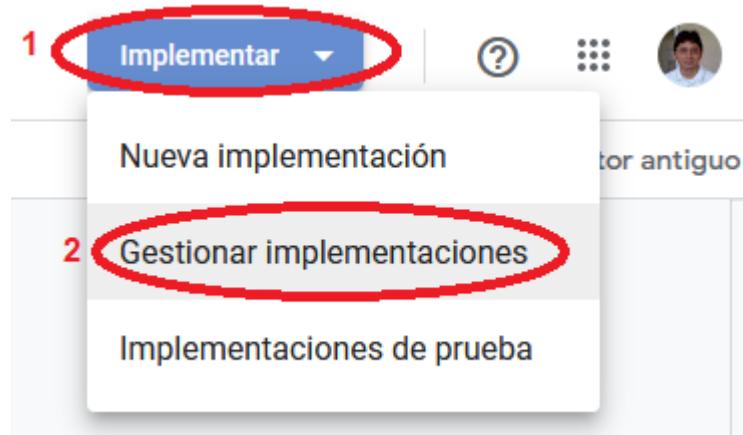
- Accesa al Google Drive y ubique la hoja de cálculo donde implementó su proyecto, debe ubicarlo por este ícono (con el nombre de su proyecto):



- Abra la hoja de cálculo y en el menú de la parte superior, seleccione en secuencia lo siguiente:



... y se abrirá el proyecto generado en el punto anterior, ubique en la parte superior derecha el botón azul llamado “Implementar” y seleccione la opción que se muestra:



... e inmediatamente se mostrará una ventana como ésta:

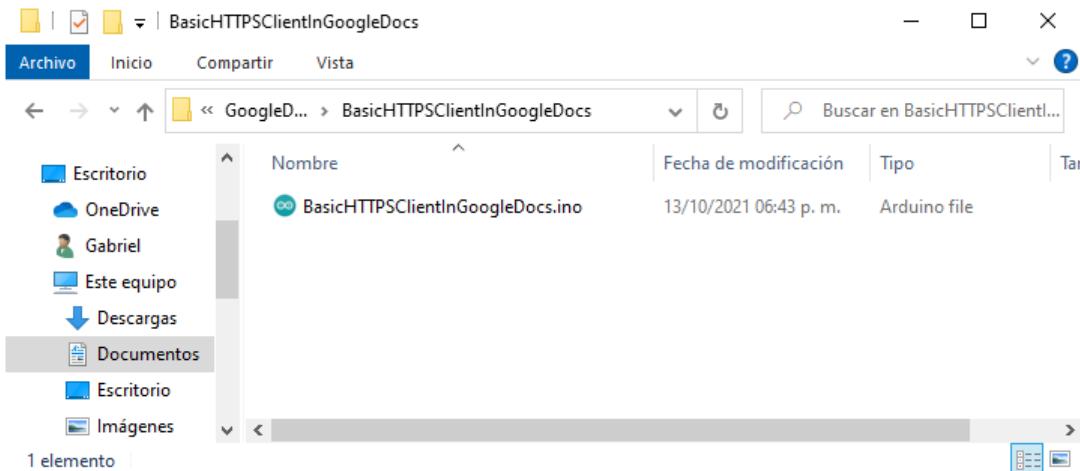
### Gestionar implementaciones

Activa	Configuración
version 3	<p>Versión 3 del 17 mar, 23:17</p> <p>Descripción version 3</p>
Archivadas	
version 2	<p>ID de implementación</p> <p>AKfyccbzmw7wMLBXnLkUBmra9g4zgeY7LmgQOjBmAG3uEUQHdb7dY8uNHZ0BXr99...</p>
version 1	<p>1 <span style="color: red;">Copiar</span></p> <p>Aplicación web</p> <p>URL <a href="https://script.google.com/macros/s/AKfyccbzmw7wMLBXnLkUBmra9g4zgeY7LmgQ...">https://script.google.com/macros/s/AKfyccbzmw7wMLBXnLkUBmra9g4zgeY7LmgQ...</a></p> <p><span style="color: blue;">Copiar</span></p> <p>Ejecutar como Yo (gabriel.jesus.pool@gmail.com)</p>

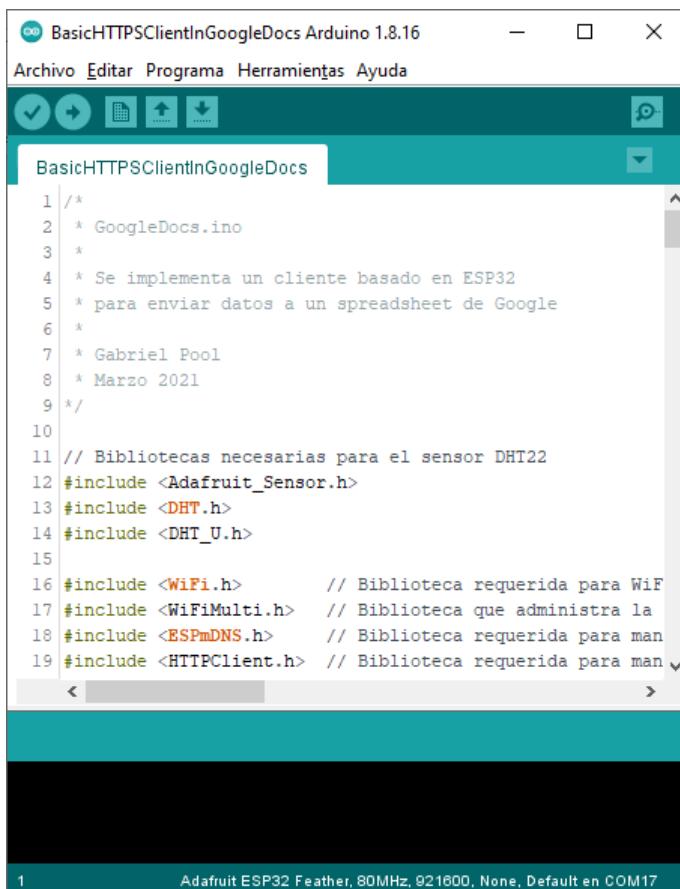
2 Cancelar Implementar

Presione el hipertexto “Copiar” y el Id del proyecto se quedará en el portapapel, cópielo temporalmente en el bloque de notas y posteriormente presione el botón “Cancelar”. Por el momento, no hay nada más que hacer en el proyecto de Google, se sugiere cerrar el navegador web.

Abra el programa del Arduino ubicado en Documentos -> GoogleDocs -> BasicHTTPSSBasicGoogleDocs:



... déle doble click al archivo y se abrirá la siguiente ventana:

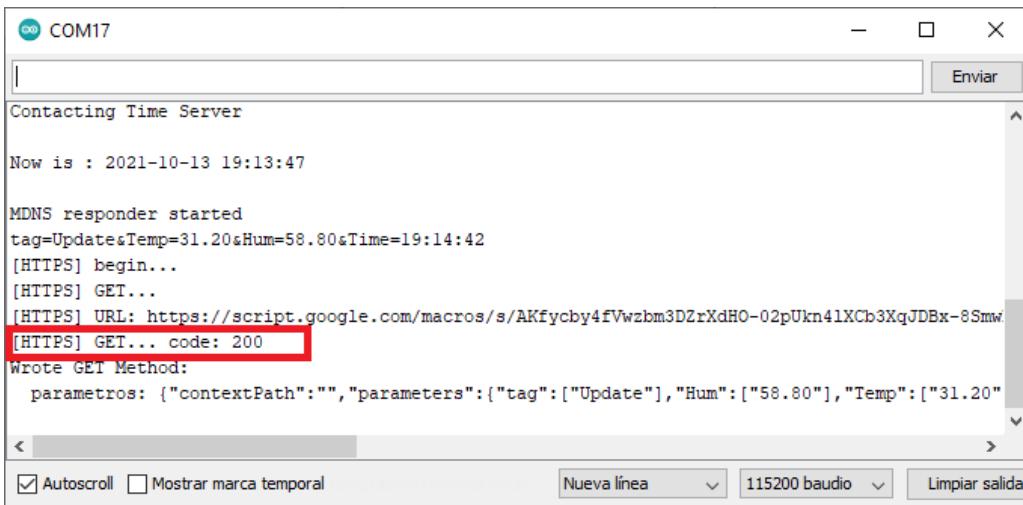


Se requiere proporcionar su nombre y contraseña de su red de WiFi, adicionalmente, proporcionar el Id del proyecto generado anteriormente (del cual,

se acaba de copiar el Id en el párrafo anterior)

```
19 #include <HTTPClient.h> // Biblioteca requerida para manejo de red
20 #include <WiFiClientSecure.h>
21
22 const char *ssid = "mi_red_wifi"; // Definición que establece la red WiFi
23 const char *password = "mi_password_de_red_wifi"; // Definición que establece la contraseña
24
25 // Constantes necesarias para enviar datos a Google
26 const char *GScriptId = "myIdScript"; // Id del proyecto generado por Google
27 const int ledPin = LED_BUILTIN; // Definición que establece el pin del LED
28
```

Compile y descargue el código a su tarjeta de desarrollo NodeMCU-32S. Abra el puerto serial del IDE del Arduino y si observa lo siguiente:



```
COM17
|
Contacting Time Server
Now is : 2021-10-13 19:13:47
MDNS responder started
tag=Update&Temp=31.20&Hum=58.80&Time=19:14:42
[HTTPS] begin...
[HTTPS] GET...
[HTTPS] URL: https://script.google.com/macros/s/AKfycby4fVwzbm3DZrXdHO-02pUkn41XCb3XqJDbx-8Smw...
[HTTPS] GET... code: 200
Wrote GET Method:
parametros: {"contextPath":"","parameters":[{"tag":["Update"],"Hum":["58.80"],"Temp":["31.20"]}]}
```

¡¡¡El código 200 indica que el envío fue exitoso!!!

Tanto el código que se acaba de descargar a la tarjeta NodeMCU-32S como el código implementado en el servidor de Google están comentados para una mayor compresión del funcionamiento de este proyecto.

La manera en que funciona el nodeMCU es muy básica, solamente hace lo siguiente:

- Se conecta a la red WiFi local
- Obtiene la hora y fecha de un servidor especializado en este servicio.
- Cada minuto obtiene los valores del sensor de temperatura y humedad
- Si la lectura del sensor fue exitosa, se envía al servidor de Google (el dato enviado será manejado de acuerdo con lo programado en el Script del proyecto).

En el método “EnviaDatosGoogle” se tiene marcado con rectángulos de colores lo cual se explica a continuación:

```
67     HttpClient httpsGoogle;
68
69     httpsGoogle.setRedirectLimit(10); // Se limita a 10 redirecciones
70     httpsGoogle.setFollowRedirects(HTTPC_FORCE_FOLLOW_REDIRECTS); // Se activa el
71     Serial.print("[HTTPS] begin...\n");
72     if (httpsGoogle.begin(*clientGoogle, url + "?" + payload)) { // HTTPS Se enví
73 //     if (httpsGoogle.begin(*clientGoogle, url)) { // HTTPS Se envía la petición
74
75         Serial.print("[HTTPS] URL: ");
76         Serial.print(url);
77         Serial.print("?");
78         Serial.println(payload);
79
80         // start connection, send HTTP header and GET request
81         Serial.print("[HTTPS] GET...\n");
82         httpCodeGoogle = httpsGoogle.GET(); // Se espera el código resultante de la
83 /* */
84         // start connection, send HTTP header and POST request
85         Serial.print("[HTTPS] POST...\n");
86         httpCodeGoogle = httpsGoogle.POST(payload);
87 */
88         // httpCode will be negative on error
89         // Si el código es positivo indica que el servidor ha respondido.
90         if (httpCodeGoogle > 0) {
```

Hablando del rectángulo verde; el propósito del uso del método setRedirectLimit(10) y setFollowRedirects(HTTPC\_FORCE\_FOLLOW\_REDIRECTS) es el redireccionamiento que Google tiene implementado en su servidor, es decir, la petición se envía a script.google.com y la respuesta se recibe en script.googleusercontent.com, si se usa una petición estándar la respuesta sería un código 302 (sitio ha sido movido temporalmente a la URL). Con esta biblioteca se puede seguir al redireccionamiento y obtener la respuesta del sitio web.

Por lo que respecta al rectángulo azul; ésta es la sección encargada de hacer la petición del tipo GET.

Por lo que respecta al rectángulo rojo; ésta es la sección encargada de hacer la petición del tipo POST.

Es importante tener en cuenta que NO se puede hacer las dos peticiones (GET y POST) al mismo tiempo por lo que, deberá mantener comentada una de las opciones.

En el método “leeSensor” se obtienen los datos del sensor y se almacena en la variable payload en formato de envío de parámetros en html:

```
225  
226 // Se prepara el response en formato estandar para su envio al servidor  
227 // tag=Update&Temp=XX.XX&Hum=XX.XX&Time=XX:XX:XX  
228 // de donde X son los caracteres numericos  
229 payload = "tag=Update&Temp=" + String (charTemp) + "&Hum=" + String (charHum)  
230 Serial.println(payload); // Se observa en el puerto serial el response en fo  
231
```

No olvidar que para el envío de parámetros se tiene que apagar al formato html, el cual, se tiene comentado en el código una breve explicación de su implementación.