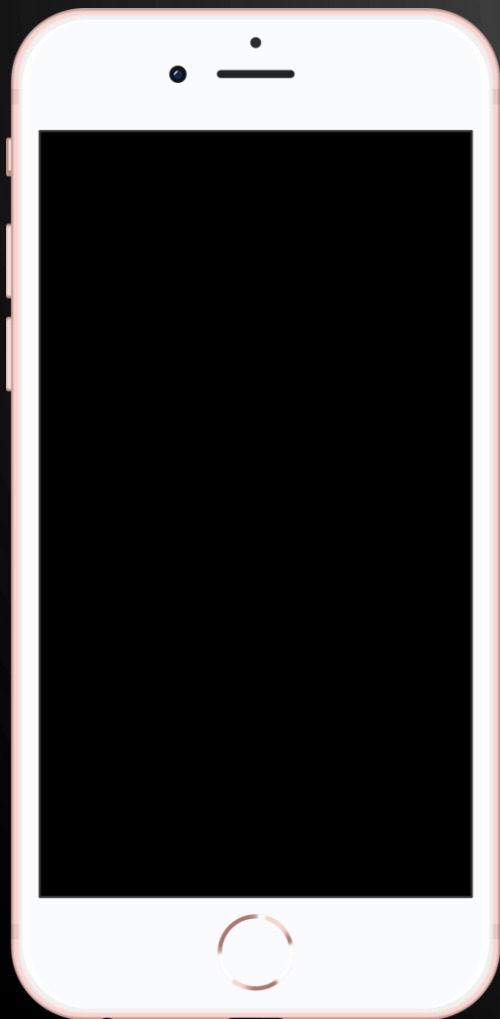
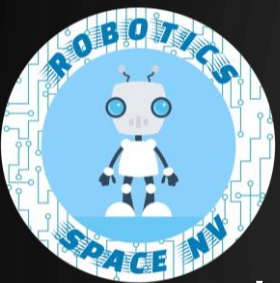


COMUNIDAD ARDUINO OPEN SOURCE

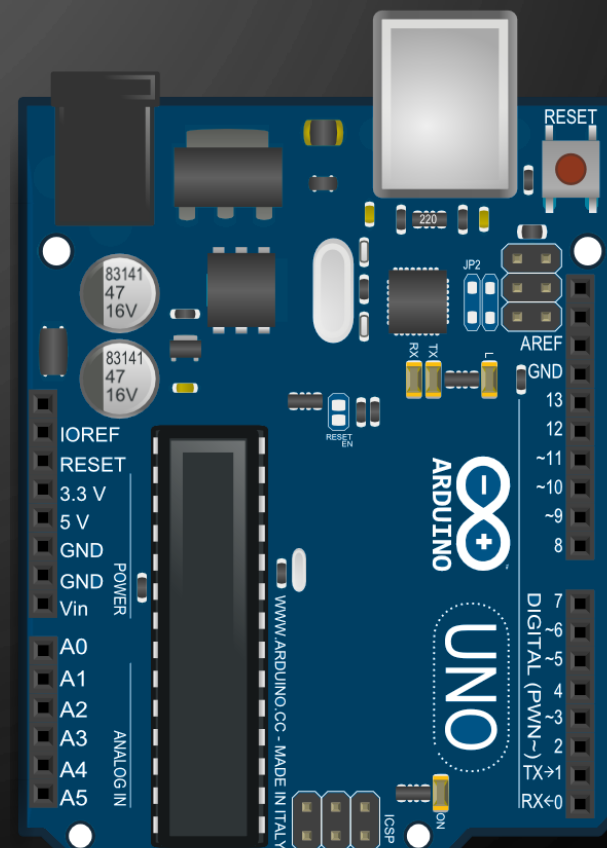
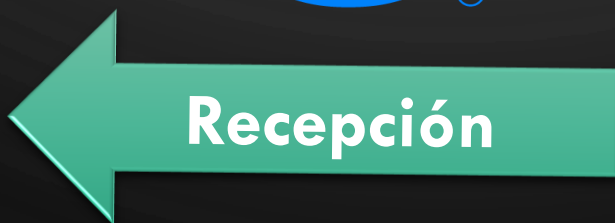
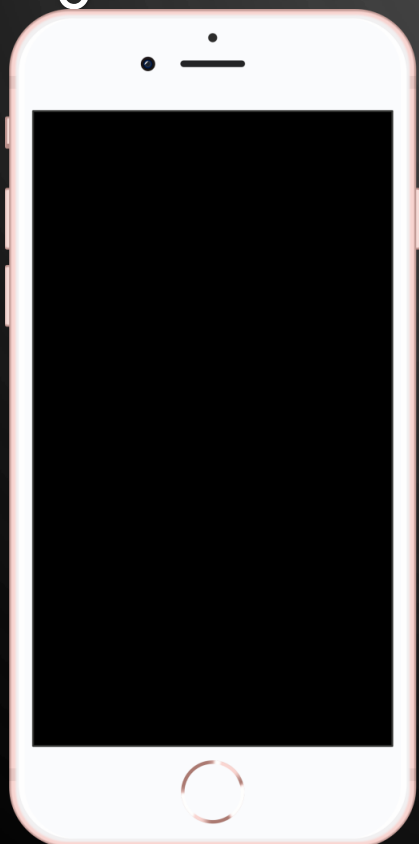
ARDUINO III CLASE 1



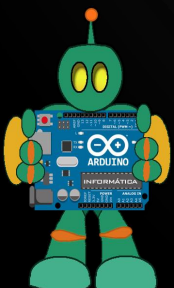


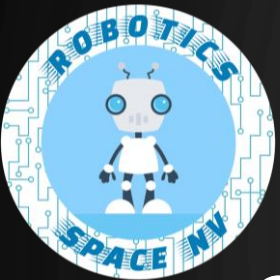
COMUNICACIÓN BIDIRECCIONAL

La comunicación bidireccional se refiere al envío y recepción de datos, en la placa arduino tenemos los pines digitales 0 y 1 configurados como tx/rx por defecto



TUTOR: NAGIB LUIS VALLEJOS M.





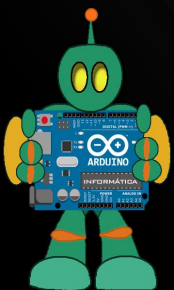
FUNCIÓN SPRINTF

Consiste en realizar el envío de datos de manera automática.

La función sprintf pertenece al lenguaje C, la cual nos permite crear una cadena multiformato, mediante la cual se puede enviar cadenas, caracteres, números enteros y con decimales.



TUTOR: NAGIB LUIS VALLEJOS M.





FUNCIÓN SPRINTF

Su sintaxis es la siguiente:

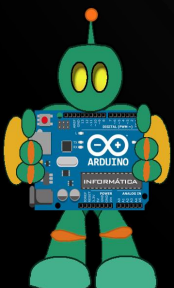
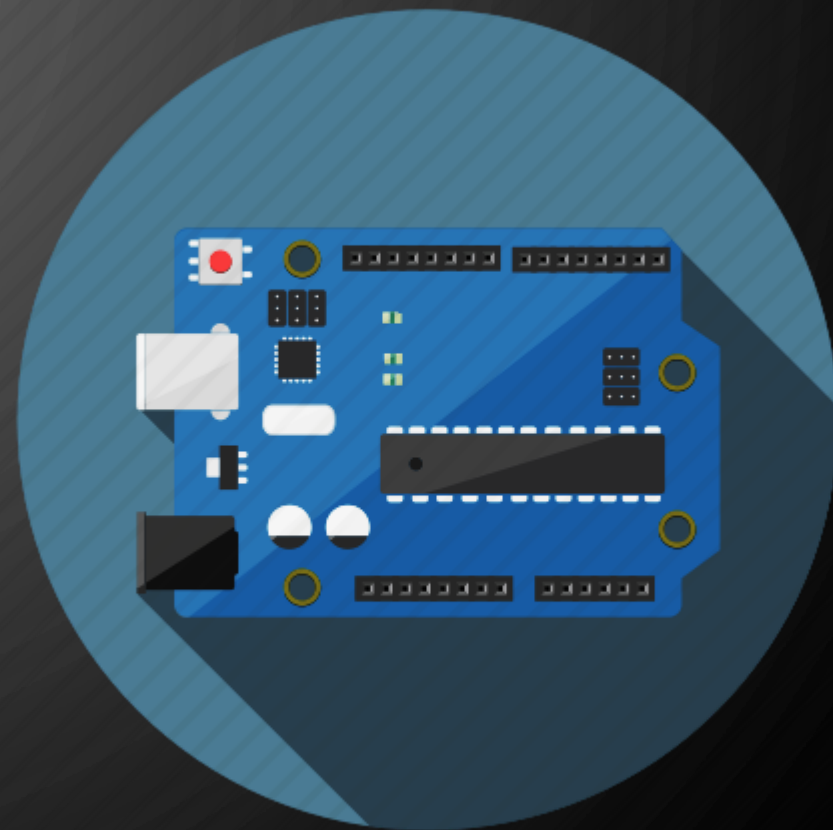
sprintf(Str, Tipo_dato, Parámetro);

Donde:

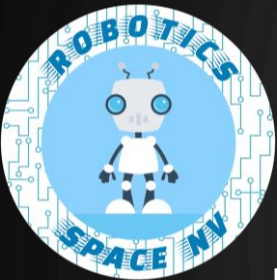
Str= Se refiere al vector de almacenamiento.

Tipo_dato= El tipo dato a ser almacenado (decimal, cadena, carácter, flotante).

Parámetro= El valor del dato a almacenar en el vector.



TUTOR: NAGIB LUIS VALLEJOS M.

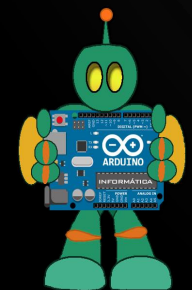


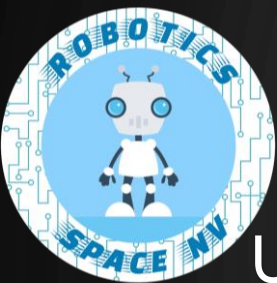
TIPOS DE FORMATOS

La función admite los siguientes formatos:

Tipo variable	Valor	Formato	Descripción	Salida
char*	"ModuloIII"	%s	Cadena	ModuloIII
char	'x'	%c	Carácter	x
char	65	%c	Carácter	A
int	65	%d	Entero	65
int	65	%8d	Anchura fija (8c)	_____65
int	65	%07d	Anchura fija(Prefijo 0)	0000065
int	65	%#o	Octal	0101
int	65	%x	Hexadecimal	41
Int	65	%#x	Hexadecimal	0x41

TUTOR: NAGIB LUIS VALLEJOS M.

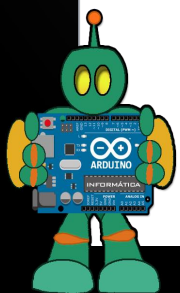




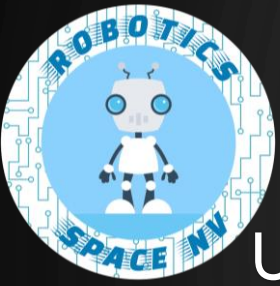
EJERCICIO 1

Uso de diferentes formatos.

```
COM9
Cadena:ModuloIII, Entero:65, Entero8: 65, Entero0:0000065, Octal:0101, CharacterE:A, Character:a
Cadena:ModuloIII, Entero:65, Entero8: 65, Entero0:0000065, Octal:0101, CharacterE:A, Character:a
Cadena:ModuloIII, Entero:65, Entero8: 65, Entero0:0000065, Octal:0101, CharacterE:A, Character:a
Cadena:ModuloIII, Entero:65, Entero8: 65, Entero0:0000065, Octal:0101, CharacterE:A, Character:a
Cadena:ModuloIII, Entero:65, Entero8: 65, Entero0:0000065, Octal:0101, CharacterE:A, Character:a
Cadena:ModuloIII, Entero:65, Entero8: 65, Entero0:0000065, Octal:0101, CharacterE:A, Character:a
Cadena:ModuloIII, Entero:65, Entero8: 65, Entero0:0000065, Octal:0101, CharacterE:A, Character:a
Cadena:ModuloIII, Entero:65, Entero8: 65, Entero0:0000065, Octal:0101, CharacterE:A, Character:a
```



TUTOR: NAGIB LUIS VALLEJOS M.



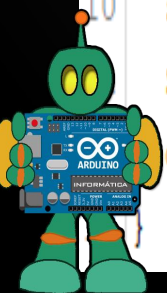
EJERCICIO 1

Uso de diferentes formatos.

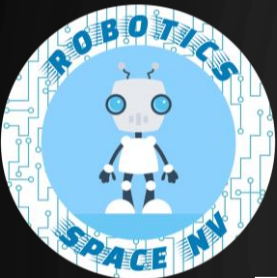
S1-E1

```
1 char datos[50];
2 void setup() {
3   Serial.begin(9600);
4 }
5
6 void loop() {
7   char* cadena="ModuloIII";
8   int num=65;
9   char car='a';
10  sprintf(datos,"Cadena:%s, Entero:%d, Entero8:%8d, Entero0:%07d, Octal:%#o, CaracterE:%c, Caracter:%c",cadena,num,num,num,num,num,car);
    Serial.println(datos);
    delay(5000);
}
```

Recomendación:
Enviar solo números a la app
para su desarrollo
Aunque depende de tipo de
formato en el envío de
información



TUTOR: NAGIB LUIS VALLEJOS M.



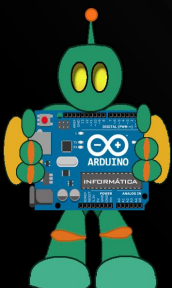
APP INVENTOR 2

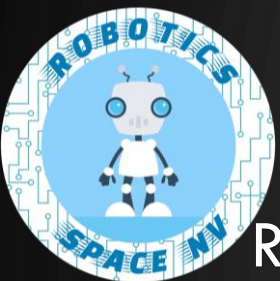
Es una herramienta online que nos permite crear aplicaciones móviles en base a bloques, creada por Google en 2011 y cedida a mit quien se encargo del soporte y desarrollo, la página de trabajo es:

ai2.appinventor.mit.edu



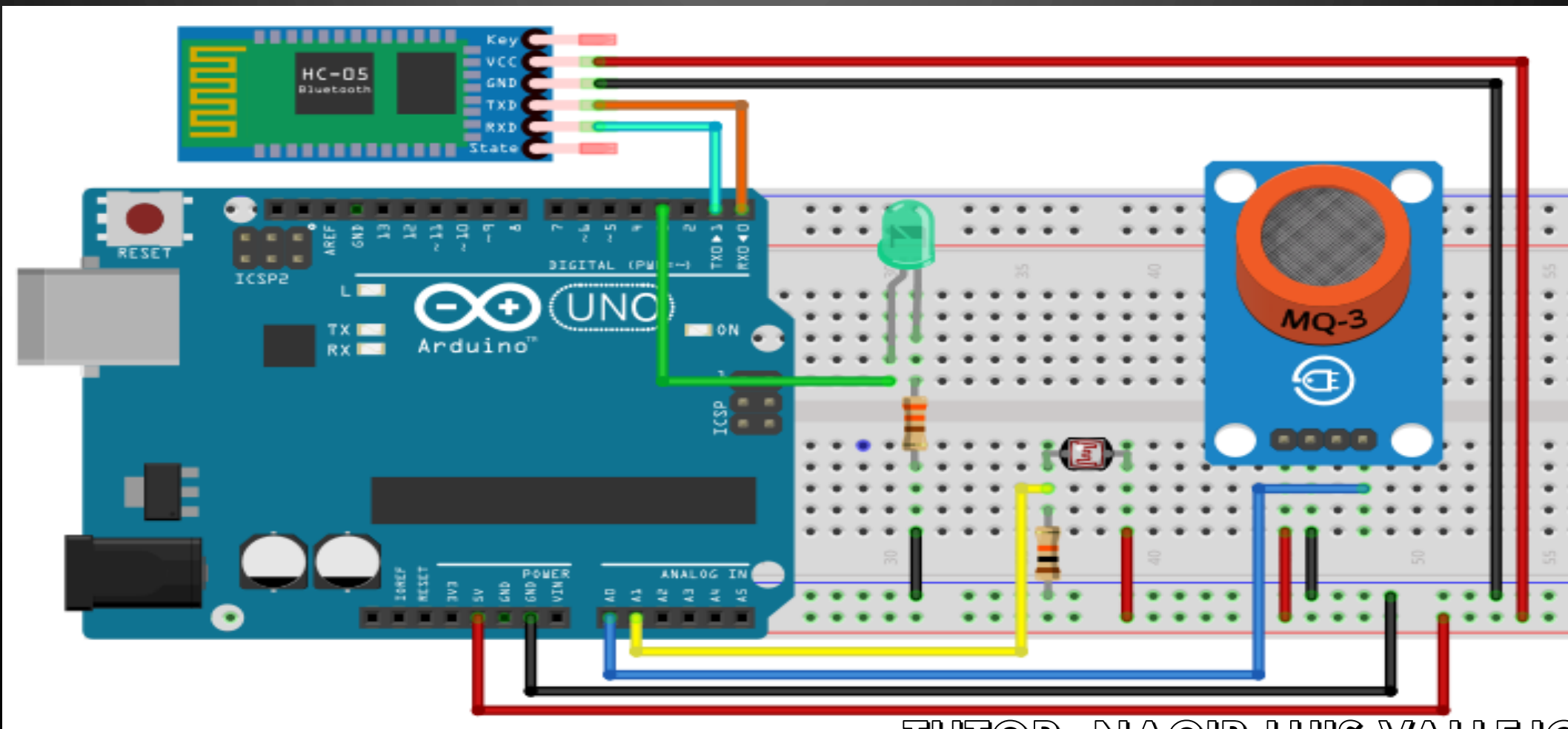
TUTOR: NAGIB LUIS VALLEJOS M.



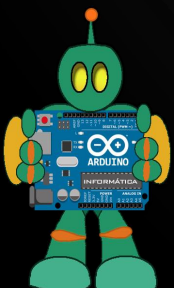


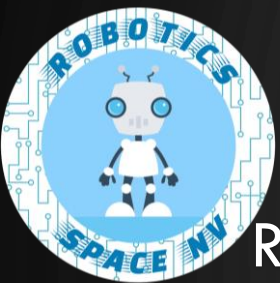
EJERCICIO 2 - CIRCUITO

Realizar el encendido y apagado de un led y también realizar visualizar los datos de 2 sensores analógicos



TUTOR: NAGIB LUIS VALLEJOS M.





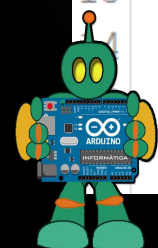
EJERCICIO 2 - SOLUCIÓN

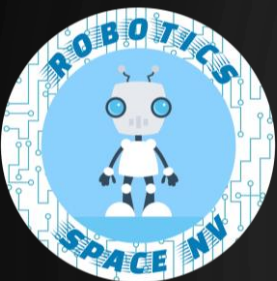
Realizar el encendido y apagado de un led y también realizar visualizar los datos de 2 sensores analógicos

S1-E2

```
1 int verde=3,LDR=A0,MQ=A1,lecturaldr,lecturamq;
2 char datos[20];
3 char aux=' ';
4 void setup() {
5     pinMode(verde,OUTPUT);
6     Serial.begin(9600);
7 }
8
9 void loop() {
10     leer_datos(); //recepcion de datos
11     lecturaldr=analogRead(LDR); //envio de datos
12     lecturamq=analogRead(MQ);
13     int porldr=map(lecturaldr,0,1023,0,100);
14     int pormq=map(lecturamq,0,1023,0,100);
15     sprintf(datos,"%d,%d,%d,%d",lecturamq,lecturaldr,porldr,lecturamq);
16     Serial.println(datos);
17     delay(1000);
18 }
19
20 void leer_datos() {
21     if(Serial.available()>0) {
22         aux=Serial.read();
23     }
24     if(aux=='A') {
25         digitalWrite(verde,1);
26     }
27     else if(aux=='B') {
28         digitalWrite(verde,0);
29     }
30 }
```

TUTOR: NAGIB LUIS VALLEJOS M.





EJERCICIO 2 - DISEÑO APP

Elemento tipo Selector de lista

Seleccionar dispositivo

Elemento tipo Etiqueta

- LUCES -

Dormitorio:



Elemento tipo Botón

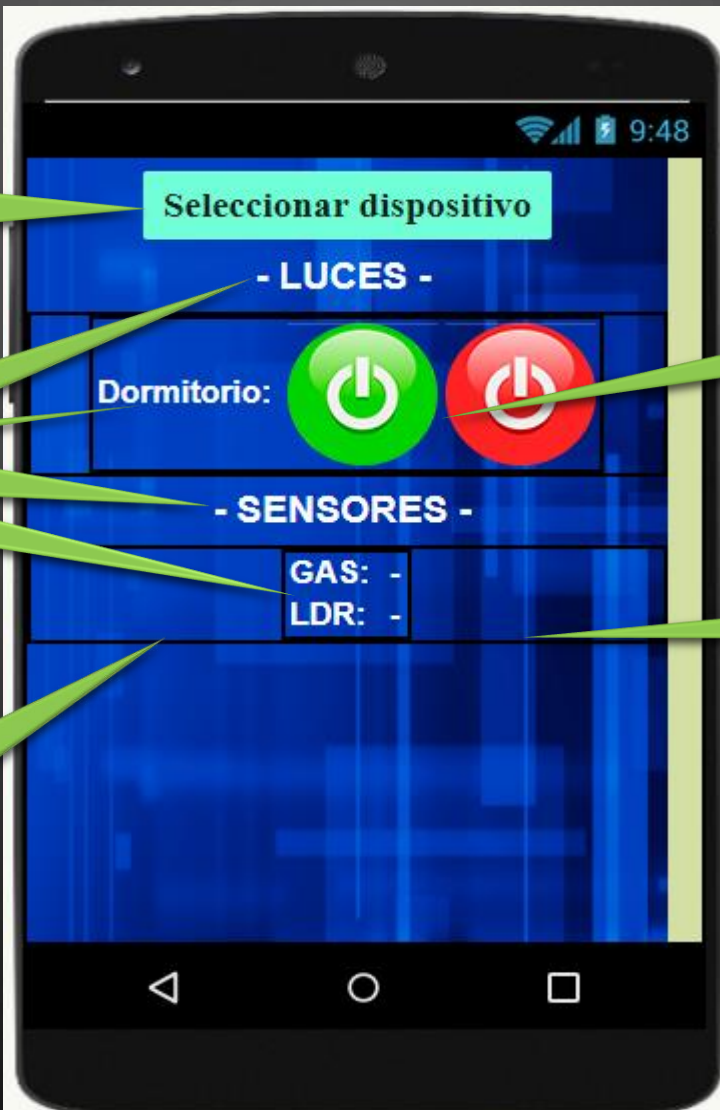
- SENSORES -

GAS: -

LDR: -

Elemento tipo Disposición horizontal

Elemento tipo Disposición tabular



Componentes no visibles



Bluetooth

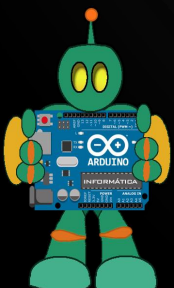


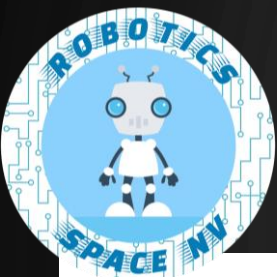
Reloj1



Reloj2

TUTOR: NAGIB LUIS VALLEJOS M.





EJERCICIO 2 - PROGRAMACIÓN

cuando Seleccionar ▾ .AntesDeSelección

ejecutar poner Seleccionar ▾ . Elementos ▾ como Bluetooth ▾ . DireccionesYNombres ▾

cuando Seleccionar ▾ .DespuésDeSelección

ejecutar poner Seleccionar ▾ . Selección ▾ como llamar Bluetooth ▾ .Conectar
dirección Seleccionar ▾ . Selección ▾

poner Reloj1 ▾ . TemporizadorHabilitado ▾ como cierto ▾

poner Reloj2 ▾ . TemporizadorHabilitado ▾ como cierto ▾

cuando Encender ▾ .Clic

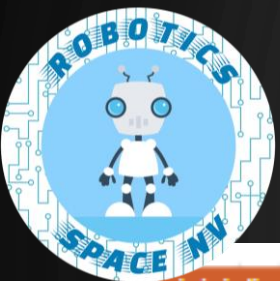
ejecutar llamar Bluetooth ▾ .EnviarTexto
texto " A "

cuando Apagar ▾ .Clic

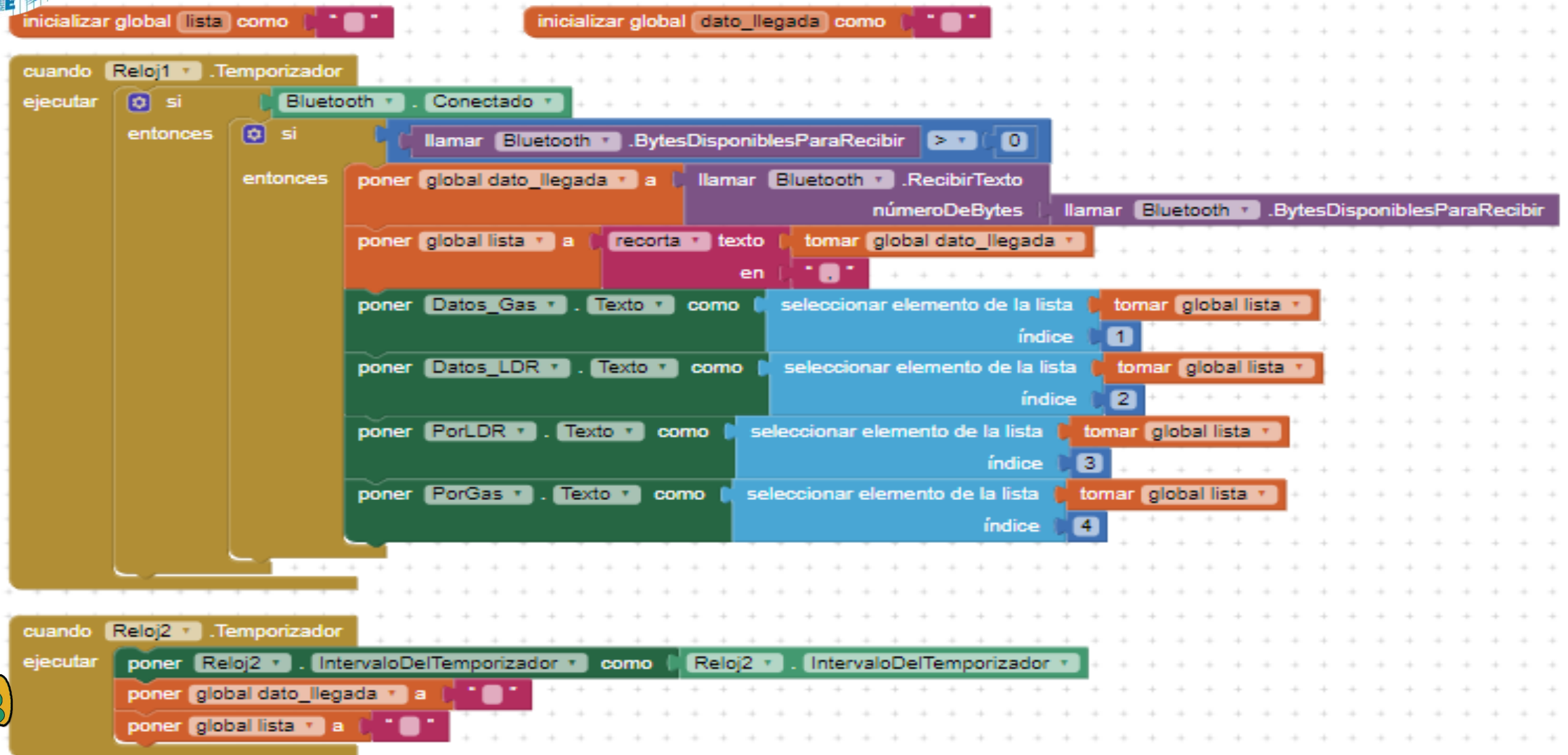
ejecutar llamar Bluetooth ▾ .EnviarTexto
texto " B "

TUTOR: NAGIB LUIS VALLEJOS M.

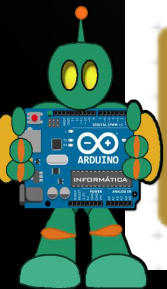




EJERCICIO 2 - PROGRAMACIÓN



TUTOR: NAGIB LUIS VALLEJOS M.



CONTACTOS



SUSCRÍBETE



(+591) 63096640



robotics.space.nv@gmail.com



fb.me/RoboticsSpaceNV



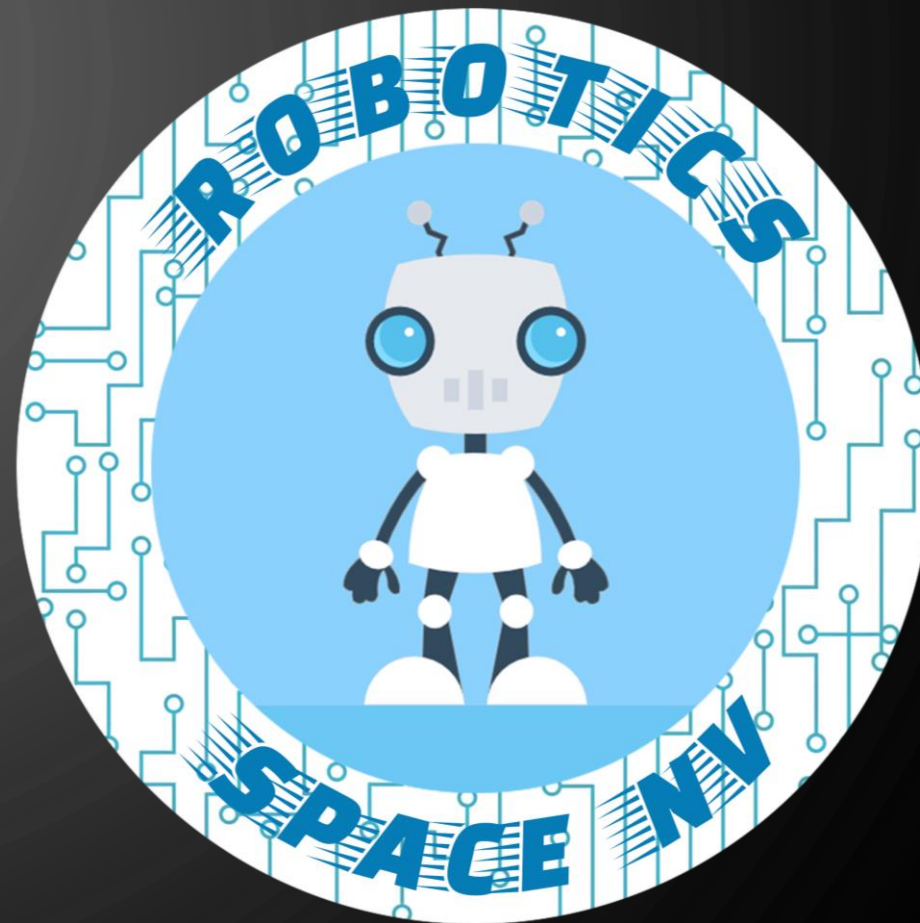
@NagibVallejos



Robotics Space NV



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



TUTOR: NAGIB LUIS VALLEJOS M.