



C4.1 Programación Microcontrolador NodeMCU ESP32

Comunicación por medio del protocolo Bluetooth y el NodeMCU ESP32



Instrucciones

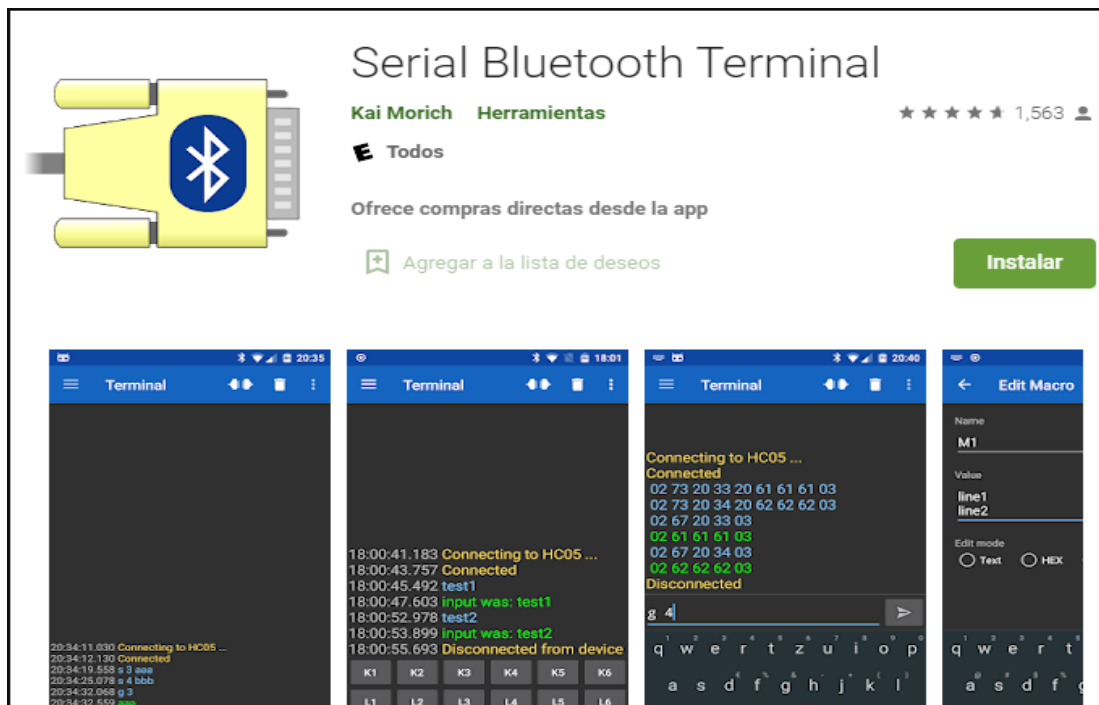
- De acuerdo con la información presentada por el asesor referente al tema, desarrollar lo que se indica dentro del apartado siguiente.
- Toda actividad o reto se deberá realizar utilizando el estilo **Markdown con extension .md** y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento **single page**, es decir si el documento cuanta 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
```

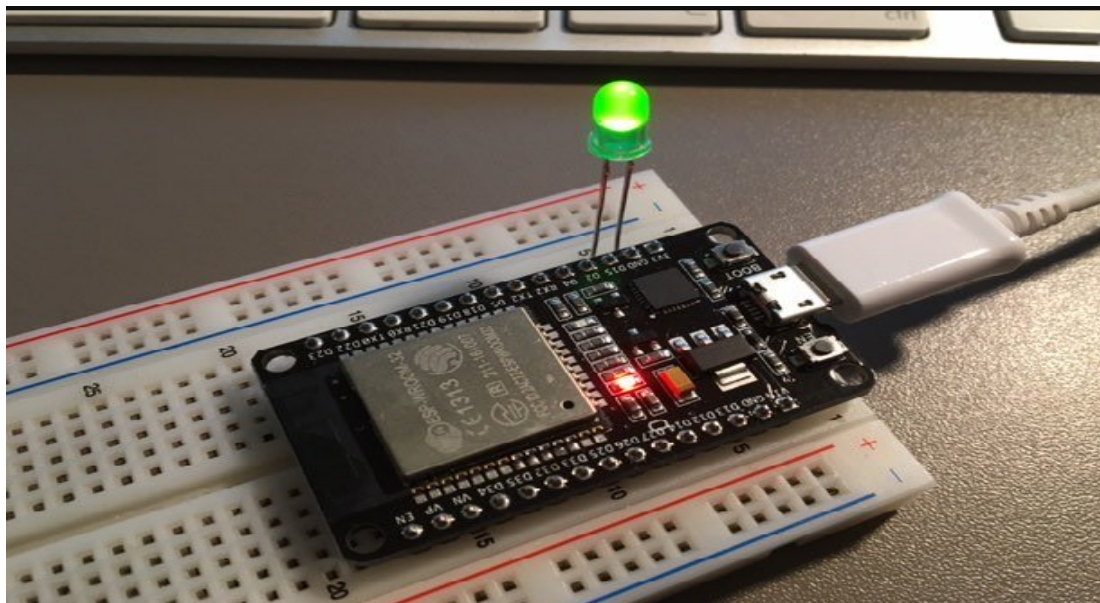


Desarrollo

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



2. Apóyese del siguiente circuito, colocando una resistencia y un diodo led en seria 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 movil 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.

SIMULACION >

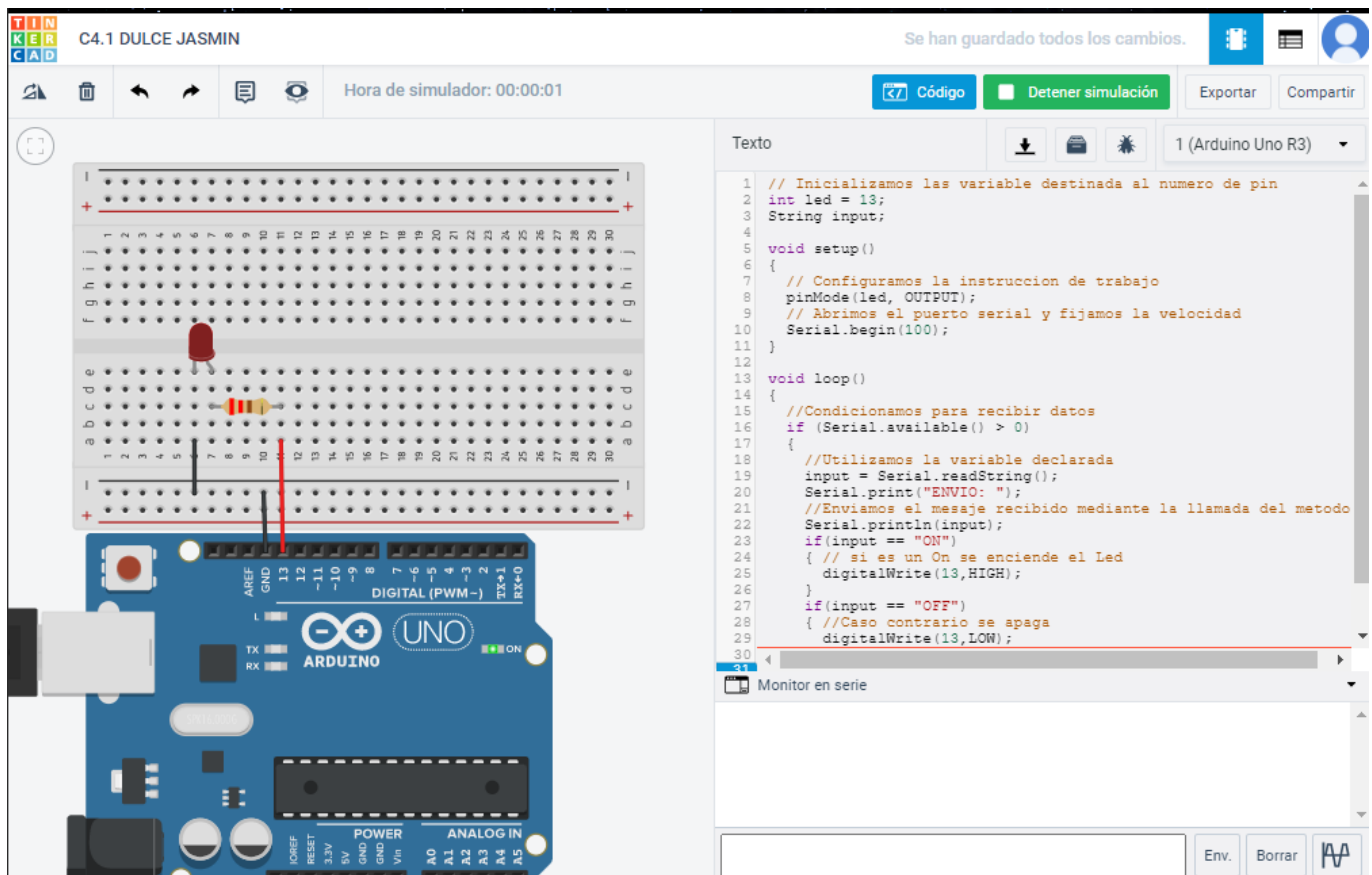
```
// Inicializamos la variable destinada al número de pin
int led = 13;
String input;

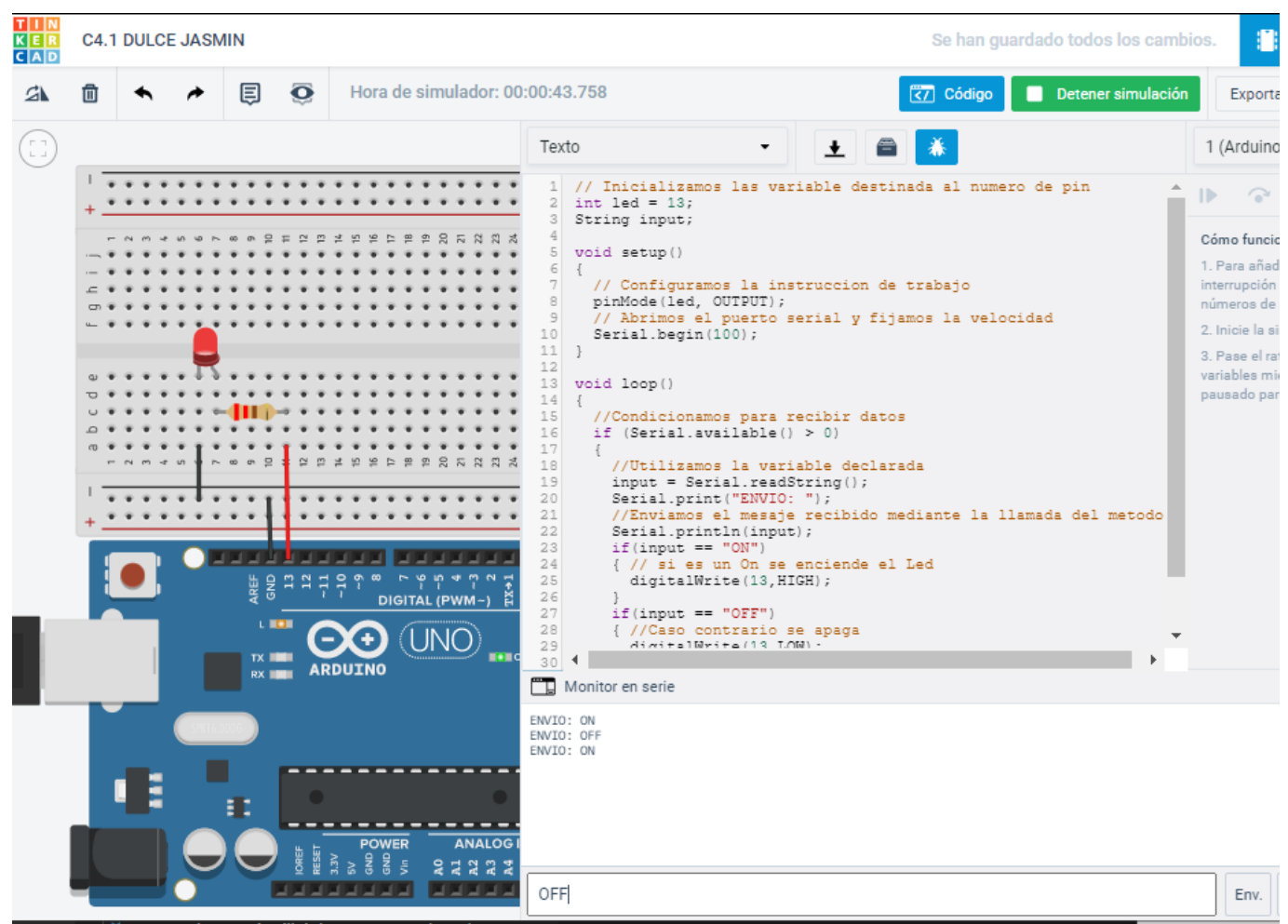
void setup()
{
  // Configuramos la instrucción de trabajo
  pinMode(led, OUTPUT);
  // Abrimos el puerto serial y fijamos la velocidad
  Serial.begin(100);
}
```

```

void loop()
{
  //Condicionamos para recibir datos
  if (Serial.available() > 0)
  {
    //Utilizamos la variable declarada
    input = Serial.readString();
    Serial.print("ENVIO: ");
    //Enviamos el mensaje recibido mediante la llamada del método
    Serial.println(input);
    if(input == "ON")
    { // si es un On se enciende el Led
      digitalWrite(13,HIGH);
    }
    if(input == "OFF")
    { //Caso contrario se apaga
      digitalWrite(13,LOW);
    }
  }
}

```

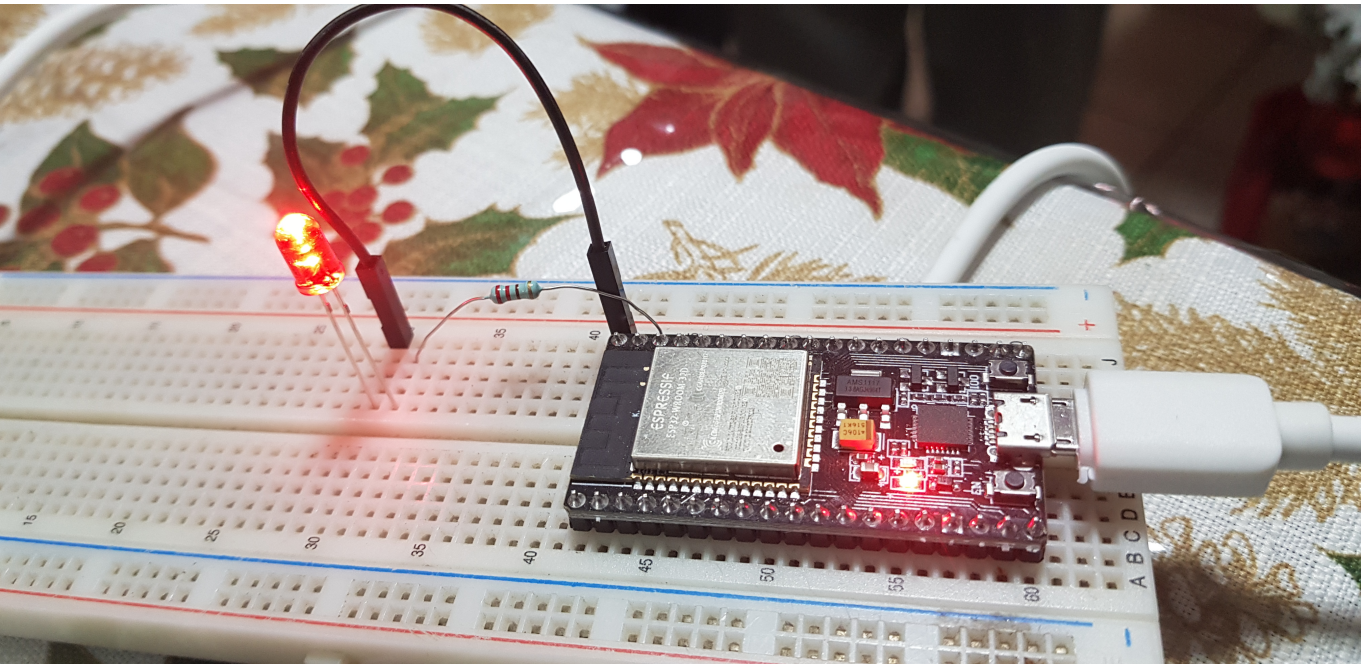




EVIDENCIAS

Venegas Medina Jose Alfredo:

VIDEO > [Evidencia](#)



Programa_C4.1 Arduino 1.8.14 Hourly Build 2020/10/09 12:33

Archivo Editar Programa Herramientas Ayuda

```

Programa_C4.1 $
//Libreria para utilizar Bluetooth
#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;

//Constantes para declarar pin del LED
const int ledPin = 22;
String message = "";
char incomingChar;

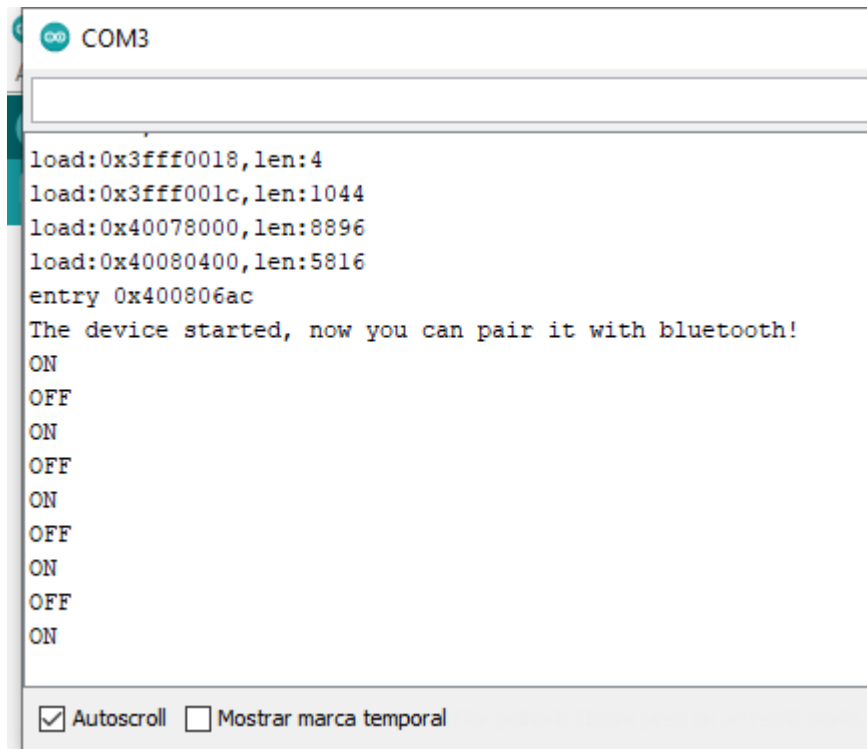
void setup() {
  //LED como salida
  pinMode(ledPin, OUTPUT);
  Serial.begin(115200);

  //Se asigna nombre para encontrar el ESP32 en Bluetooth
  SerialBT.begin("ESP32test"); //Bluetooth device name
  //Mensaje para indicar que se inicio la libreria Bluetooth
  Serial.println("The device started, now you can pair it with bluetooth!");
}

void loop() {
  //Si la libreria de Bluetooth esta activada va a leer el puerto serial
  if (SerialBT.available()){
    char incomingChar = SerialBT.read();
    if (incomingChar != '\n'){
      message += String(incomingChar);
    }
    else{
      message = "";
    }
    Serial.write(incomingChar);
  }
  // Si el mensaje es ON se prende el LED
  if (message == "ON"){
    digitalWrite(ledPin, HIGH);
  }

  //Si el mensaje es OFF se apaga el LED
  else if (message == "OFF"){
    digitalWrite(ledPin, LOW);
  }
  delay(20);
}

```



```

COM3
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:8896
load:0x40080400,len:5816
entry 0x400806ac
The device started, now you can pair it with bluetooth!
ON
OFF
ON
OFF
ON
OFF
ON
OFF
ON

```

☒ Autoscroll ☐ Mostrar marca temporal

Rubrica

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

  [ENLACE - MI GITHUB](#)