



C4.1 Programación Microcontrolador NodeMCU ESP32

Comunicación por medio del protocolo Bluetooth y el NodeMCU ESP32



Instrucciones

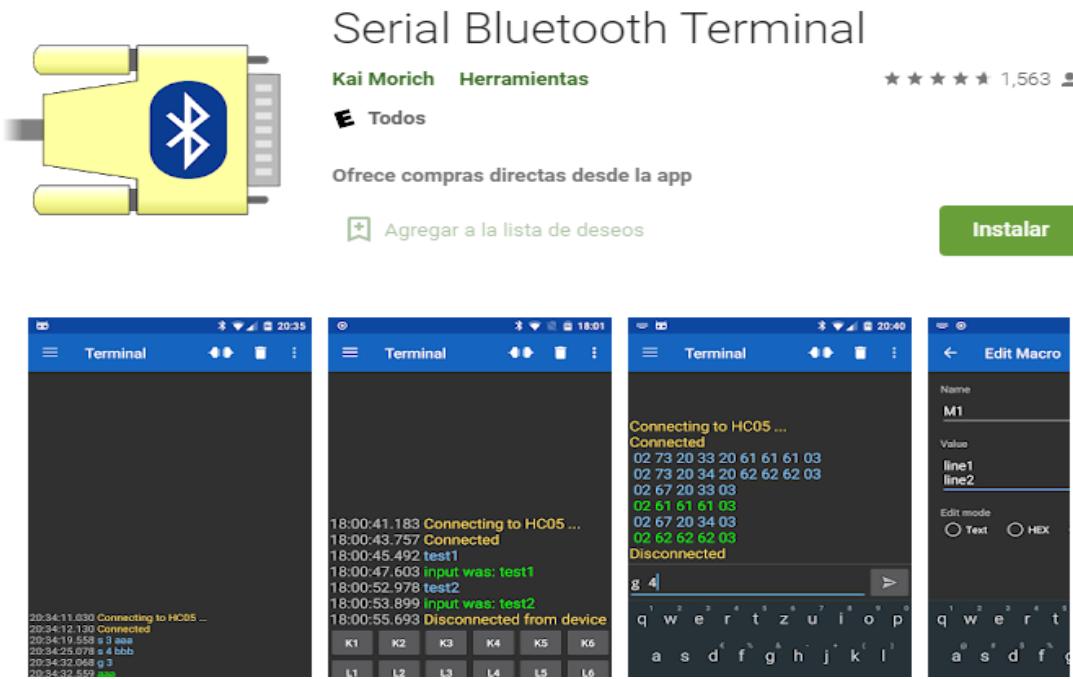
- De acuerdo con la información presentada por el asesor referente al tema, desarrollar lo que se indica dentro del apartado siguiente.
- Toda actividad o reto se deberá realizar utilizando el estilo **MarkDown con extension .md** y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento **single page**, es decir si el documento cuenta con imágenes, enlaces o cualquier documento externo debe ser accedido desde etiquetas y enlaces.
- Es requisito que el archivo .md contenga una etiqueta del enlace al repositorio de su documento en Github, por ejemplo **Enlace a mi GitHub**
- Al concluir el reto el reto se deberá subir a github el archivo .md creado.
- Desde el archivo **.md** se debe exportar un archivo **.pdf** con la nomenclatura **C4.1_NombreAlumno_Equipo.pdf**, el cual deberá subirse a classroom dentro de su apartado correspondiente, para que sirva como evidencia de su entrega; siendo esta plataforma **oficial** aquí se recibirá la calificación de su actividad por individual.
- Considerando que el archivo .pdf, fue obtenido desde archivo .md, ambos deben ser idénticos y mostrar el mismo contenido.
- Su repositorio ademas de que debe contar con un archivo **readme.md** dentro de su directorio raíz, con la información como datos del estudiante, equipo de trabajo, materia, carrera, datos del asesor, e incluso logotipo o imágenes, debe tener un apartado de contenidos o indice, los cuales realmente son ligas o **enlaces a sus documentos .md**, *evite utilizar texto* para indicar enlaces internos o externo.
- Se propone una estructura tal como esta indicada abajo, sin embargo puede utilizarse cualquier otra que le apoye para organizar su repositorio.

```
| readme.md  
| | blog  
| | | C4.1_TituloActividad.md  
| | | C4.2_TituloActividad.md  
| | | C4.3_TituloActividad.md  
| | | C4.4_TituloActividad.md  
| | | C4.5_TituloActividad.md  
| | img  
| | docs  
| | | A4.1_TituloActividad.md  
| | | A4.2_TituloActividad.md
```

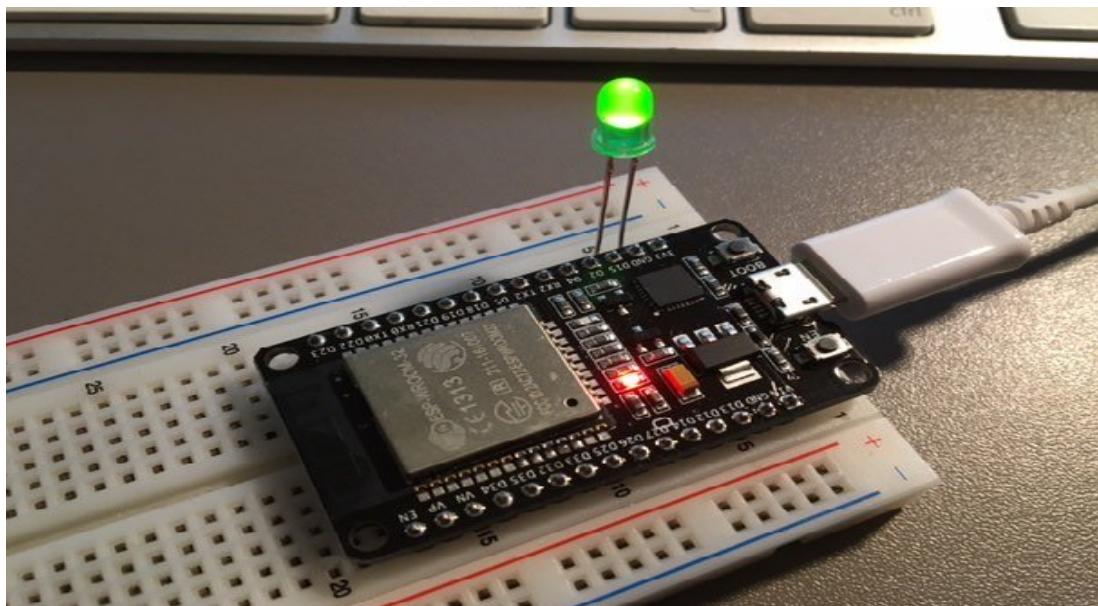


Desarrollo

1. Busque en Play Store la app "Serial Bluetooth terminal" e instale a su dispositivo móvil.



2. Apóyese del siguiente circuito, colocando una resistencia y un diodo led en serie a una de las terminales del Nodemcu ESP32 de tal forma que puede general una señal digital.



3. Analice y utilice el código que se indica a continuación para emparejar su dispositivo móvil con el dispositivo ESP32 via bluetooth

```
//This example code is in the Public Domain (or CC0 licensed, at your option.)
//By Evandro Copercini - 2018
//
//This example creates a bridge between Serial and Classical Bluetooth (SPP)
//and also demonstrate that SerialBT have the same functionalities of a normal
//Serial

#include "BluetoothSerial.h"
```

```
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

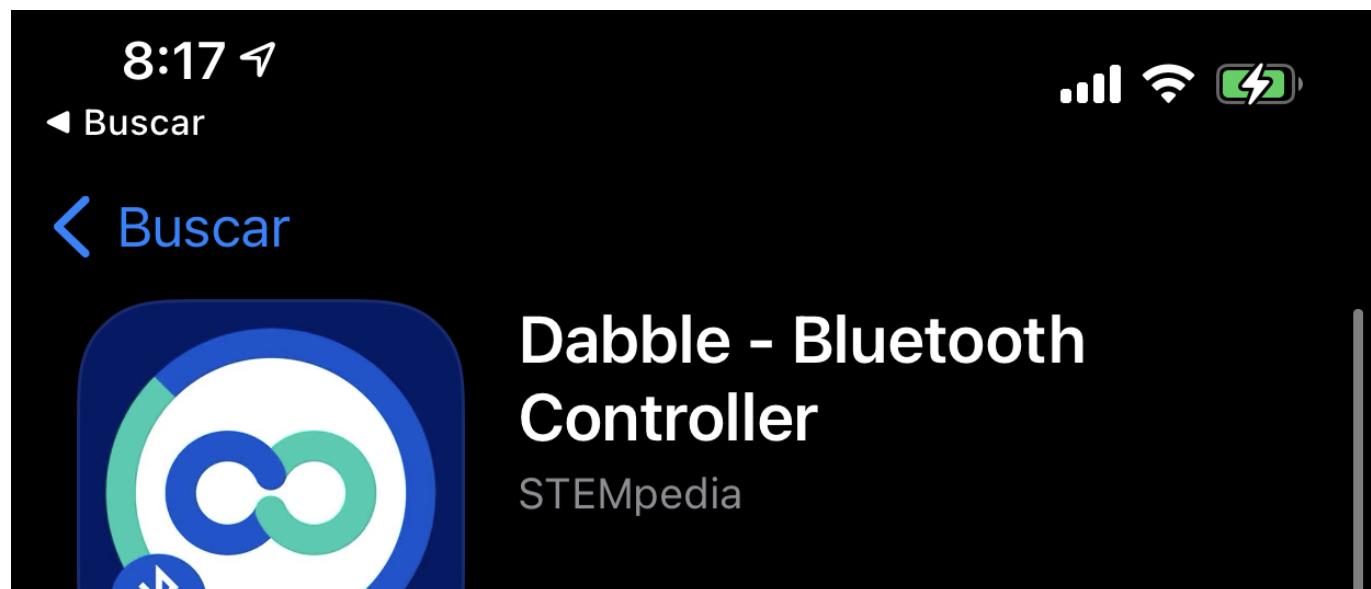
BluetoothSerial SerialBT;

void setup() {
    Serial.begin(115200);
    SerialBT.begin("ESP32test"); //Bluetooth device name
    Serial.println("The device started, now you can pair it with bluetooth!");
}

void loop() {
    if (Serial.available()) {
        SerialBT.write(Serial.read());
    }
    if (SerialBT.available()) {
        Serial.write(SerialBT.read());
    }
    delay(20);
}
```

4. Compile y cargue el programa a su dispositivo NodeMCU-ESP32
5. Empareje su dispositivo móvil con el ESP32, y utilice la app bajada del play store para establecer la comunicación.
6. Abra la terminal serial del IDE de Arduino, y desde el móvil envíe cualquier instrucción la cual se deberá observar en la terminal serial del IDE Arduino.
7. Una vez realizado lo anterior, agregue instrucciones al programa para que le indique al ESP32 que encienda o apague el LED que ha sido colocado en el circuito desde su dispositivo móvil y la app instalada.
8. Inserte aquí las imágenes que considere como evidencias para demostrar el resultado obtenido.

Yo tuve que utilizar la aplicación que se muestra a continuación para poder conectar el ESP32 a mi celular. Al principio batale un poco, pero por fin lo pude resolver.



 **ABRIR** 

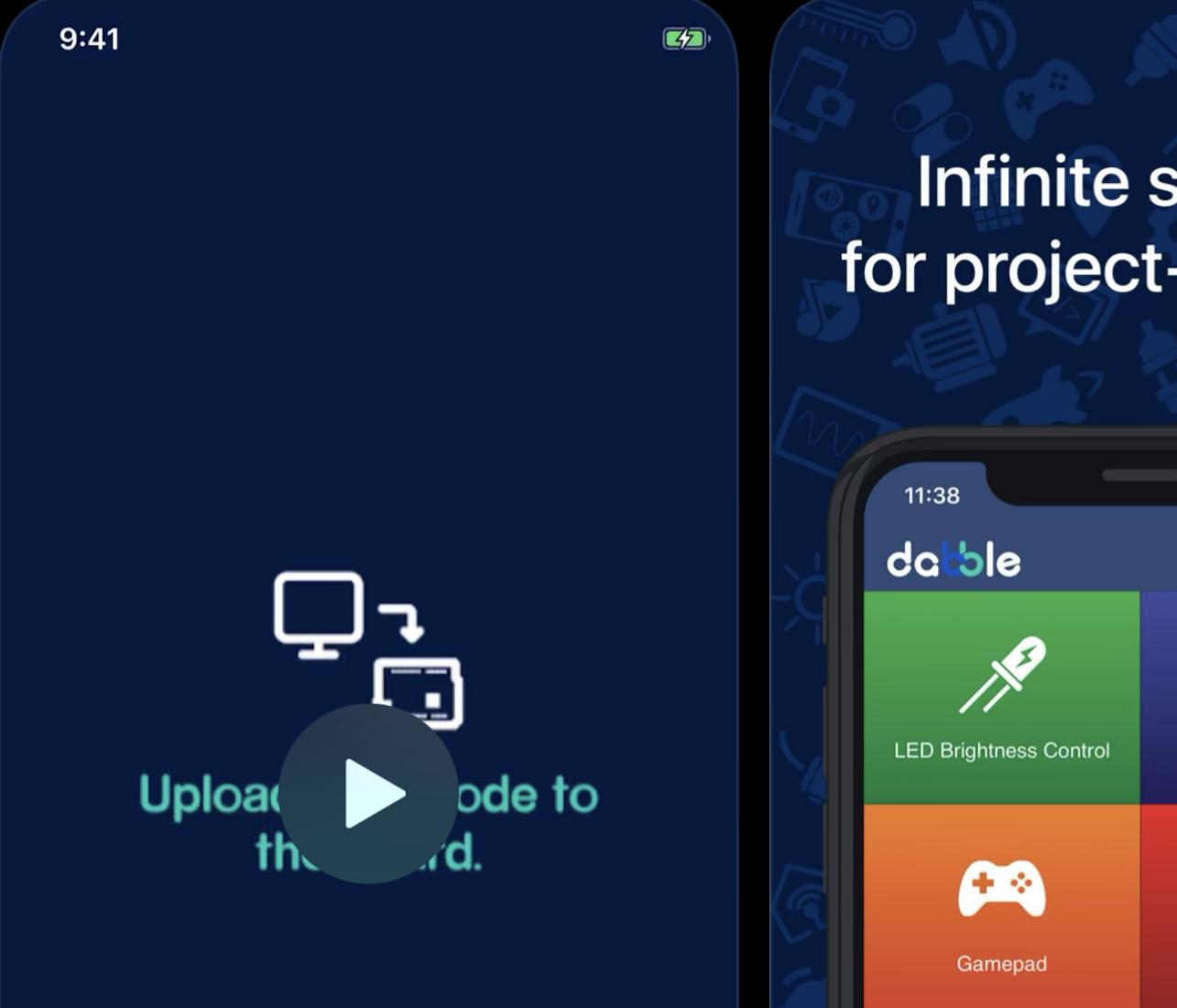
26 CALIFICACIONES 4.4 	EDAD 4+ años	DESARROLLADOR  STEMpedia	IDIOMA  Inglés
--	---------------------------	---	---

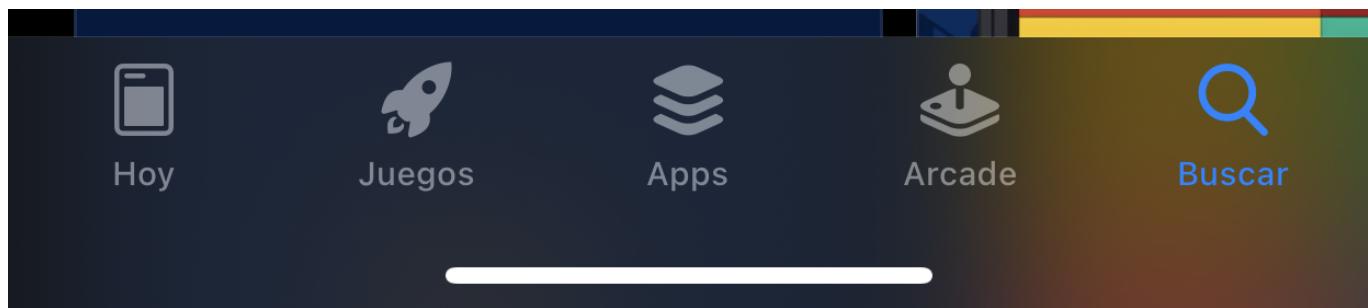
Novedades

Versión 2.0 Hace 7 meses

*Bug fixes.

Previsualización

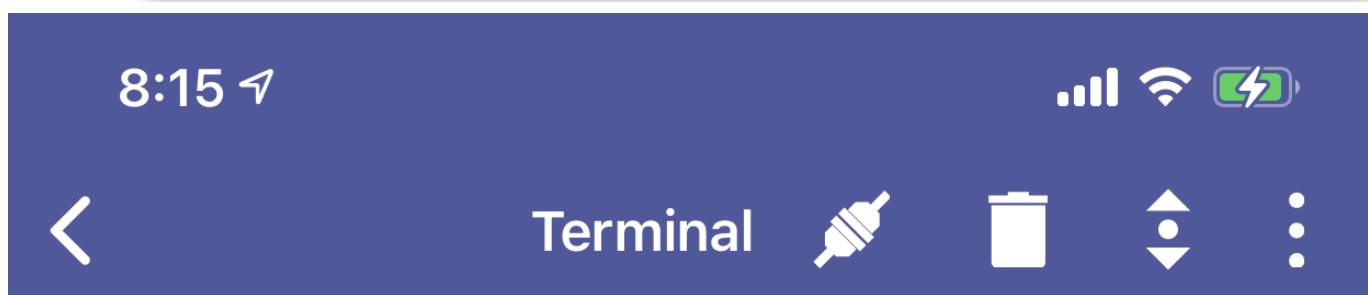
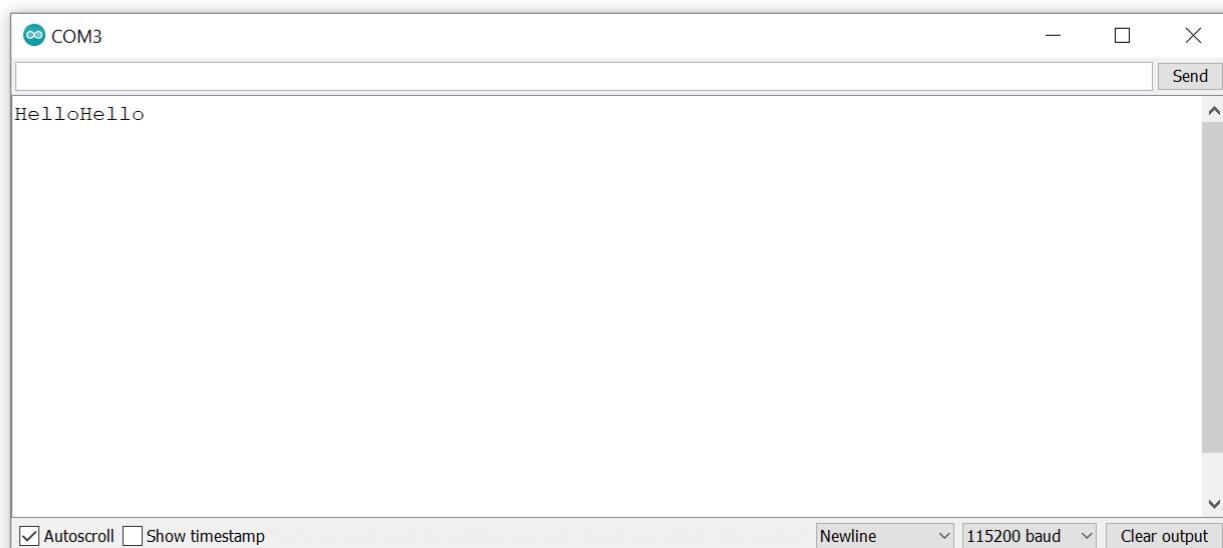




```
#include <DabbleESP32.h>

void setup() {
    Serial.begin(115200);
    Dabble.begin("MiESP32");
    Dabble.waitForAppConnection();
    Notification.clear();
    Terminal.print("Listo para usar");
}

void loop()
{
    Dabble.processInput();
    if(Serial.available())
    {
        Terminal.write(Serial.read());
    }
    if (Terminal.available())
    {
        Serial.write(Terminal.read());
    }
    delay(20);
}
```



8:13:42:1342 PM, December 05, 2020

8:13:42:1342 PM, December 05, 2020

8:13:42:1342 PM, December 05, 2020

Listo para usar

8:13:42:1342 PM, December 05, 2020

Hello

8:13:56:1356 PM, December 05, 2020

H

8:14:05:145 PM, December 05, 2020

O

8:14:05:145 PM, December 05, 2020

I

8:14:05:145 PM, December 05, 2020

a

8:14:05:145 PM, December 05, 2020

8:14:06:146 PM, December 05, 2020

Hello

8:14:55:1455 PM, December 05, 2020



Actividad de encender y apagar el led.

```
#include <DabbleESP32.h>
unsigned long lasttime=0;

#define LEDR 25 // Se declaran los pines para cada parte del led RGB
#define LEDG 26
#define LEDB 27

#define R_channel 0 // Canales dedicados a cada uno de los colores
#define G_channel 1
#define B_channel 2

#define pwm_Frequency 5000 // Frecuencia

#define pwm_resolution 8 // 8 bit de resolucion

String message = "";
char incomingChar;

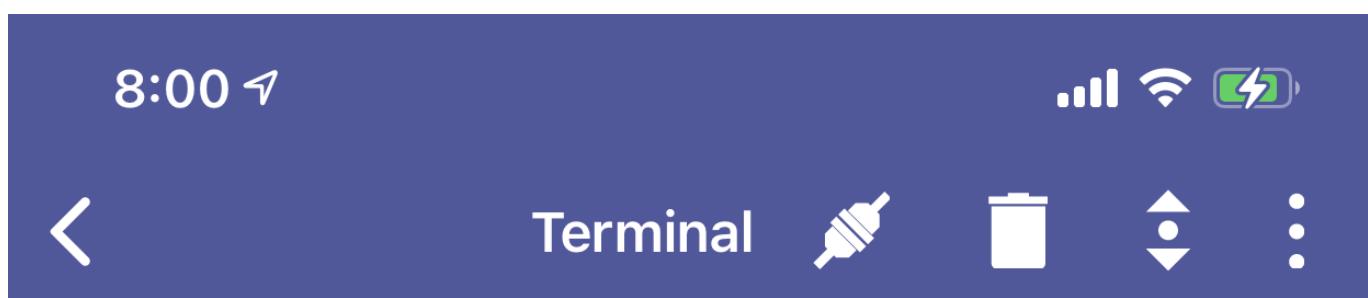
void setup() {
    Serial.begin(115200);
    Dabble.begin("MiESP32");
    Dabble.waitForAppConnection();
    Notification.clear();
    Terminal.print("Listo para usar");

        // Configuracion de las funciones del led RGB
    ledcAttachPin(LEDR, R_channel);
    ledcAttachPin(LEDG, G_channel);
    ledcAttachPin(LEDB, B_channel);

    // Conectar el canal al GPIO para controlarlo
    ledcSetup(R_channel, pwm_Frequency, pwm_resolution);
    ledcSetup(G_channel, pwm_Frequency, pwm_resolution);
```

```
    ledcSetup(B_channel, pwm_Frequency, pwm_resolution);
}

void loop()
{
    Dabble.processInput();
    if(Serial.available())
    {
        Terminal.write(Serial.read());
    }
    if(Terminal.available())
    {
        char incomingChar = Terminal.read();
        if(incomingChar != '\n')
        {
            message += String(incomingChar);
        }
        if(message == "1")
        {
            ledcWrite(R_channel, 255);
            Serial.write("~~Prendido~~");
        }
        if (message == "0")
        {
            ledcWrite(R_channel, 0);
            Serial.write("~~Apagado~~");
        }
        else
        {
            message = "";
        }
    }
    delay(20);
}
```



8:00:19:019 PM, December 05, 2020

8:00:19:019 PM, December 05, 2020

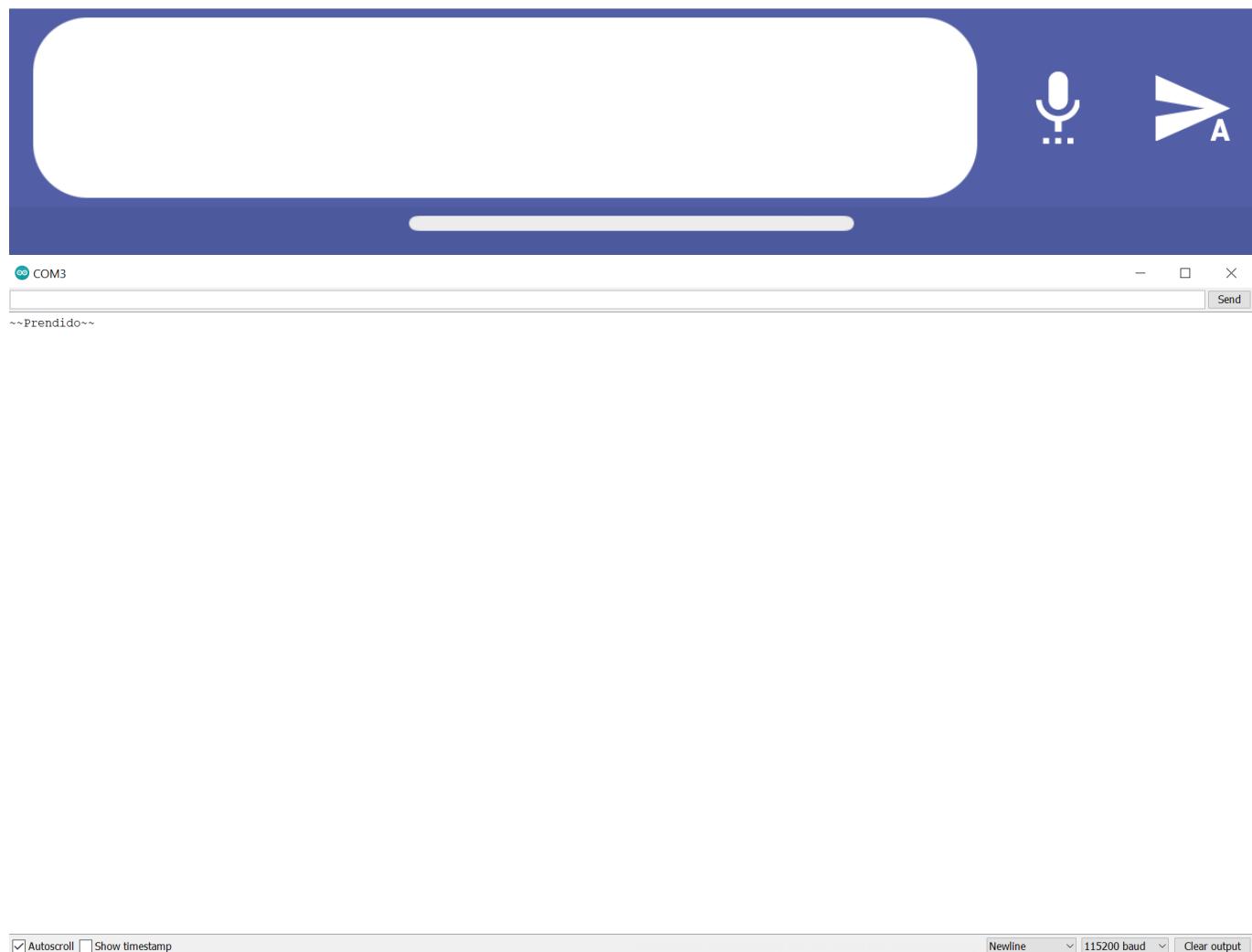
8:00:19:019 PM, December 05, 2020

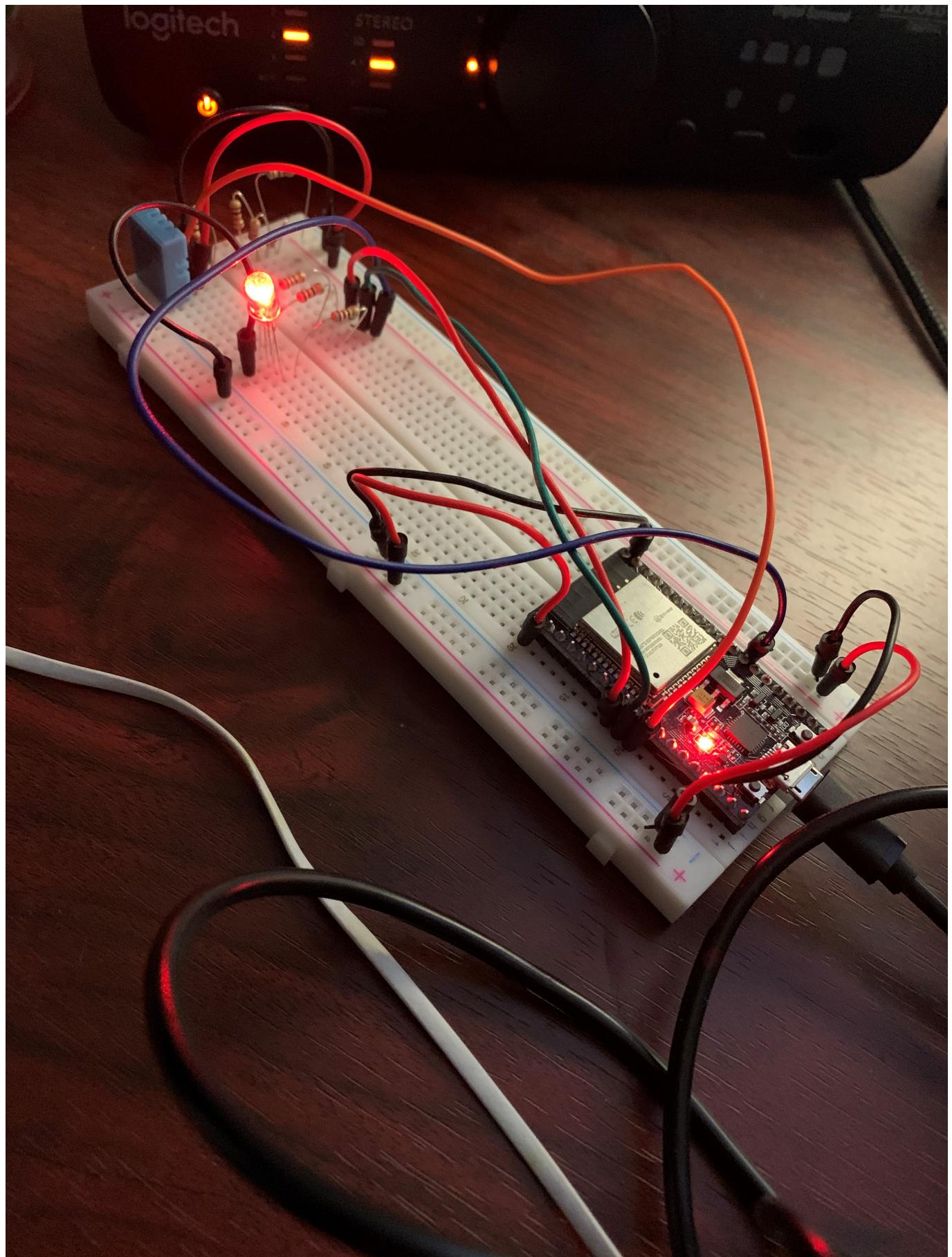
Listo para usar

8:00:20:020 PM, December 05, 2020

1

8:00:25:025 PM, December 05, 2020





8:01 ↗



Terminal

11 / 16

8:00:19:019 PM, December 05, 2020

Listo para usar

8:00:20:020 PM, December 05, 2020

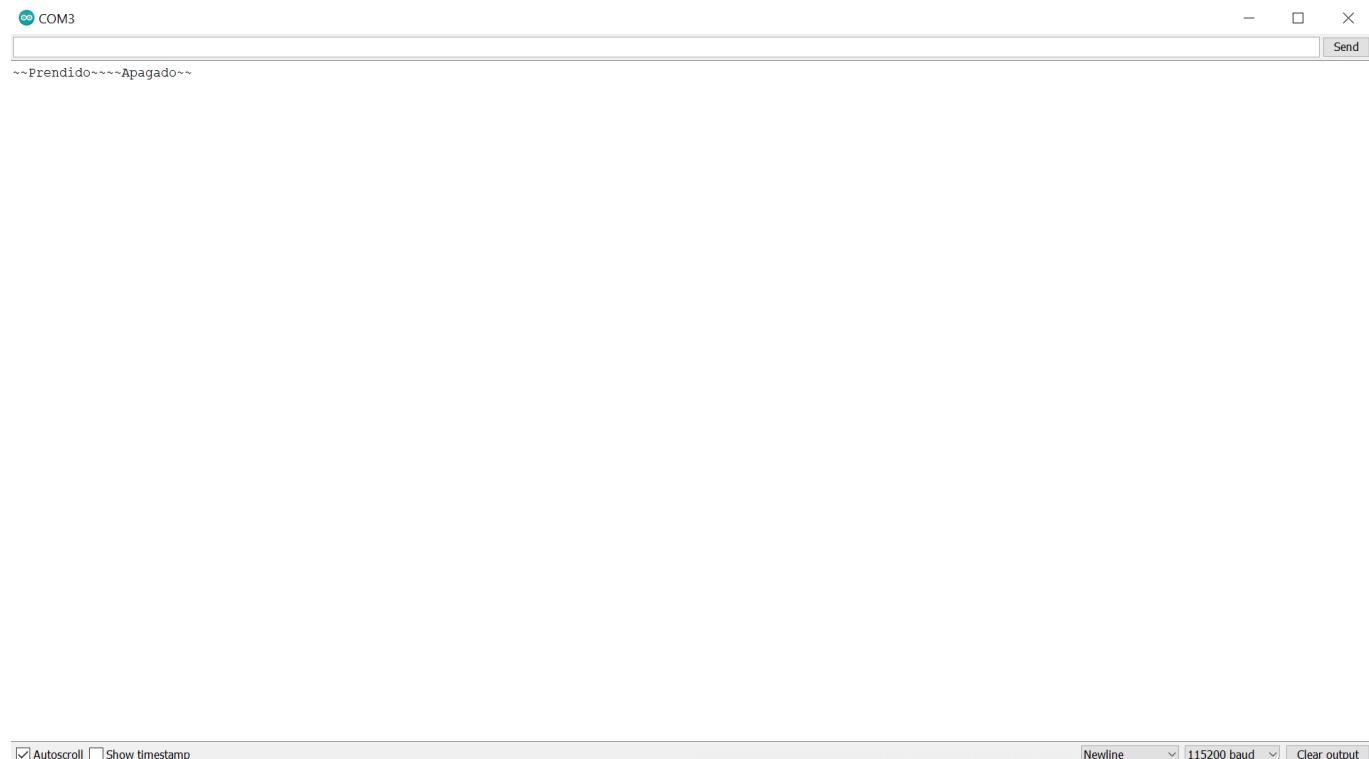
1

8:00:25:025 PM, December 05, 2020

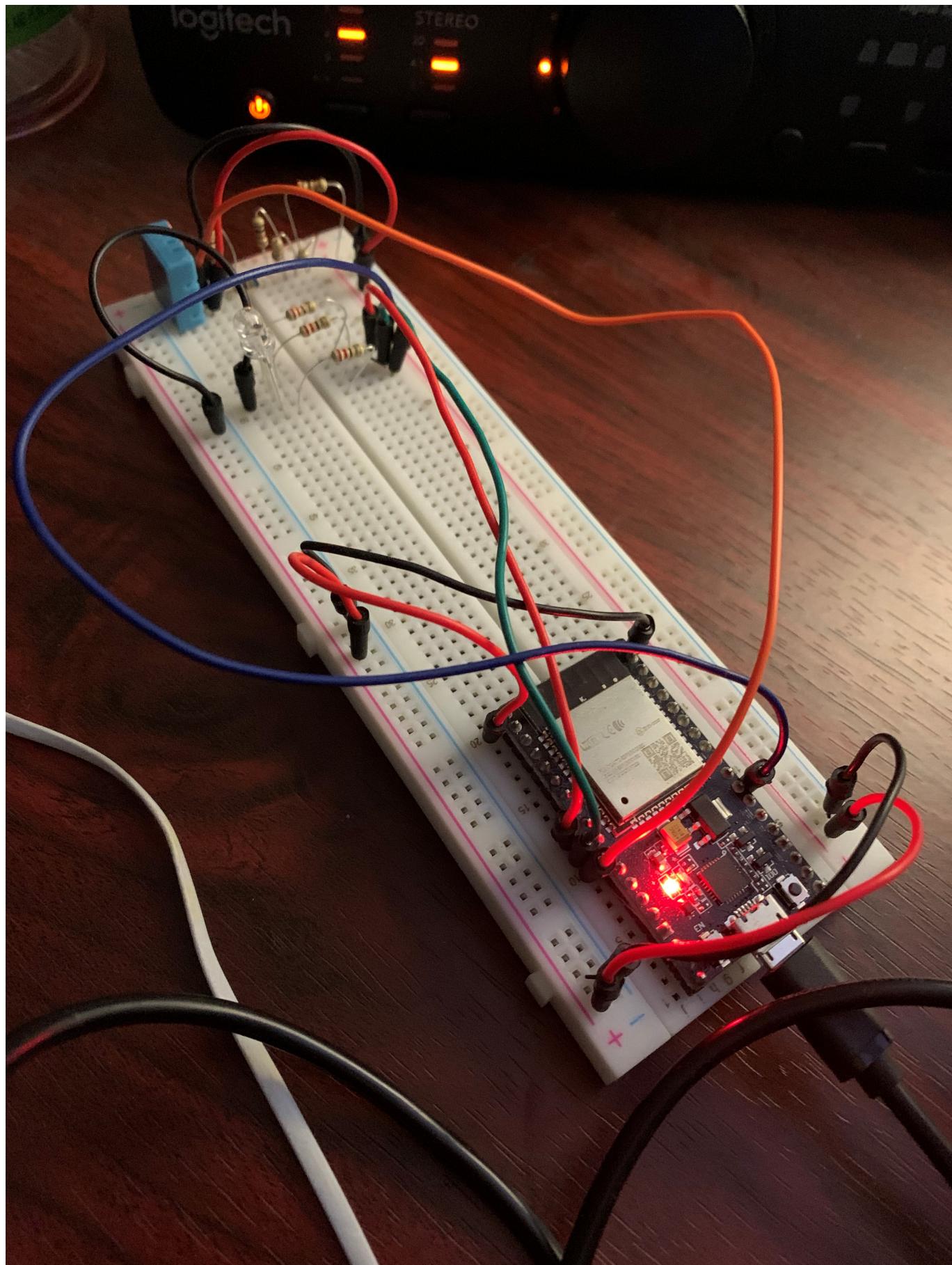
0

8:01:22:122 PM, December 05, 2020





The screenshot shows a terminal window titled "COM3". The main text area contains the string "~~Prendido~~~~Apagado~~". At the bottom of the window, there is a toolbar with several buttons and dropdown menus. On the left side of the toolbar, there are checkboxes for "Autoscroll" (checked) and "Show timestamp" (unchecked). On the right side, there are dropdown menus for "Newline" (set to "Newline"), "115200 baud" (set to "115200 baud"), and a "Clear output" button.



Rubrica

Criterios	Descripción	Puntaje
-----------	-------------	---------

Criterios	Descripción	Puntaje
Instrucciones	Se cumple con cada uno de los puntos indicados dentro del apartado Instrucciones?	20
Desarrollo	Se respondió a cada uno de los puntos solicitados dentro del desarrollo de la actividad?	80

 [Ir a Github](#)