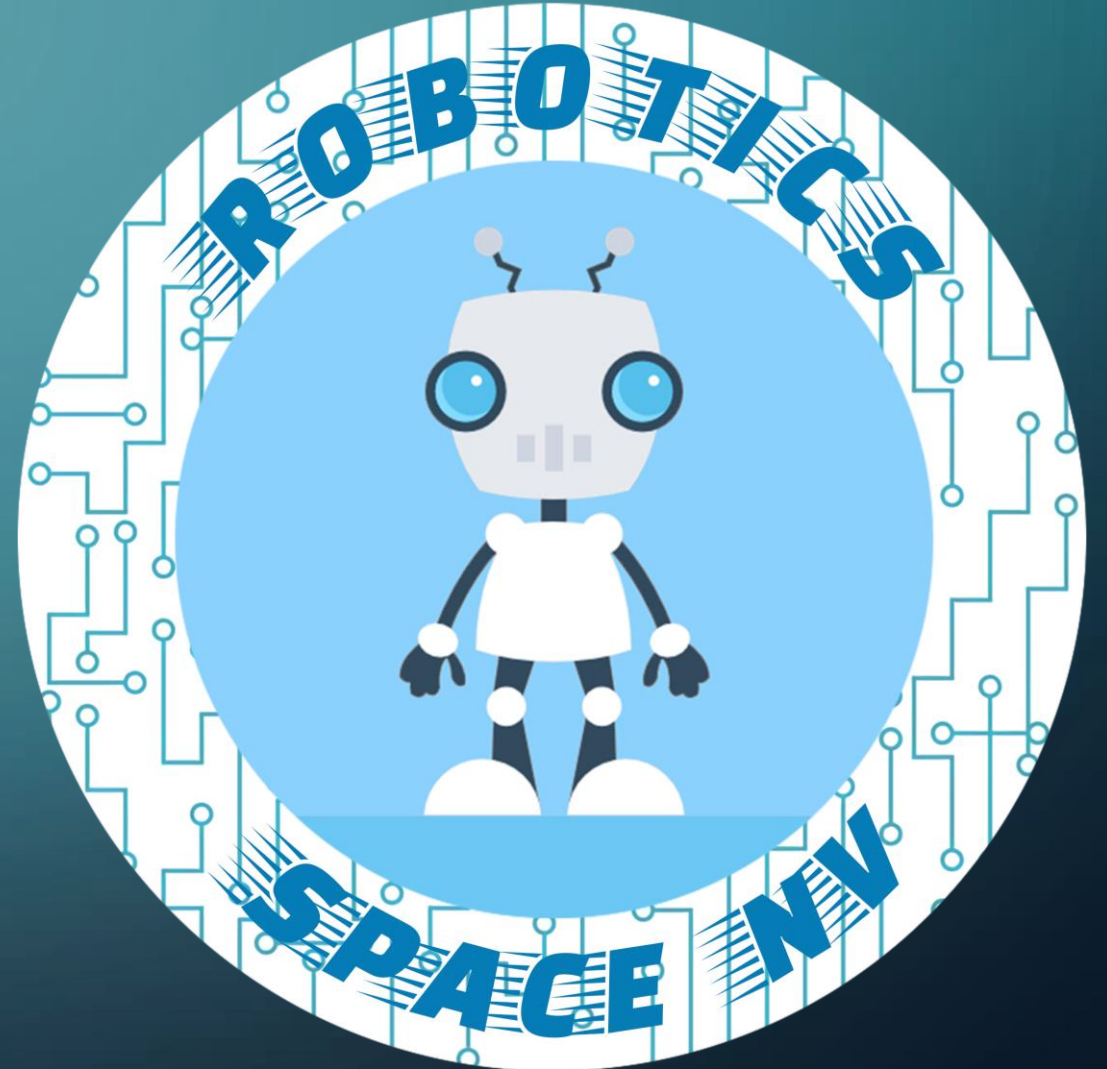


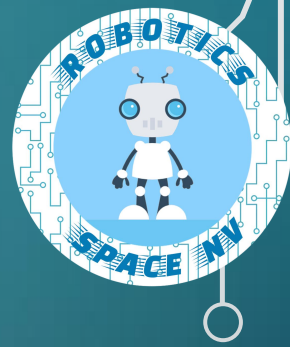
ARDUINO MÓDULO III

SESIÓN 2

- GSM/GPRS
- CARACTERÍSTICAS
- EJERCICIOS
- GPRS + BLYNK
- GPRS + THINGSPEAK



TUTOR:NAGIB LUIS VALLEJOS M.



GSM / GPRS

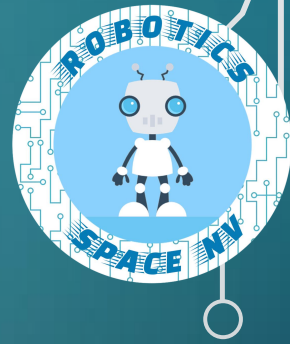
GSM: Es el **Sistema Global de Comunicaciones Móviles** del inglés *Global System for Mobile communications*.

GPRS: Es el **Servicio General de Paquetes vía Radio**, es decir, una extensión del GSM basada en la transmisión por paquetes que ofrece un servicio más eficiente para las comunicaciones de datos, especialmente en el caso del acceso a Internet.

La velocidad máxima del GPRS es de 171kb/s, aunque en la práctica es bastante más pequeña.

TUTOR:NAGIB LUIS VALLEJOS M.

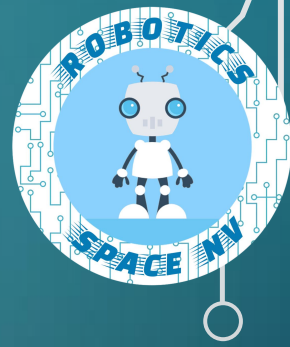
FRECUENCIA DE OPERADORAS EN EL MUNDO



Banda	Nombre	Canales	Uplink (MHz)	Downlink (MHz)	Notas
GSM 850	GSM 850	128 - 251	824,0 - 849,0	869,0 - 894,0	Usada en los EE.UU., Sudamérica y Asia .
GSM 900	P-GSM 900	0-124	890,0 - 915,0	935,0 - 960,0	La banda con que nació GSM en Europa y la más extendida
	E-GSM 900	974 - 1023	880,0 - 890,0	925,0 - 935,0	E-GSM, extensión de GSM 900
	R-GSM 900	n/a	876,0 - 880,0	921,0 - 925,0	GSM ferroviario (GSM-R).
GSM1800	GSM 1800	512 - 885	1710,0 - 1785,0	1805,0 - 1880,0	
GSM1900	GSM 1900	512 - 810	1850,0 - 1910,0	1930,0 - 1990,0	Usada en Norteamérica , incompatible con GSM-1800 por solapamiento de bandas.

TUTOR:NAGIB LUIS VALLEJOS M.

FRECUENCIA DE OPERADORAS EN BOLIVIA PARA GSM/GPRS



GSM/GPRS de
1900 MHz

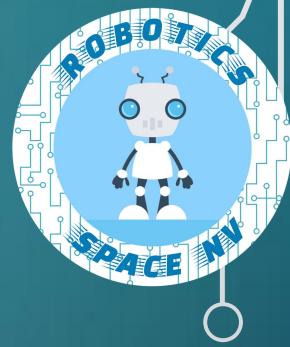


GSM/GPRS,
de 850 MHz

GSM/GPRS
850 MHz/1900
MHz



TUTOR: NAGIB LUIS VALLEJOS M.



GSM/GPRS PARA ARDUINO

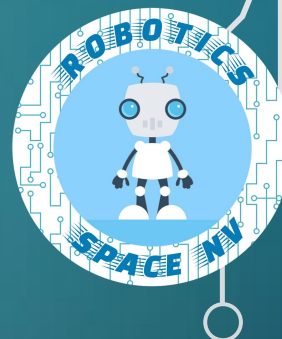
Existen diferentes tipos:



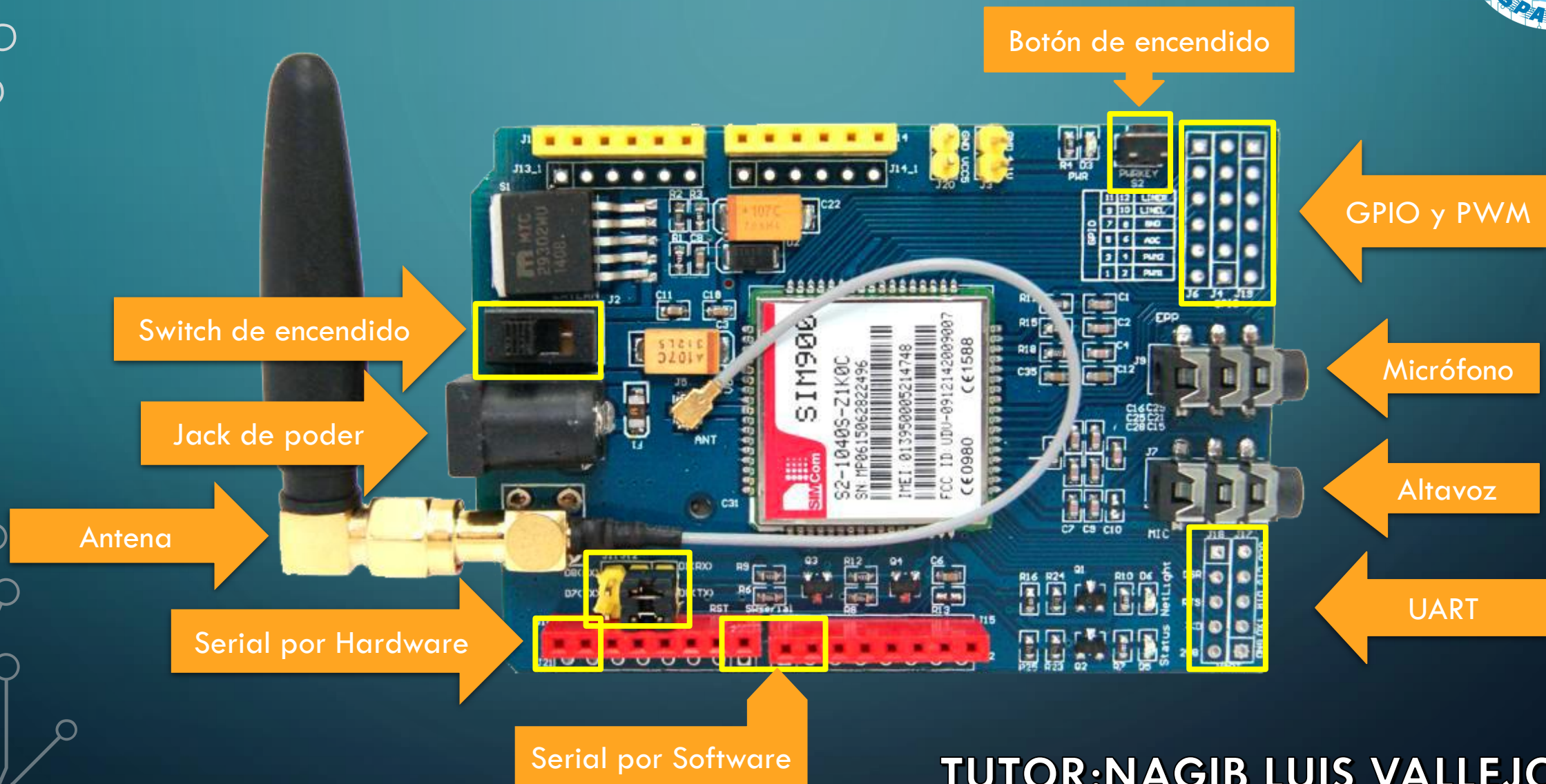
Módulo cuatribanda GSM de:
850/900/1800/1900 MHz.



TUTOR:NAGIB LUIS VALLEJOS M.

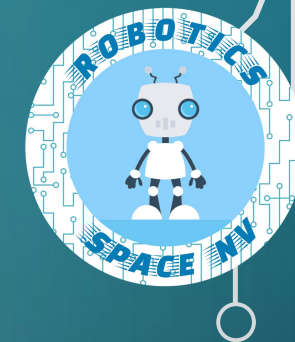


SHIELD GSM/GPRS SIM900



TUTOR: NAGIB LUIS VALLEJOS M.

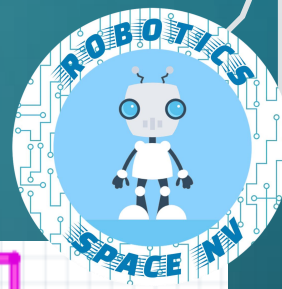
COMANDOS AT MÁS IMPORTANTES



COMANDOS AT	DESCRIPCIÓN
AT+CREG?	Comprueba la conexión de la red
ATD###	Realiza una llamada. Reemplazar ### por número telefónico
ATA	Contesta una llamada entrante
ATH	Finaliza la llamada
AT+CMGF=1	Configura el modo texto para enviar y recibir mensajes
AT+CMGS="#"	Número a que se desea enviar el mensaje
AT+CLIP=1	Activa el identificador de llamadas
AT+CNMI=2,2,0,0,0	Configura el modulo para mostrar los sms por el puerto serie
AT+CGATT=1	Activa la red GPRS
AT+CSTT="APN","us","pass"	Permite configurar APN, usuario y contraseña
AT+CIICR	Permite comunicación CSD, es decir, transmisión de datos
AT+CIFSR	Permite mostrar nuestra IP local
AT+CIPSPRT=0	Establece el inicio de la comunicación
AT+CIPSTART="TCP","IP","Port"	Permite configurar tipo de conexión, dirección IP al que deseamos conectarnos y contraseña
AT+CIPSEND	Permite enviar por TCP/UDP
AT+CIPSHUT	Cierra el contexto PDP del GPRS
AT+CIPSTATUS	Comprueba el estado de conexión
AT+CIPMUX=0	Configura conexión única o múltiple. 0=Única / 1=Múltiple

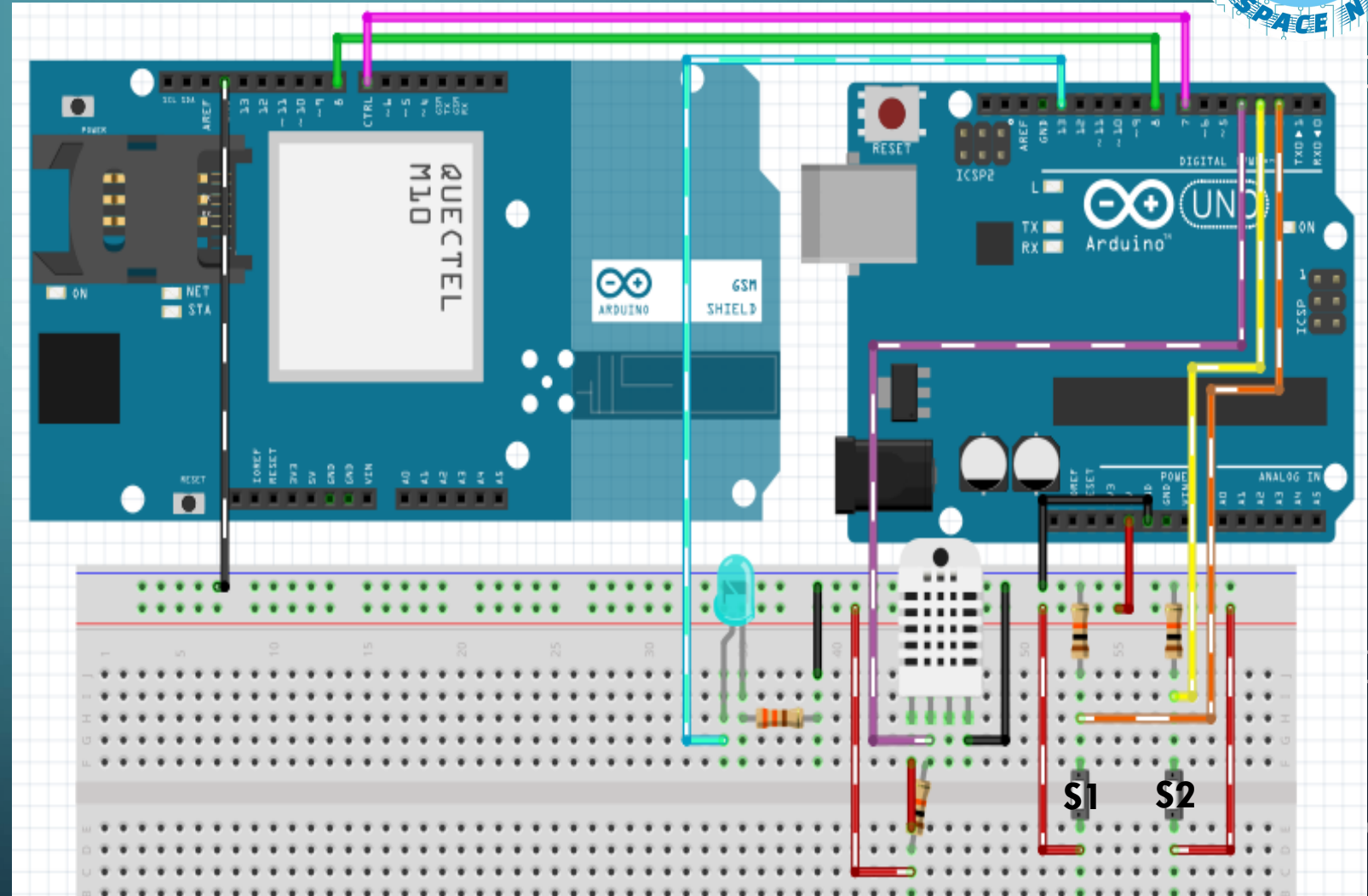
TUTOR:NAGIB LUIS VALLEJOS M.

EJERCICIO 1 – CIRCUITO



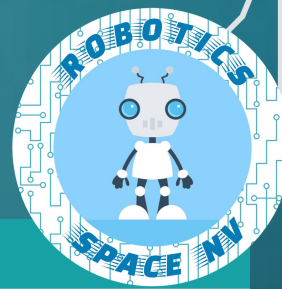
Si presionamos S1, realiza una llamada y se enciende un led mientras ésta esté activa.

Si presionamos S2, envía un SMS con los datos censados a un número telefónico.



TUTOR:NAGIB LUIS VALLEJOS M.

EJERCICIO 1 – SOLUCIÓN



S2-E1

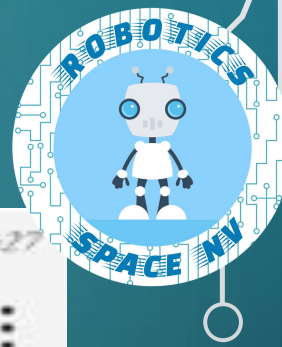
```
1 #include <SoftwareSerial.h>
2 #include <DHT.h>
3
4 SoftwareSerial SIM900(7,8);
5 DHT dht(4,DHT22);
6
7 int led=l3, bi=2, bd=3,s1,s2;
8 void setup(){
9     digitalWrite(9,1);
10    delay(1000);
11    digitalWrite(9,0);
12    SIM900.begin(19200);
13    Serial.begin(19200);
14    dht.begin();
15    Serial.println("Buscando señal");
16    pinMode(led,OUTPUT);
17    pinMode(bi,INPUT);
18    pinMode(bd,INPUT);
19    delay(25000);
20    Serial.println("Tiempo de espera finalizado");
21 }
22
```

```
23 void mensaje_sms(){
24     float h = dht.readHumidity();
25     float t = dht.readTemperature();
26     float i = dht.computeHeatIndex(t,h,false);
27     String sms="Datos sensor: Humedad=";
28     sms.concat(h);
29     sms.concat("% HR - Temperatura=");
30     sms.concat(t);
31     sms.concat("°C - Indice de calor=");
32     sms.concat(i);
33     sms.concat("°C");
34     Serial.println(sms);
35     Serial.println("Enviando SMS...");
36     SIM900.print("AT+CMGF=1\r");
37     delay(1000);
38     SIM900.println("AT+CMGS=\"60628973\"");
39     delay(1000);
40     SIM900.println(sms);// Texto del SMS
41     delay(100);
42     SIM900.println((char)26);
43     delay(100);
44     SIM900.println();
45     delay(5000);
46     Serial.println("SMS enviado");
47 }
```

```
48 void llamada(){
49     Serial.println("Realizando llamada...");
50     SIM900.println("ATD60628973;"); //Comando AT
51     delay(20000); // Espera 20 segundos mientras
52     SIM900.println("ATH"); // Cuelga la llamada
53     delay(1000);
54     Serial.println("Llamada finalizada");
55 }
56
57 void loop(){
58     s1=digitalRead(bi);
59     s2=digitalRead(bd);
60     if(s1){
61         digitalWrite(led,1);
62         llamada();
63         digitalWrite(led,0);
64     }
65     if(s2){
66         mensaje_sms();
67         delay(5000);
68     }
69 }
```

TUTOR: NAGIB LUIS VALLEJOS M.

EJERCICIO 1 – PRUEBAS



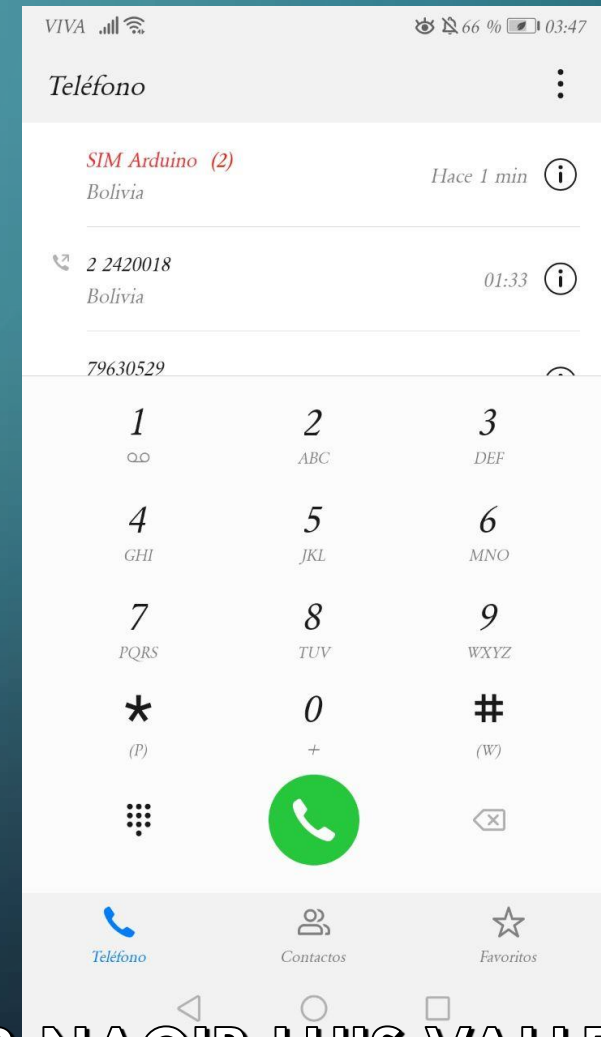
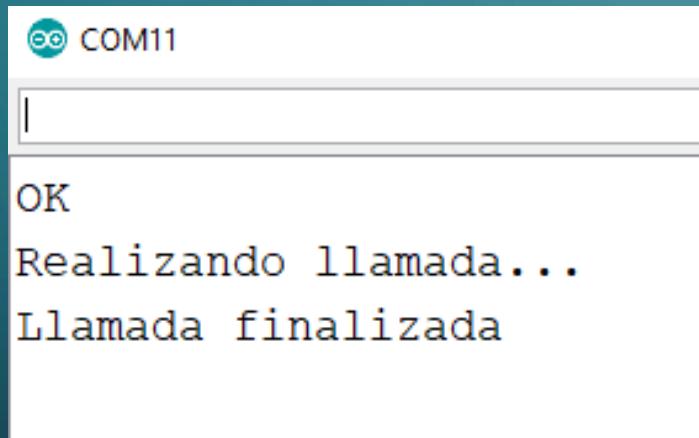
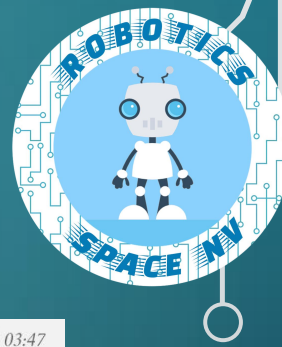
Si presionamos S1, realiza una llamada y se enciende un led mientras ésta esté activa, si presionamos S2, envía un SMS con los datos censados a un número telefónico.



TUTOR: NAGIB LUIS VALLEJOS M.

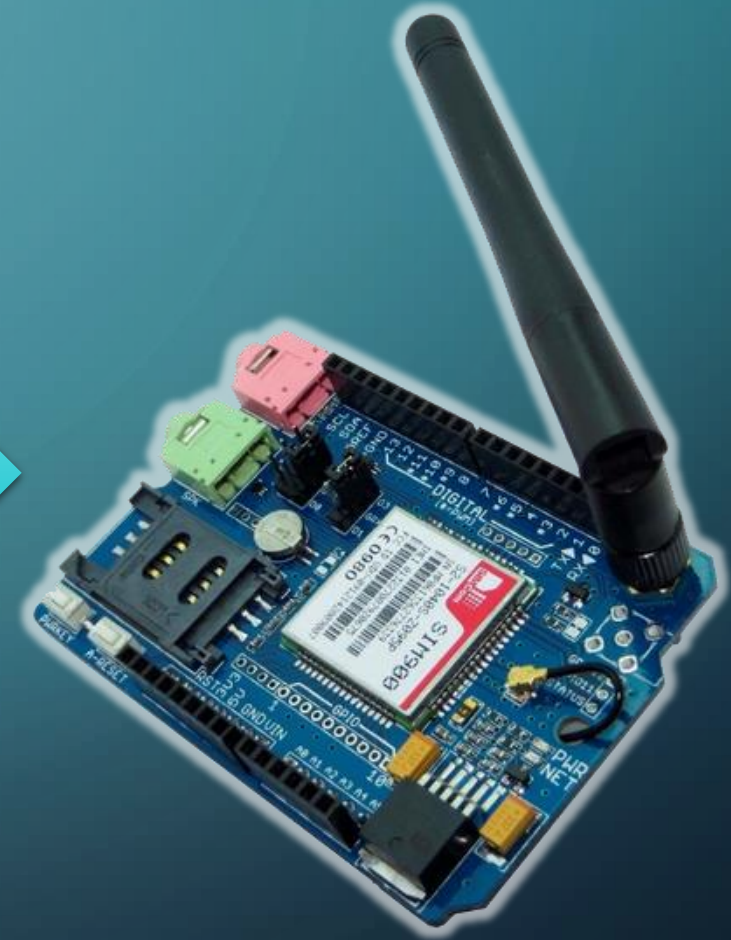
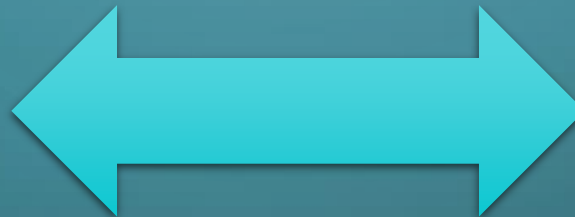
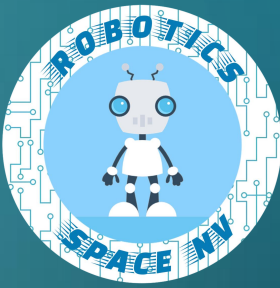
EJERCICIO 2 – PRUEBAS

Realizar una llamada después de presionar un pulsador.



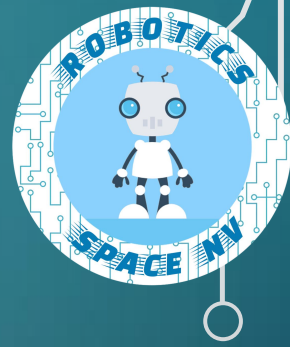
TUTOR:NAGIB LUIS VALLEJOS M.

GPRS + BLYNK



TUTOR:NAGIB LUIS VALLEJOS M.

GPRS/BLYNK



Blynk nos permite diferentes tipos de conexión para poder realizar sistemas y aplicativos IoT.

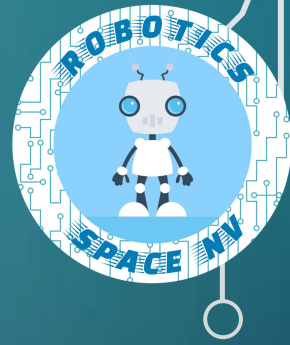
Entre estos tipos tenemos:

- Ethernet
- Wi-Fi
- USB
- GSM
- Bluetooth
- BLE



TUTOR: NAGIB LUIS VALLEJOS M.

GPRS/BLYNK



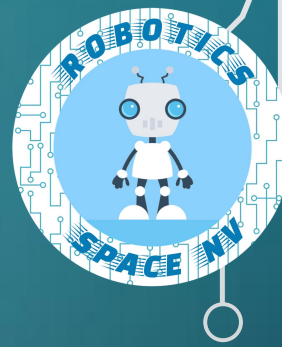
Nos enfocaremos en la conexión GSM. Este tipo de conexión permite que nuestros sistemas y/o aplicaciones puedan conectarse a Internet a través de la tecnología GPRS, la cual esta integrada en los módulos SIM(800,808,900,etc).

El beneficio conectarlo a través de GSM, permite que podamos desarrollar aplicaciones portátiles, es decir que pueda estar en funcionamiento si transportamos dicha aplicación a cualquier lugar del mundo.



TUTOR:NAGIB LUIS VALLEJOS M.

BLYNK – CREACIÓN DEL PROYECTO



Iniciamos un
nuevo
proyecto



Asignamos
nombre

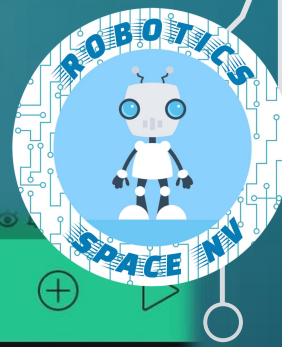
Asignamos
placa de
desarrollo

Ingresamos el
tipo de conexión:
GSM

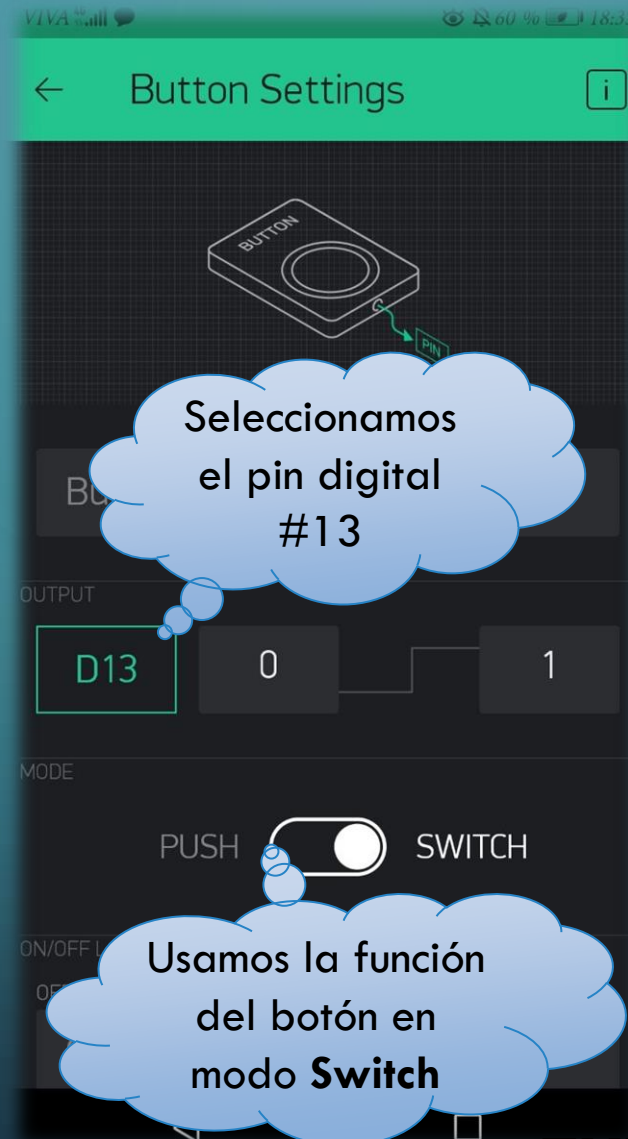
Creamos el
proyecto y
copiamos el token

TUTOR: NAGIB LUIS VALLEJOS M.

BLYNK – CREACIÓN DEL PROYECTO

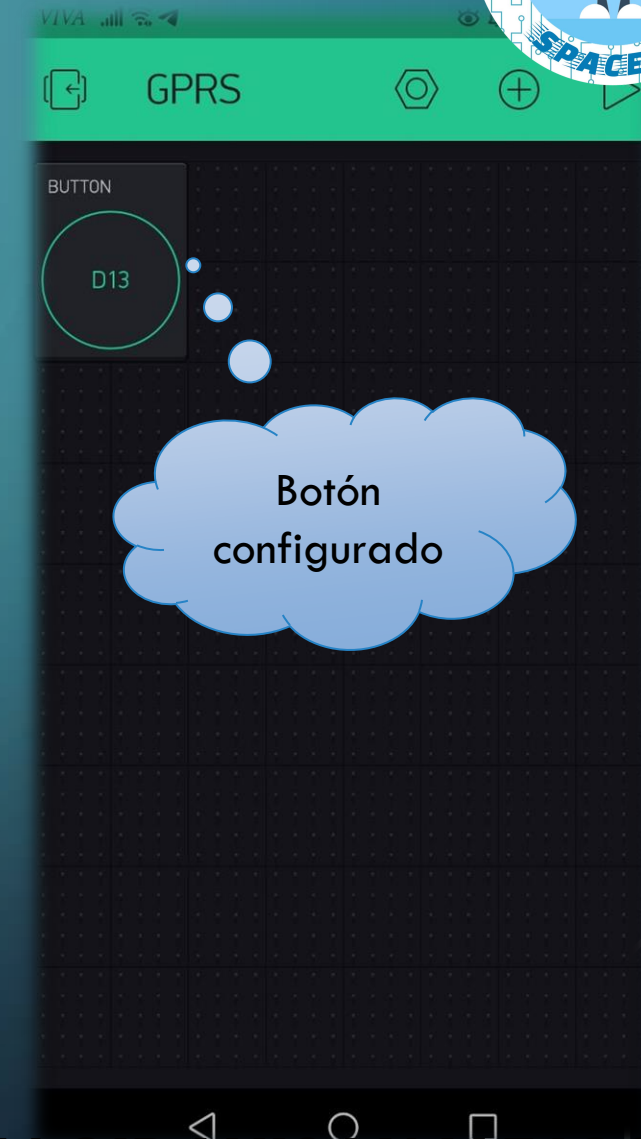


Agregamos un Widget de tipo **botón**



Seleccionamos el pin digital #13

Usamos la función del botón en modo **Switch**

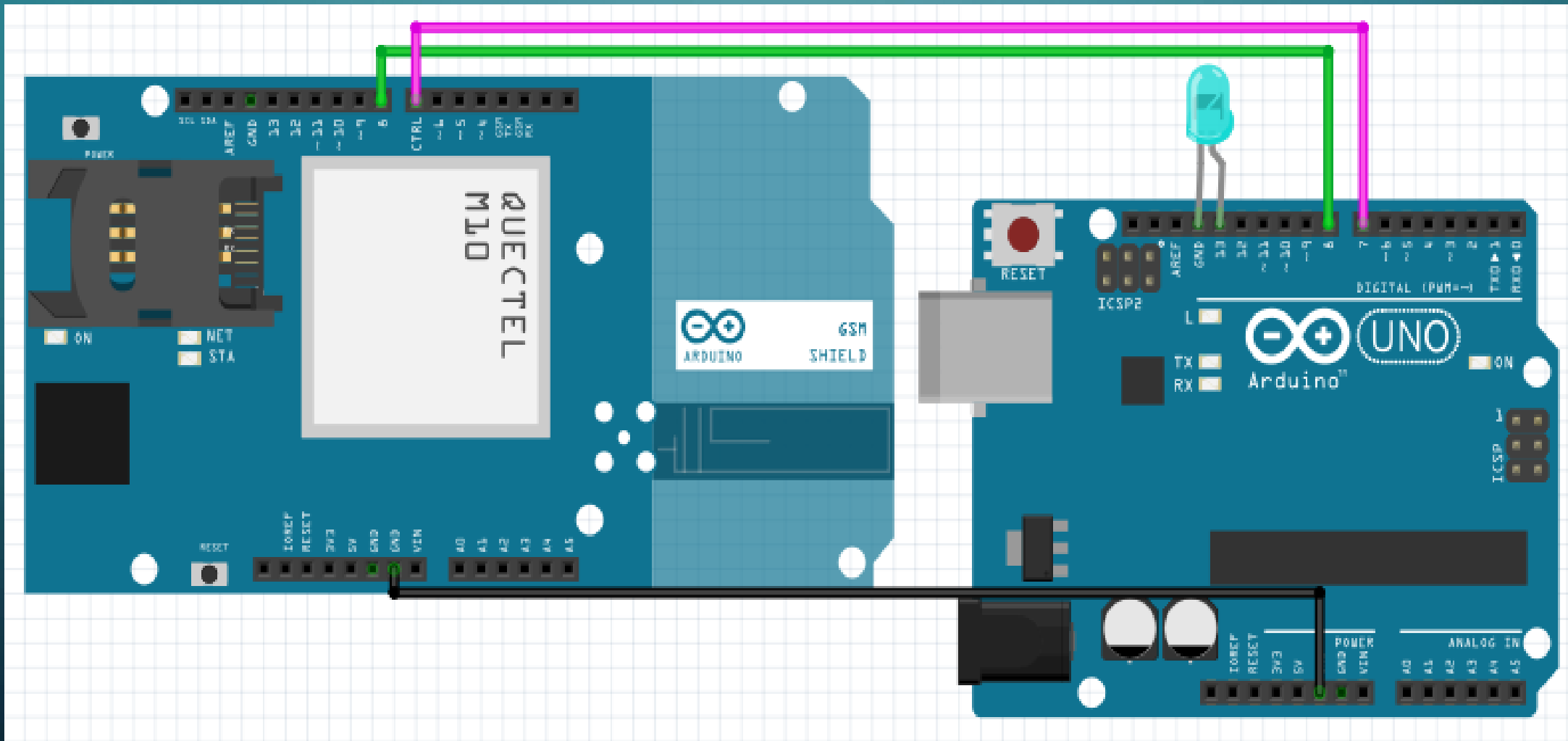
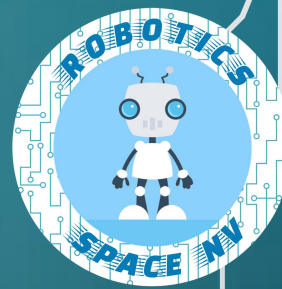


Botón configurado

TUTOR:NAGIB LUIS VALLEJOS M.

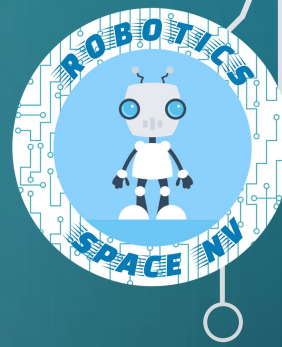
EJERCICIO 2 – CIRCUITO

Realizar el encendido y apagado de un led mediante GPRS



TUTOR:NAGIB LUIS VALLEJOS M.

ANEXO – APN DE LAS EMPRESAS TELEFÓNICAS BOLIVIANAS



APN: 4g.entel
Usuario: " "
Contraseña: " "



APN: internet.nuevatel.com
Usuario: " "
Contraseña: " "

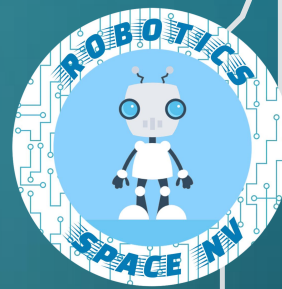


APN: internet.tigo.bo
Usuario: " "
Contraseña: " "

TUTOR: NAGIB LUIS VALLEJOS M.

EJERCICIO 2 – CÓDIGO

Realizar el encendido y apagado de un led mediante GPRS



S2-E2

```
1 #define BLYNK_PRINT Serial
2 // Dependiendo el modelo del sim ejecutamos
3 // #define TINY_GSM_MODEM_SIM800
4 #define TINY_GSM_MODEM_SIM900
5 // #define TINY_GSM_MODEM_M590
6 // #define TINY_GSM_MODEM_A6
7 // #define TINY_GSM_MODEM_A7
8 // #define TINY_GSM_MODEM_BG96
9 // #define TINY_GSM_MODEM_XBEE
10
11 #include <TinyGsmClient.h>
12 #include <BlynkSimpleTinyGSM.h>
13 #include <SoftwareSerial.h>
14
15 SoftwareSerial SerialAT(7,8); // RX, TX
16 TinyGsm modem(SerialAT);
17
```

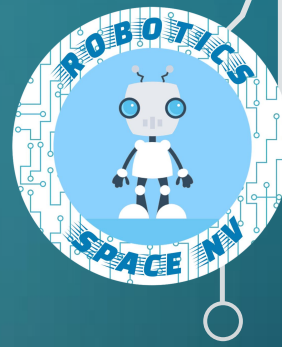
```
18 char auth[] = "MWGKcdVVhpEsWhBF2XNYsIN2FLH9p6Sw";
19 char apn[] = "4g.entel";
20 char user[] = "";
21 char pass[] = "";
22 void setup() {
23     Serial.begin(19200);
24     delay(10);
25     SerialAT.begin(19200);
26     delay(3000);
27     Serial.println("Iniciando el modem...");
28     modem.restart();
29     Blynk.begin(auth, modem, apn, user, pass);
30 }
31
32 void loop() {
33     Blynk.run();
34 }
```

Token generado
al crear el
proyecto en Blynk

TUTOR: NAGIB LUIS VALLEJOS M.

EJERCICIO 2 – CONEXIÓN

Realizar el encendido y apagado de un led mediante GPRS



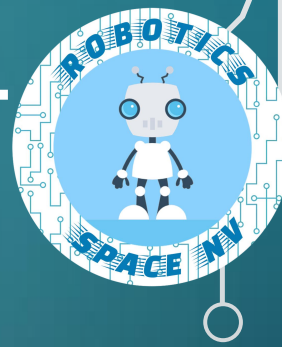
```
COM11
|
|
Iniciando el modem...
[14429]
  _ _ _ _ _
 / _ \ / _ \ / _ \
/_ _ \/_ _ \/_ _ \
      / _ \ v0.6.1 on Arduino Uno

[14476] Modem init...
[15056] Connecting to network...
[16212] Network: BOMOV
[16212] Connecting to 4g.entel ...
[21744] Connected to GPRS
[21882] Connecting to blynk-cloud.com:80
[24792] Ready (ping: 808ms).
```

Después de conectarse al GPRS, recién ejecutamos el proyecto en la app

TUTOR: NAGIB LUIS VALLEJOS M.

EJERCICIO 2 – ERROR DE CONEXIÓN



Realizar el encendido y apagado de un led mediante GPRS

```
COM12

Iniciando el modem...
[13253]

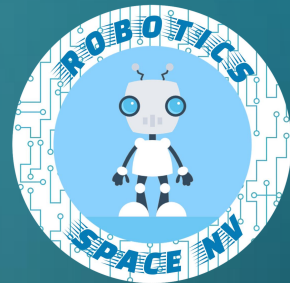
  _ _ _ _ _
 / _ \ / _ \ / _ \ / _ \
/_ _ \/_ _ \/_ _ \/_ _ \
      / _ \ v0.6.1 on Arduino Uno

[13300] Modem init...
[23558] Cannot init
Iniciando el modem...
[13254]

  _ _ _ _ _
 / _ \ / _ \ / _ \ / _ \
/_ _ \/_ _ \/_ _ \/_ _ \
      / _ \ v0.6.1 on Arduino Uno

[13301] Modem init...
[23560] Cannot init
Iniciando el modem...
```

TUTOR:NAGIB LUIS VALLEJOS M.

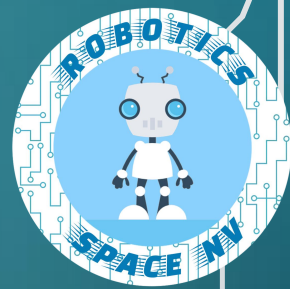


GPRS + THINGSPEAK



TUTOR:NAGIB LUIS VALLEJOS M.

THINGSPEAK



ThingSpeak es una plataforma para IoT, permite almacenar y recopilar datos de objetos conectados a través del protocolo HTTP a través de Internet o de una red local.

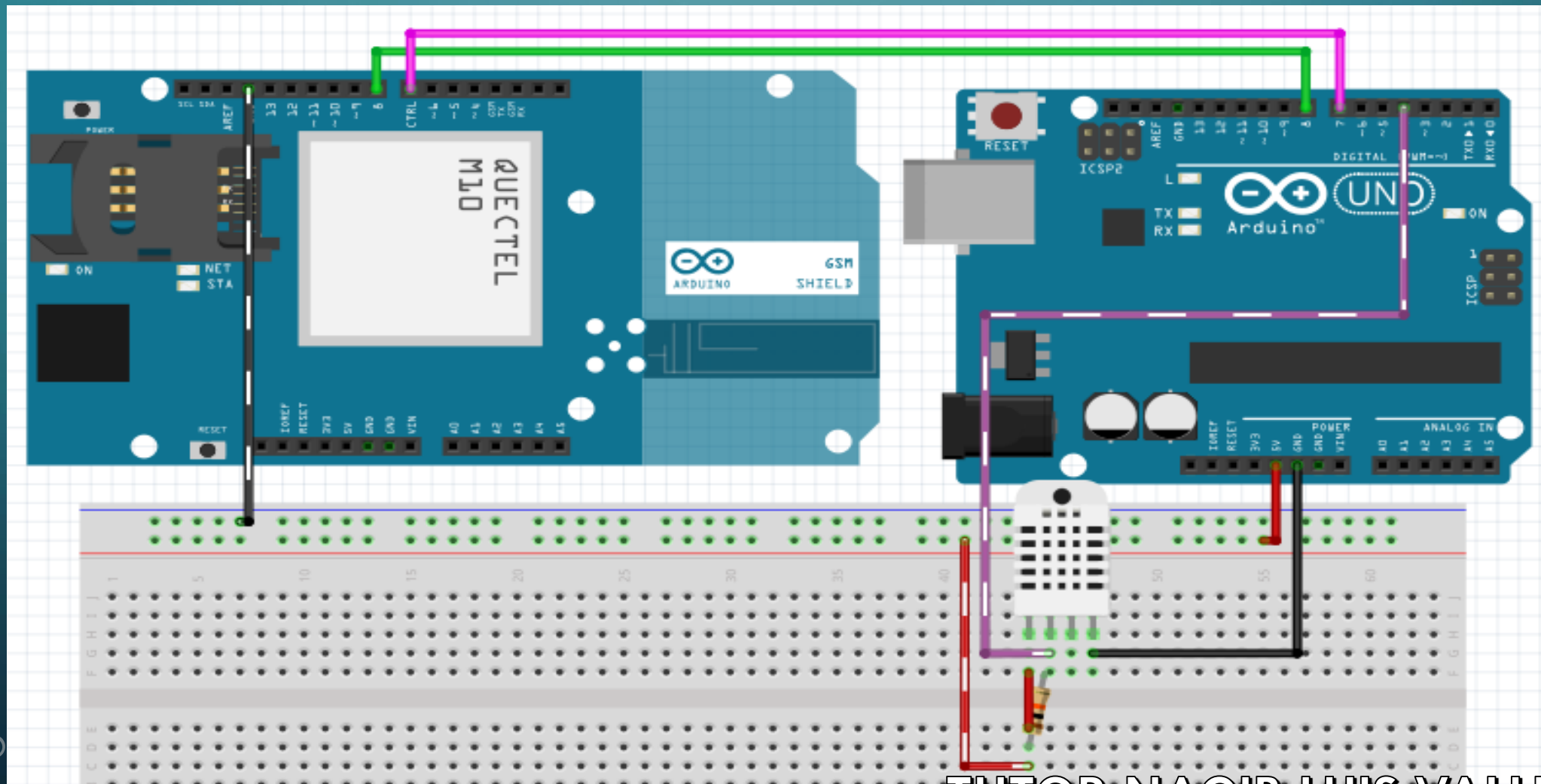
Con ThingSpeak, el usuario puede crear aplicaciones de registro de datos de sensores, aplicaciones de seguimiento de ubicación y una red social para los objetos conectados y actualizaciones de estado. Entre algunas funciones tenemos: Recolección de datos en tiempo real, datos de geolocalización, Procesamiento y visualización de datos.



ThingSpeak

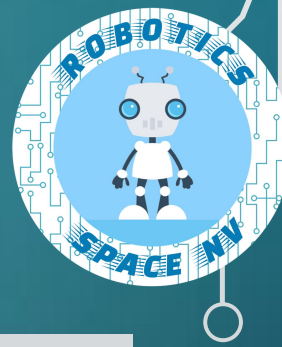
TUTOR: NAGIB LUIS VALLEJOS M.

Graficar la temperatura actual a través de Thingspeak

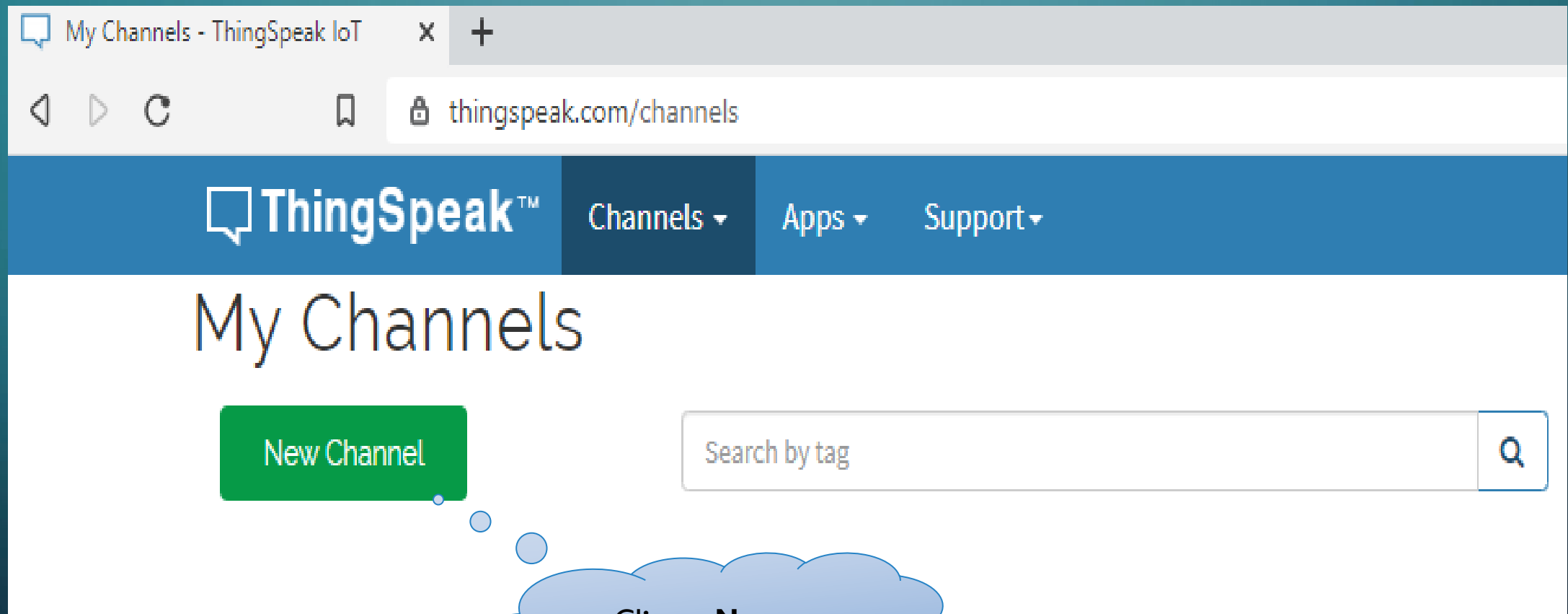


TUTOR: NAGIB LUIS VALLEJOS M.

EJERCICIO 3 – THINGSPEAK



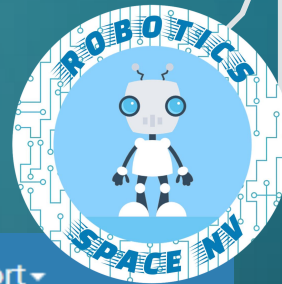
Login en **thingspeak.com** y creamos un nuevo canal:



Clic en **New Channel**

TUTOR:NAGIB LUIS VALLEJOS M.

EJERCICIO 3 – THINGSPEAK



ThingSpeak™ Canales ▾ Aplicaciones ▾ Apoyo ▾

Nuevo canal

Nombre

Descripción

Campo 1 ☒

Campo 2 ☐

Campo 3 ☐

Clic en **Save Channel**

ThingSpeak™ Channels ▾ Apps ▾ Support ▾

Link to GitHub

Elevation

Show Channel Location ☐

Latitude

Longitude

Show Video ☐
☒ YouTube
☐ Vimeo

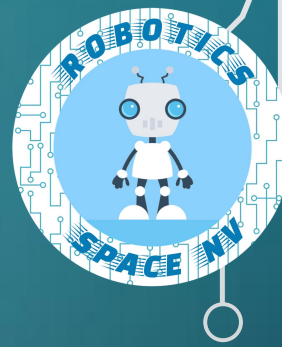
Video URL

Show Status ☐

☐ ☐ ☒ **Save Channel**

TUTOR:NAGIB LUIS VALLEJOS M.

EJERCICIO 3 – THINGSPEAK



ThingSpeak™ Canales ▾ Aplicaciones ▾ Apoyo ▾ [Uso comercial](#) [Cómo comprar](#) NV

Temperatura DHT22

ID de canal: **1042545** | Gráfica de datos del sensor DHT
Autor: [nagibvallejos](#)
Acceso: Público

Vista privada Vista pública Configuración del canal Compartir **Claves DE API** Importación / Exportación de Datos

[+ Añadir visualizaciones](#) [+ Añadir widgets](#) [Análisis de MATLAB](#) [Visualización de MATLAB](#)

[Exportar datos recientes](#)

Estadísticas del canal

Creado: [about 4 hours ago](#)
Ultima entrada: [hace unas 3 horas](#)
Entradas: 83

Gráfico de Campo 1

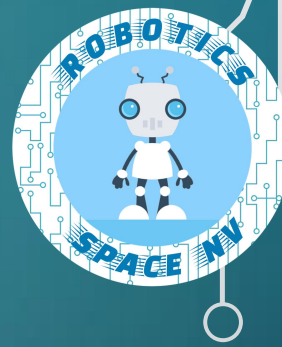
Temperatura DHT22

ThingSpeak.com

Canal creado:
Ahora clic en **API
Keys**

TUTOR:NAGIB LUIS VALLEJOS M.

EJERCICIO 3 – THINGSPEAK



ThingSpeak™ Canales ▾ Aplicaciones ▾ Apoyo ▾

Temperatura DHT22

ID de canal: **1042545** | Gráfica de datos del sensor DHT

Autor: [nagibvallejos](#)

Acceso: Público

[Vista privada](#) [Vista pública](#) [Configuración del canal](#) [Compartir](#) [Claves DE API](#)

Escribir clave de API

Clave

[Generar nueva clave de API de escritura](#)

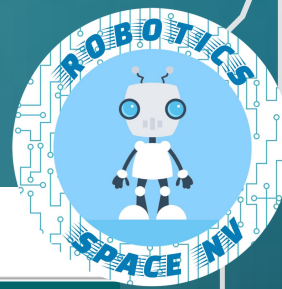
Ayuda

Las claves de API se dividen en dos tipos: público y privado. Las

Copiamos el **Write API Key**, en
mi caso:
P6A2S1TSVJSVSC1G

TUTOR:NAGIB LUIS VALLEJOS M.

EJERCICIO 3 – CÓDIGO



S2-E3 Arduino 1.8.12

Archivo Editar Programa Herramientas Ayuda

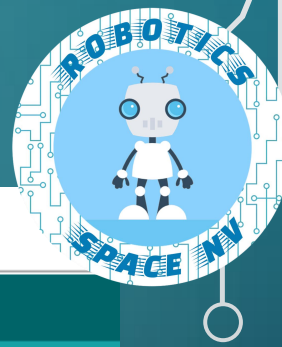


S2-E3

```
1 #include <SoftwareSerial.h>
2 #include <DHT.h>
3 #include <String.h>
4 SoftwareSerial SIM900(7,8);
5 DHT dht(4,DHT22);
6 float temperatura;
7
8 void setup(){
9     SIM900.begin(19200);
10    Serial.begin(19200);
11    dht.begin();
12    digitalWrite(9,1);
13    delay(1000);
14    digitalWrite(9,0);
15    delay(20000);
16 }
17
18 void loop(){
19     comandos();
20     if(SIM900.available())
21         Serial.write(SIM900.read());
22 }
23 void comandos(){
24     SIM900.println("AT+CIPSTATUS");
25     delay(2000);
26     SIM900.println("AT+CIPMUX=0");
27     delay(3000);
28     mostrarDatosSeriales();
29     SIM900.println("AT+CSTT=\"4g.entel\", \"\", \"\"");
30     delay(1000);
31     mostrarDatosSeriales();
32     SIM900.println("AT+CIICR");
33     delay(3000);
34     mostrarDatosSeriales();
35     SIM900.println("AT+CIFSR");
36     delay(2000);
37     mostrarDatosSeriales();
38     SIM900.println("AT+CIPSPRT=0");
39     sensor();
40     delay(3000);
41     mostrarDatosSeriales();
42     SIM900.println("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", \"80\"");
43     delay(6000);
44     mostrarDatosSeriales();
45     SIM900.println("AT+CIPSEN
```

TUTOR: NAGIB LUIS VALLEJOS M.

EJERCICIO 3 – CÓDIGO



S2-E3 Arduino 1.8.12

Archivo Editar Programa Herramientas Ayuda

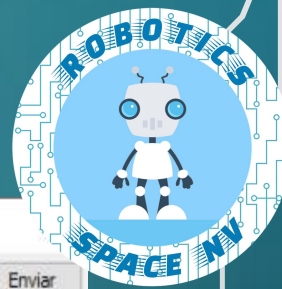


S2-E3

```
44  mostrarDatosSeriales();
45  SIM900.println("AT+CIPSEND");
46  delay(4000);
47  mostrarDatosSeriales();
48  String datos="GET https://api.thingspeak.com/update?api_key=P6A2S1TSVJSVSC1G&field1=0" + String(temperatura);
49  SIM900.println(datos);
50  delay(4000);
51  mostrarDatosSeriales();
52  SIM900.println((char)26);
53  delay(5000);
54  SIM900.println();
55  mostrarDatosSeriales();
56  SIM900.println("AT+CIPSHUT");
57  delay(5000);
58  mostrarDatosSeriales();
59  }
60  void mostrarDatosSeriales(){
61    while(SIM900.available() !=0)
62      Serial.write(SIM900.read());
63  }
64  void sensor(){
65    temperatura = dht.readTemperature();
66    Serial.print("La temperatura es: ");
67    Serial.print(temperatura);
68    Serial.println("°C");
69  }
```

TUTOR:NAGIB LUIS VALLEJOS M.

EJERCICIO 3 – PRUEBAS



Temperatura DHT22 - ThingSpeak

← → ↻ 🔒 https://

Thi

Estadísticas del canal

Creado: Hace 2 minutos

Entradas: 1

Gráfico de Campo 1

Temperatura DHT22

DHT22

19:58:45.000

Fecha

ThingSpeak.com

COM12

Enviar

ERROR

ERROR

*PSNWID: "736","02", "Entel S.A Bolivia", 0, "Entel"La temper

OK

OK

CONNECT OK

SEND OK

1

SHUT OK

OK

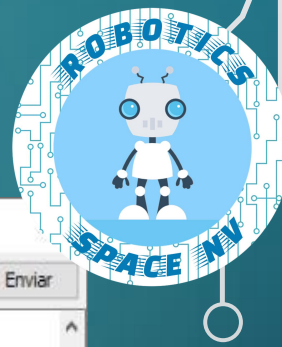
STATE: IP INITIAL

OK

*Primer dato de
llegada al servidor*

Primer dato enviado

EJERCICIO 3 – PRUEBAS



Quinto dato de llegada al servidor

Entradas: 5

Gráfico de Campo 1

Temperatura DHT22

ThingSpeak.com

SEND OK
5

Quinto dato enviado

COM12

Enviar

SHUT OK

OK

STATE: IP INITIAL

OK

OK

OK

OK

10.155.82.43

La temperatura es: 19.60°C

OK

OK

CONNECT OK

TUTOR:NAGIB LUIS VALLEJOS M.

CONTACTOS

Suscríbete



(+591) 63096640



robotics.space.nv@gmail.com



fb.me/RoboticsSpaceNV



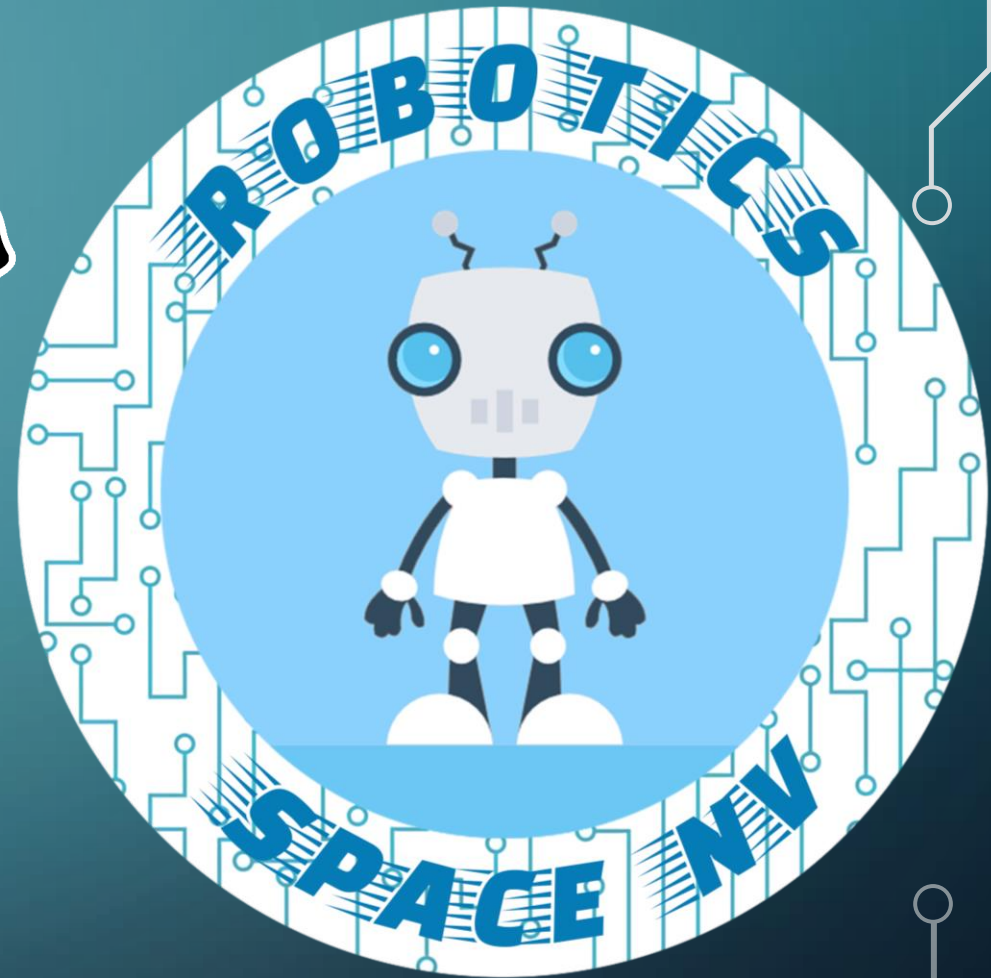
@NagibVallejos



Robotics Space NV



<https://github.com/nagibvalejos/Robotics-Space-NV>



TUTOR:NAGIB LUIS VALLEJOS M.