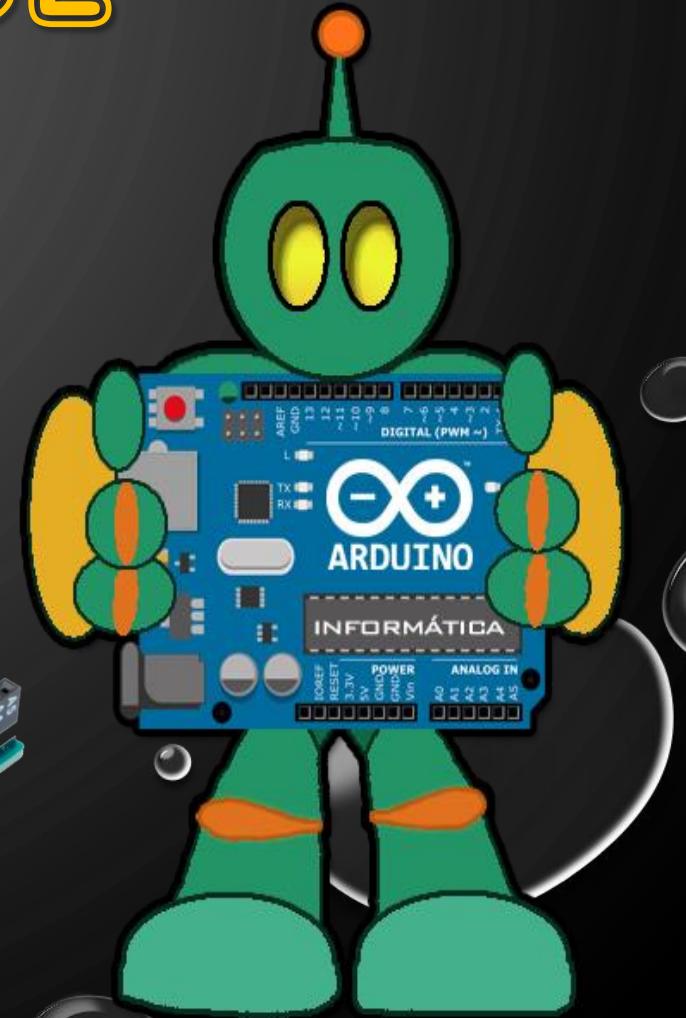
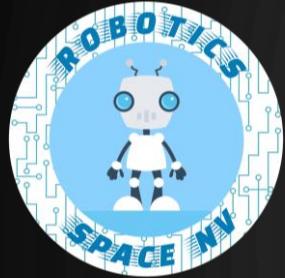


COMUNIDAD ARDUINO

OPEN SOURCE

ARDUINO III CLASE 4

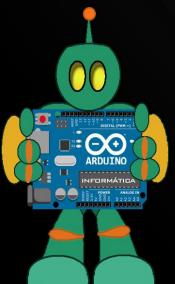




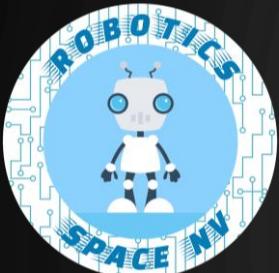
PROCESSING

Es un lenguaje de programación basado en java de código abierto y enfocado a la realización de entornos gráficos.

Processing cuenta con diferentes librerías que se encargan de integrar diferentes funciones como por ejemplo: animaciones en 2d, 3d, aplicaciones visuales, creación de imágenes entre otros.



TUTOR: NAGIB LUIS VALLEJOS M.



COMANDOS BÁSICOS

mousepressed(): Función invocada cada vez que se le da un click al ratón (mouse).

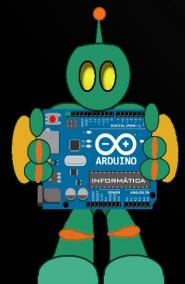
mousereleased(): Función invocada cada vez que se suelte el click del ratón (mouse).

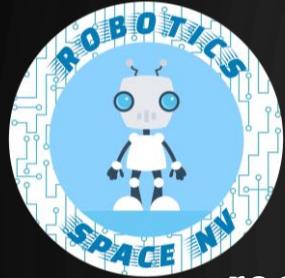
keypressed(): Función invocada cada vez que se pulsa un tecla (keyboard).

keyreleased(): Función invocada cada vez que se suelta una tecla (keyboard).

line(): Comando para crear una línea (puede ejecutarse sobre 2 o 3 ejes: x,y ó x,y,z).

point(): Comando para crear un punto en el programa (puede ejecutarse sobre 2 o 3 ejes).





COMANDOS BÁSICOS

rect(): Comando para crear un rectángulo

ej: **rect(a,b,c,d)**

triangle(): Comando para crear un triángulo con coordenadas en x e y

ej: **triangle(x1, y1, x2, y2, x3, y3)**

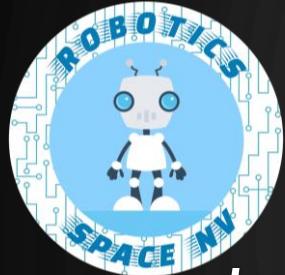
text(): Función para escribir un texto dentro del programa el cual puede ubicarse en los ejes x, y ó x,y,z

ej: **text("escrito", x, y)** ó **text("escrito", x, y, z)**

textFont(): Comando para colocar el tipo de fuente que va a usar el texto



TUTOR: NAGIB LUIS VALLEJOS M.



COMANDOS BÁSICOS

textsize(): Comando para establecer un tamaño al texto (medido en píxeles).

background(): Función para llenar de color el fondo del entorno gráfico.

fill(): Usado para llenar de color una figura.

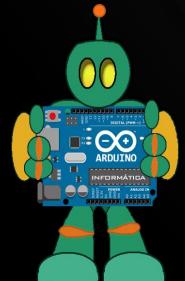
nofill(): Figura sin color.

stroke(): Darle color al borde de las imágenes.

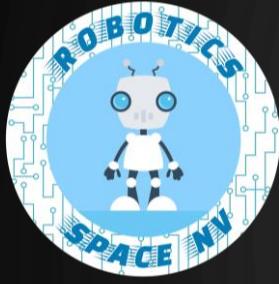
stroke(): Imagen o figura sin borde.



<http://processing.org/reference/>



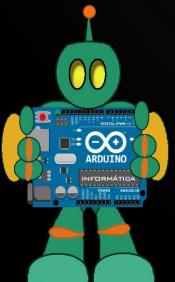
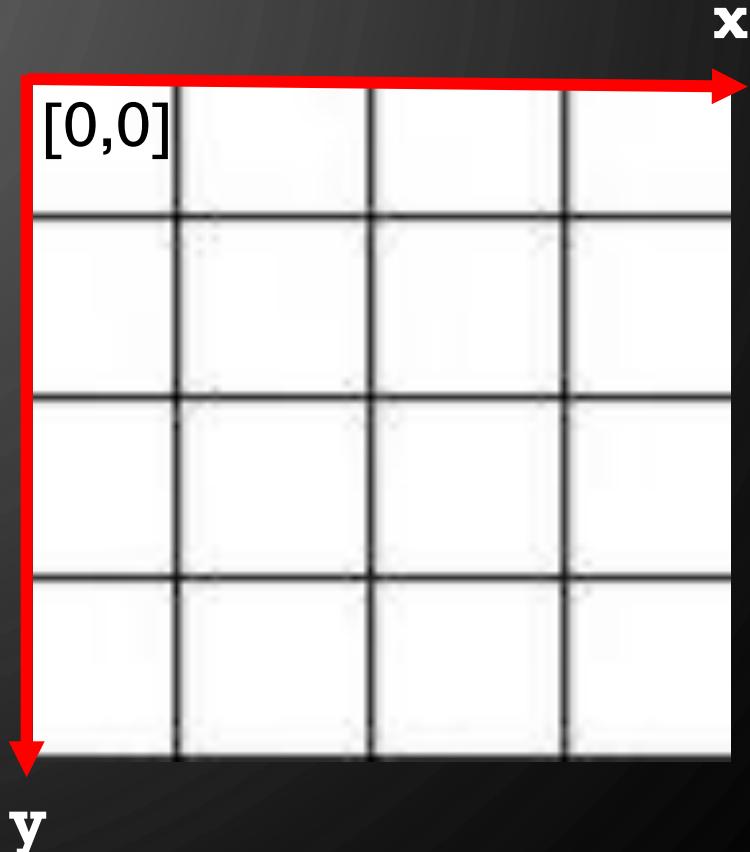
TUTOR: NAGIB LUIS VALLEJOS M.

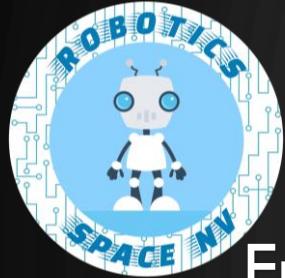


POSICIONAMIENTO DE PANTALLA

El posicionamiento de pantalla en Processing inicia en el extremo superior izquierdo.

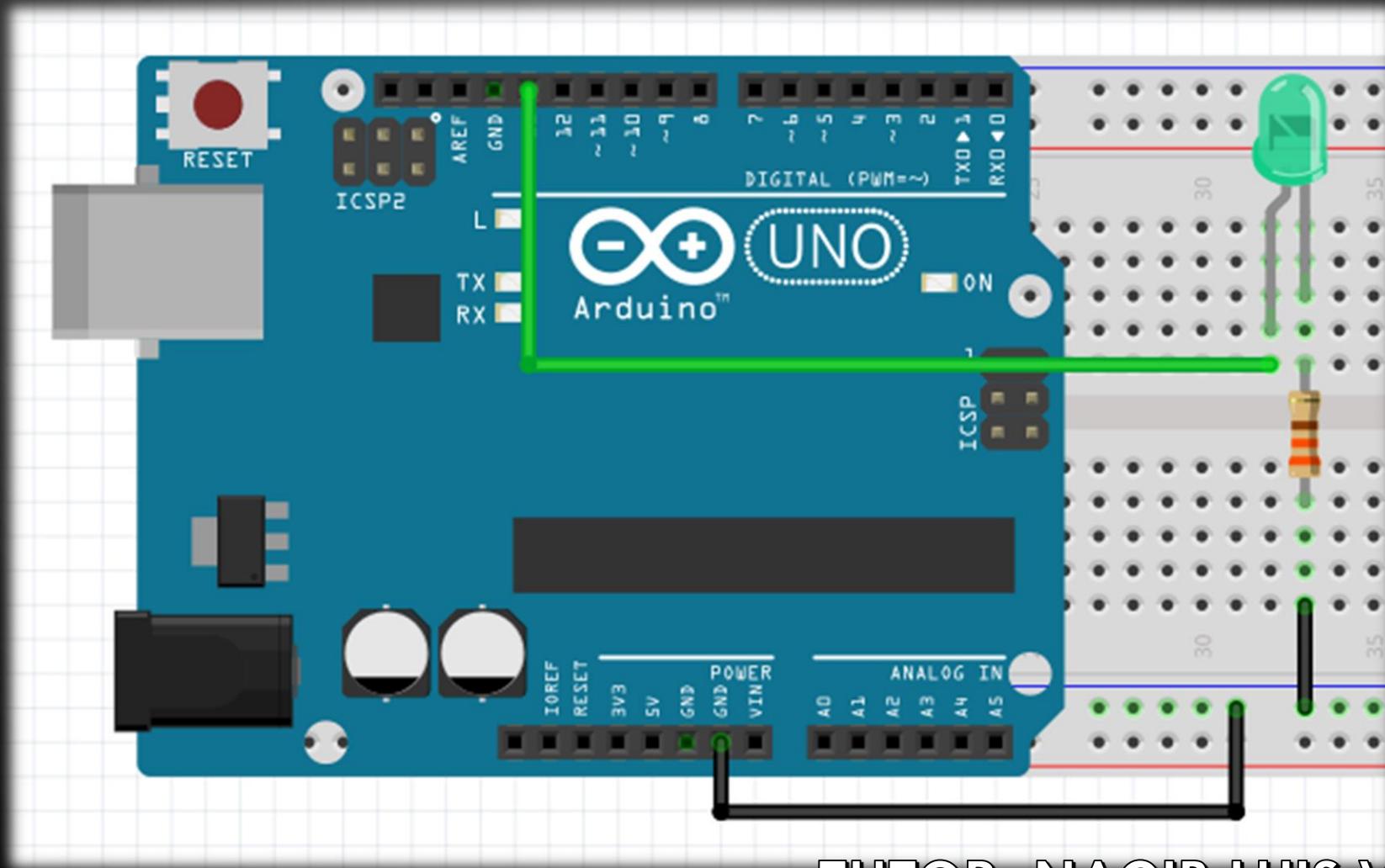
Para “x” el movimiento se realiza hacia la derecha y en el caso de “y” hacia abajo.



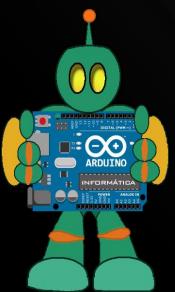


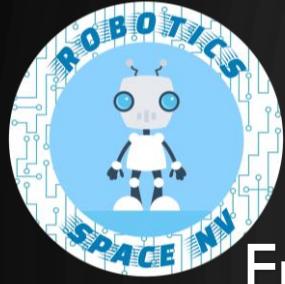
EJERCICIO 1 - CIRCUITO

Encender y apagar un LED con botones virtuales



TUTOR: NAGIB LUIS VALLEJOS M.





EJERCICIO 1 - SOL. ARDUINO

Encender y apagar un LED con botones virtuales

S4-E1

```
1 int led=13;
2 char valor=' ';
3 void setup() {
4     Serial.begin(9600);
5     pinMode(led, OUTPUT);
6     digitalWrite(led, 0);
7 }
8 void loop() {
9     if(Serial.available()>0) {
10         valor=Serial.read();
11         if(valor=='E') {
12             digitalWrite(led, 1);
13         }
14         else if(valor=='A') {
15             digitalWrite(led, 0);
16         }
17     }
18 }
```



TUTOR: NAGIB LUIS VALLEJOS M.

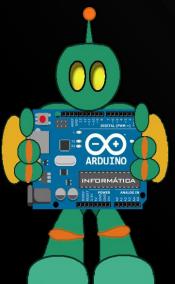


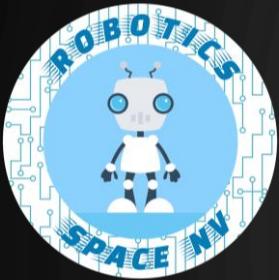
EJERCICIO 1 - SOL. PROCESSING

Blynk

```
1 import processing.serial.*; // Iniciamos la librería Serial
2 Serial puerto; // se declara una variable para la com. serial
3 void setup(){
4     puerto = new Serial(this,"COM101", 9600);
5     size(300,150); // se crea una ventana de 300px de ancho * 150 px
6 }
7
8 void draw(){
9     background(0,128,125); // fondo en formato RGB
10    fill(255,255,255); // color del texto
11    textSize(20); // tamaño del texto
12    textAlign(CENTER, TOP); // alineación del texto
13    text("Led interno del Arduino",150,25); // texto, posición X,Y
14    stroke(90); // línea bordeada
15    strokeWeight(2); // grosor de la línea bordeada
16    // ----- CUADRO 1 -----
17    fill(255,255,255); // relleno del cuadro 1
18    rect(50,60,80,40,10); // (A,B,C,D,E)
19    // A = Posición en X ~~ B = Posición en Y
20    // C = Ancho del rectángulo ~~ D = Alto del rectángulo
21    // E = Borde del rectángulo
22    fill(0,0,0); // color del texto = blanco
23    text("ON", 90, 70); // texto, posición X,Y
24    // ----- CUADRO 2 -----
25    fill(255,255,255); // relleno del cuadro 2
26    rect(175,60,80,40,10); // texto, posición X,Y
27    fill(0,0,0); // color del texto = blanco
28    text("OFF", 215, 70); // texto, posición X,Y
```

TUTOR: NAGIB LUIS VALLEJOS M.

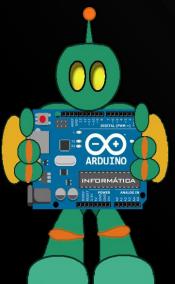




EJERCICIO 1 - SOL. PROCESSING

```
Blynk ▾  
29 }  
30 void mousePressed(){  
31 // evaluar si el botón de "ON" es pulsado  
32 // EjeX"ON"= 50(inicio del cuadro) + 80(largo del cuadro)= 130  
33 // EjeY"ON"= 60(inicio del cuadro) + 40(alto del cuadro)= 100  
34 // mouseX= coordinada en X del cursor  
35 // mouseY= coordinada en Y del cursor  
36 fill(60,60,60); // color del botón al ser presionado  
37 rect(50,60,80,40,10); // crear rectángulo 1  
38 puerto.write('E');  
39 // enviar la letra "E" por el puerto creado para que el  
40 // carácter sea recibido por el puerto serial del Arduino  
41 // y de esta forma encender el led.  
42 println("Led ENCENDIDO"); // imprimir en pantalla estado del  
43 }  
44 if((mouseX>175 & mouseX<255) & (mouseY>60 & mouseY<100)){  
45 // evaluar si el botón de "OFF" es pulsado  
46 // EjeX"OFF"= 175(inicio del cuadro) + 80(largo del cuadro)= 255  
47 // EjeY"OFF"= 60(inicio del cuadro) + 40(alto del cuadro)= 100  
48 // mouseX= coordinada en X del cursor  
49 // mouseY= coordinada en Y del cursor  
50 fill(60,60,60); // color del botón al ser presionado  
51 rect(175,60,80,40,10); // crear rectángulo 2  
52 puerto.write('A');  
53 println("Led APAGADO"); // imprimir en pantalla estado del  
54 }  
55 }  
56 }
```

TUTOR: NAGIB LUIS VALLEJOS M.





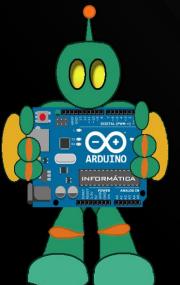
EJERCICIO 1 - PRUEBAS DE FUNCIONAMIENTO



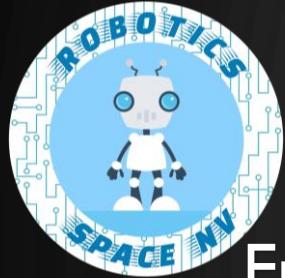
Led interno del Arduino

ON

OFF

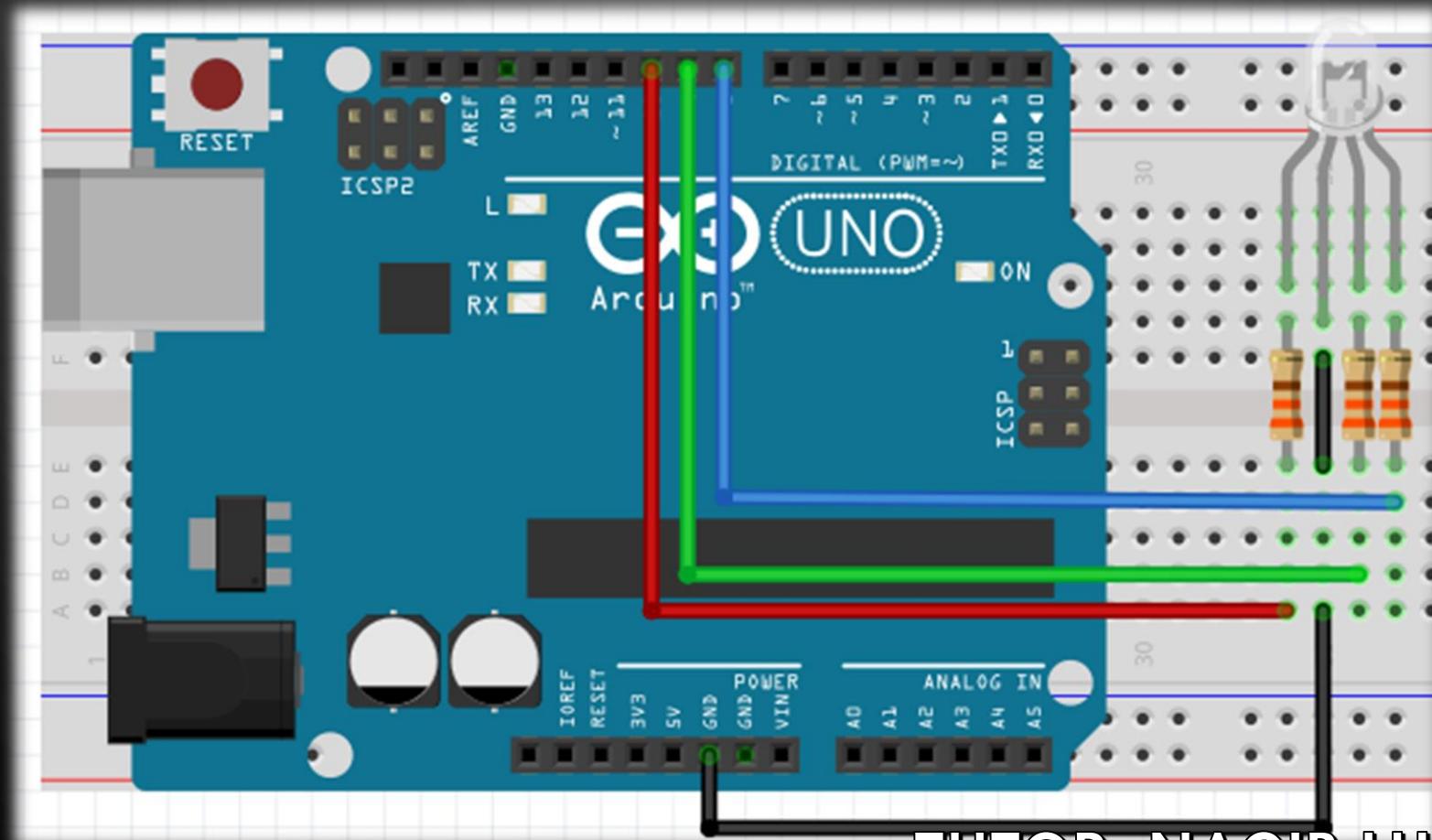


TUTOR: NAGIB LUIS VALLEJOS M.

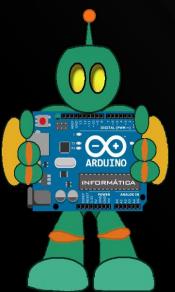


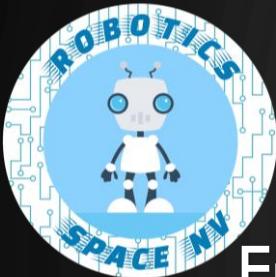
EJERCICIO 2 - CIRCUITO

Encender los colores primarios de un LED RGB y apagarlo con botones virtuales



TUTOR: NAGIB LUIS VALLEJOS M.





EJERCICIO 2 - SOL. ARDUINO

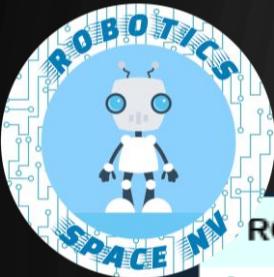
Encender los colores primarios de un LED RGB y apagarlo con botones virtuales

S4-E2

```
1 void setup() {                                13     else if(valor=='B') {  
2   Serial.begin(9600);                         14       ledRGB(0,0,1);  
3 }                                              15   }  
4 void loop() {                                 16     else if(valor=='A') {  
5   if(Serial.available()>0){                   17       ledRGB(0,0,0);  
6     valor=Serial.read();                      18   }  
7     if(valor=='R') {                          19 }  
8       ledRGB(1,0,0);                        20 }  
9   }                                              21  
10    else if(valor=='G') {                     22 void ledRGB(int r, int g, int b){  
11      ledRGB(0,1,0);                        23   digitalWrite(R,r);  
12    }                                         24   digitalWrite(G,g);  
13 }                                              25   digitalWrite(B,b);  
14 }                                              26 }
```

TUTOR: NAGIB LUIS VALLEJOS M.





EJERCICIO 2 - SOL. PROCESSING

RGB

```
1 import processing.serial.*;
2 Serial puerto;
3 int EstadoBoton=0;
4 color ColorBlanco=color(250,250,250);
5 color ColorRojo=color(200,0,0);
6 color ColorVerde=color(0,200,0);
7 color ColorAzul=color(0,0,200);
8 void setup(){
9     puerto= new Serial(this,"COM6",9600);
10    size(450,500);
11 }
12 void draw(){
13     background(0,0,0);
14     if(EstadoBoton==0){
15         fill(ColorBlanco);
16     }
17     else if(EstadoBoton==1){
18         fill(ColorRojo);
19     }
20     else if(EstadoBoton==2){
21         fill(ColorVerde);
22     }
23     else if(EstadoBoton==3){
24         fill(ColorAzul);
25     }
26     ellipse(220,300,300,300);
27     //CUADRO1
28     fill(255,255,255);
29     rect(50,60,80,40,10);
30     fill(0,0,0);
31     text("ROJO",73,85);
32     textSize(15);
33     //CUADRO2
34     fill(255,255,255);
35     rect(175,60,80,40,10);
36     fill(0,0,0);
37     text("VERDE",193,85);
38     textSize(15);
39     //CUADRO3
40     fill(255,255,255);
41     rect(300,60,80,40,10);
42     fill(0,0,0);
43     text("AZUL",323,85);
44     textSize(15);
45     //CUADRO4
46     fill(255,255,255);
47     rect(175,105,80,40,10);
48     fill(0,0,0);
49     text("OFF",200,130);
50     textSize(15);
51 }
```

TUTOR: NAGIB LUIS VALLEJOS M.

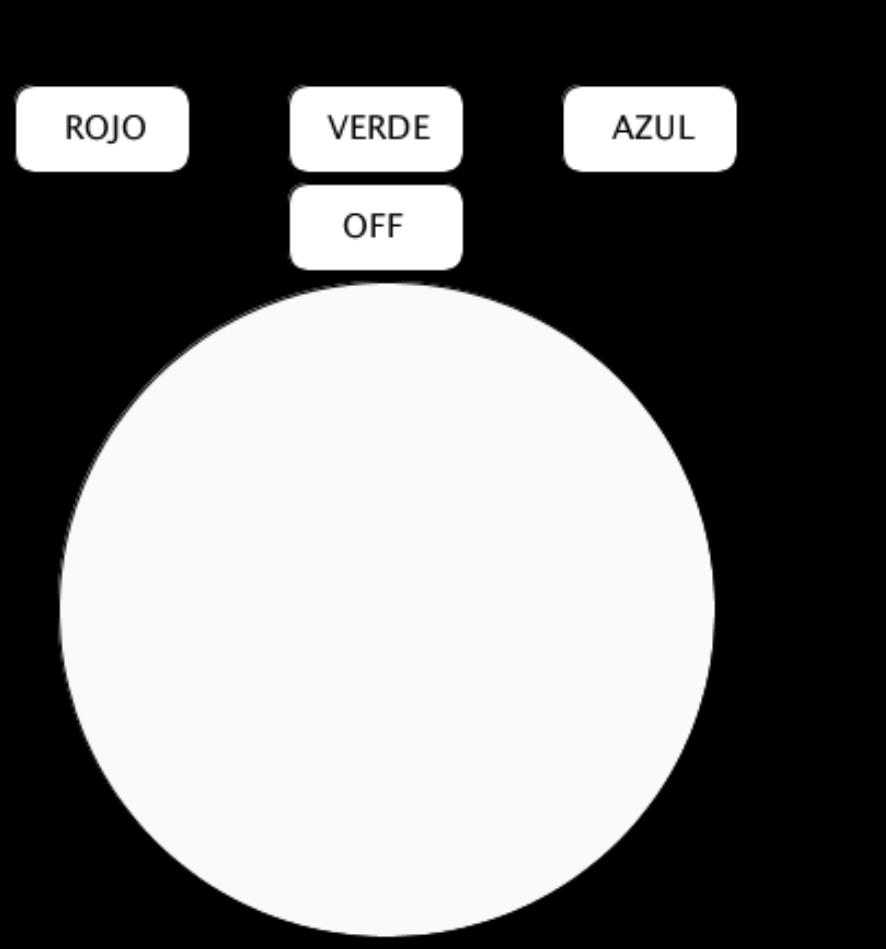




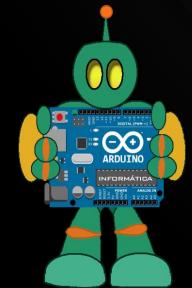
EJERCICIO 2 - SOL. Y PRUEBAS

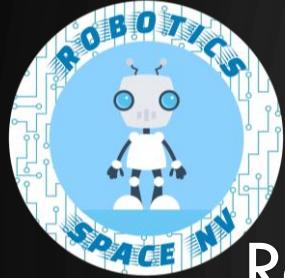
RGB ▾

```
52 void mousePressed(){
53   if((mouseX>50 & mouseX<130)&(mouseY>60 & mouseY<100)){
54     fill(60,60,60);
55     rect(50,60,80,40,10);
56     puerto.write('R');
57     EstadoBoton=1;
58     println("Rojo");
59   }
60   if((mouseX>175 & mouseX<255)&(mouseY>60 & mouseY<100)){
61     fill(128,128,0);
62     rect(175,60,80,40,10);
63     puerto.write('G');
64     EstadoBoton=2;
65     println("Verde");
66   }
67   if((mouseX>300 & mouseX<380)&(mouseY>60 & mouseY<100)){
68     fill(60,60,60);
69     rect(300,60,80,40,10);
70     puerto.write('B');
71     EstadoBoton=3;
72     println("Azul");
73   }
74   if((mouseX>175 & mouseX<255)&(mouseY>105 & mouseY<155)){
75     fill(60,60,60);
76     rect(175,105,80,40,10);
77     puerto.write('A');
78     EstadoBoton=0;
79     println("Apagado");
80   }
81 }
```



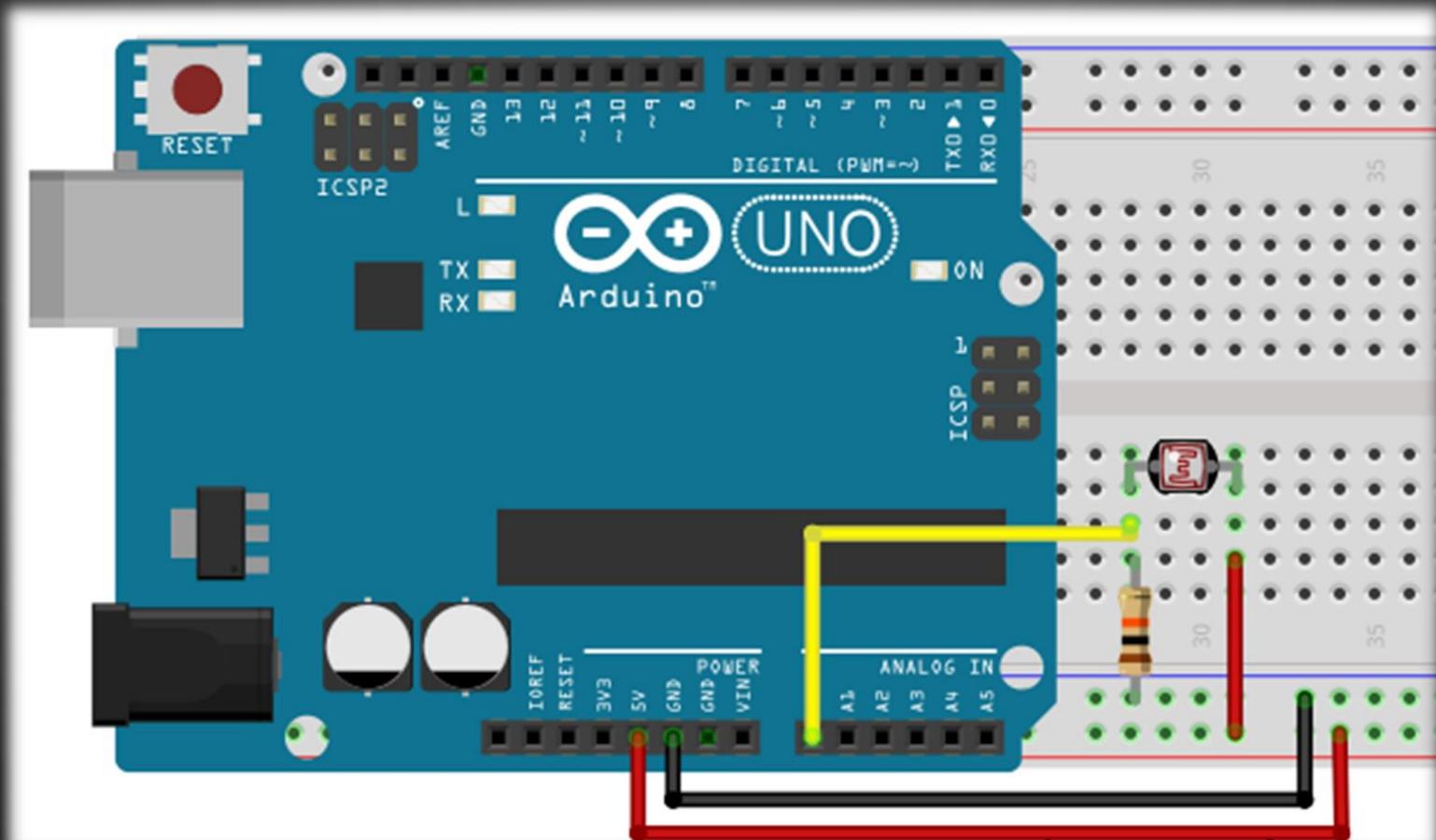
TUTOR: NAGIB LUIS VALLEJOS M.



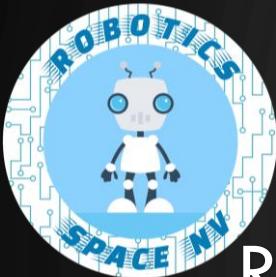


EJERCICIO 3 - CIRCUITO

Recolectar datos de arduino, mostrarlos en Processing y generar un archivo con los datos almacenados



TUTOR: NAGIB LUIS VALLEJOS M.



EJERCICIO 3 - SOL. ARDUINO

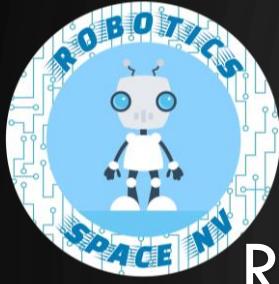
Recolectar datos de arduino, mostrarlos en Processing y generar un archivo con los datos almacenados

S4-E3

```
1 void setup() {  
2     Serial.begin(9600);  
3 }  
4 void loop() {  
5     int dato= analogRead(A0);  
6     int porcentaje=map(dato, 0, 1023, 0, 100);  
7     Serial.write(porcentaje); //0-255  
8     delay(1000);  
9 }
```



TUTOR: NAGIB LUIS VALLEJOS M.



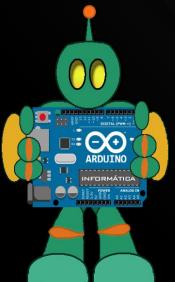
EJERCICIO 3 - SOL. PROCESSING

Recolectar datos de arduino, mostrarlos en Processing y generar un archivo con los datos almacenados

```
archivos ▾

1 import processing.serial.*;
2 Serial puerto;
3 PrintWriter datos;
4 int valor;
5
6 void setup(){
7   puerto= new Serial(this,"COM6",9600);
8   datos= createWriter("porcentajeLDR.txt");
9   size(800,400);
10 }
11 void draw(){
12   background(255,255,255);
13   if(puerto.available()>0){
14     valor=puerto.read();
15   }
16   textSize(25);
17   text("Dato = ",350,200);
18   text(valor,450,200);
19   fill(0,0,153);
20   //Abrimos el archivo
21   datos.print(valor + " unid.      ");
22   datos.print(hour() + ":" );
23   datos.print(minute() + ":" );
24   datos.print(second());
25   datos.println("");
26   datos.flush();
27   delay(1000);
28 }
```

TUTOR: NAGIB LUIS VALLEJOS M.

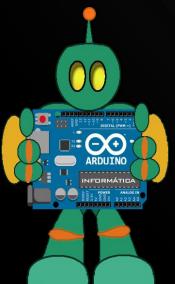


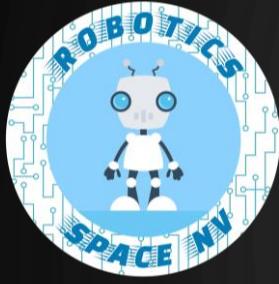


EJERCICIO 3 - PRUEBAS DE FUNCIONAMIENTO

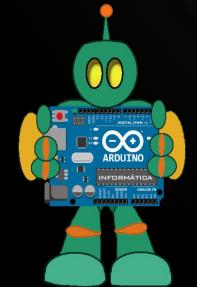
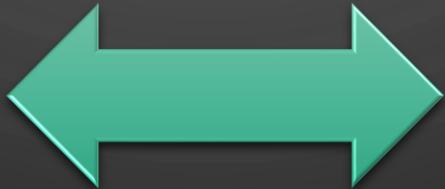
Dato = 57

		porcentajeLDR.txt: Bloc de notas	
		Archivo	Edición
Equipo		53	unid.
Disco local (G:)		51	unid.
Archivos		55	unid.
Arduino		54	unid.
		53	unid.
		55	unid.
		56	unid.
		56	unid.
		55	unid.
		57	unid.
		22:12:10	
		22:12:11	
		22:12:12	
		22:12:13	
		22:12:14	
		22:12:15	
		22:12:16	
		22:12:17	
		22:12:18	
		22:12:19	
		22:12:20	

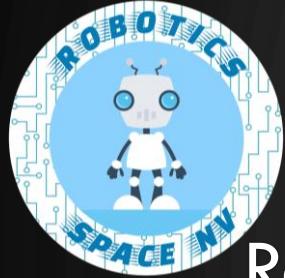




VIDEOJUEGOS CON PROCESSING Y ARDUINO

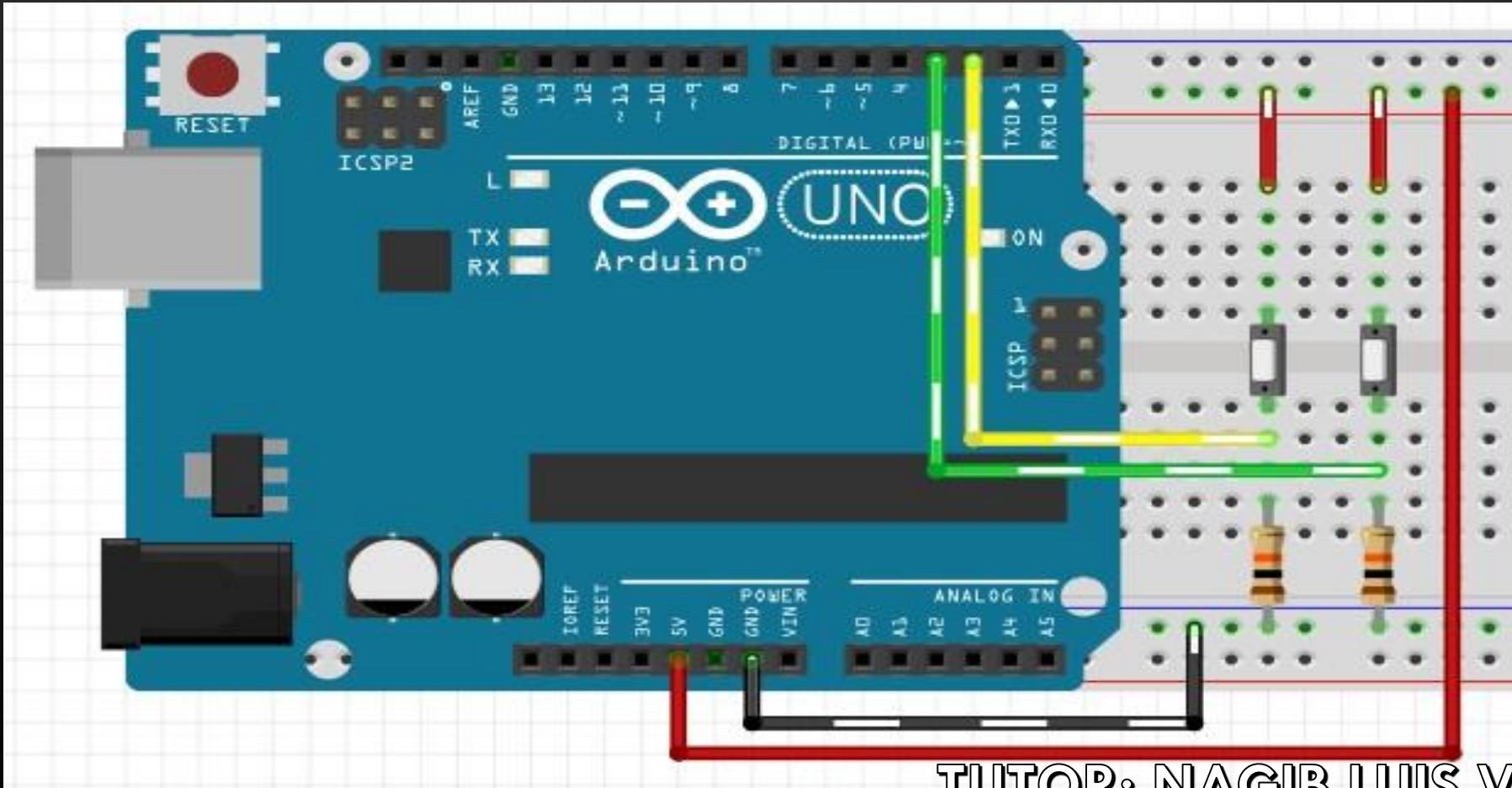


TUTOR: NAGIB LUIS VALLEJOS M.

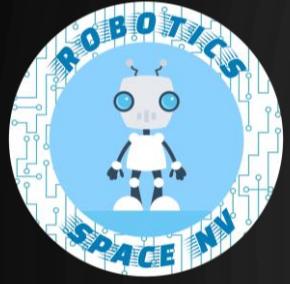


EJERCICIO 4 - CIRCUITO

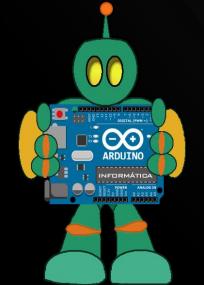
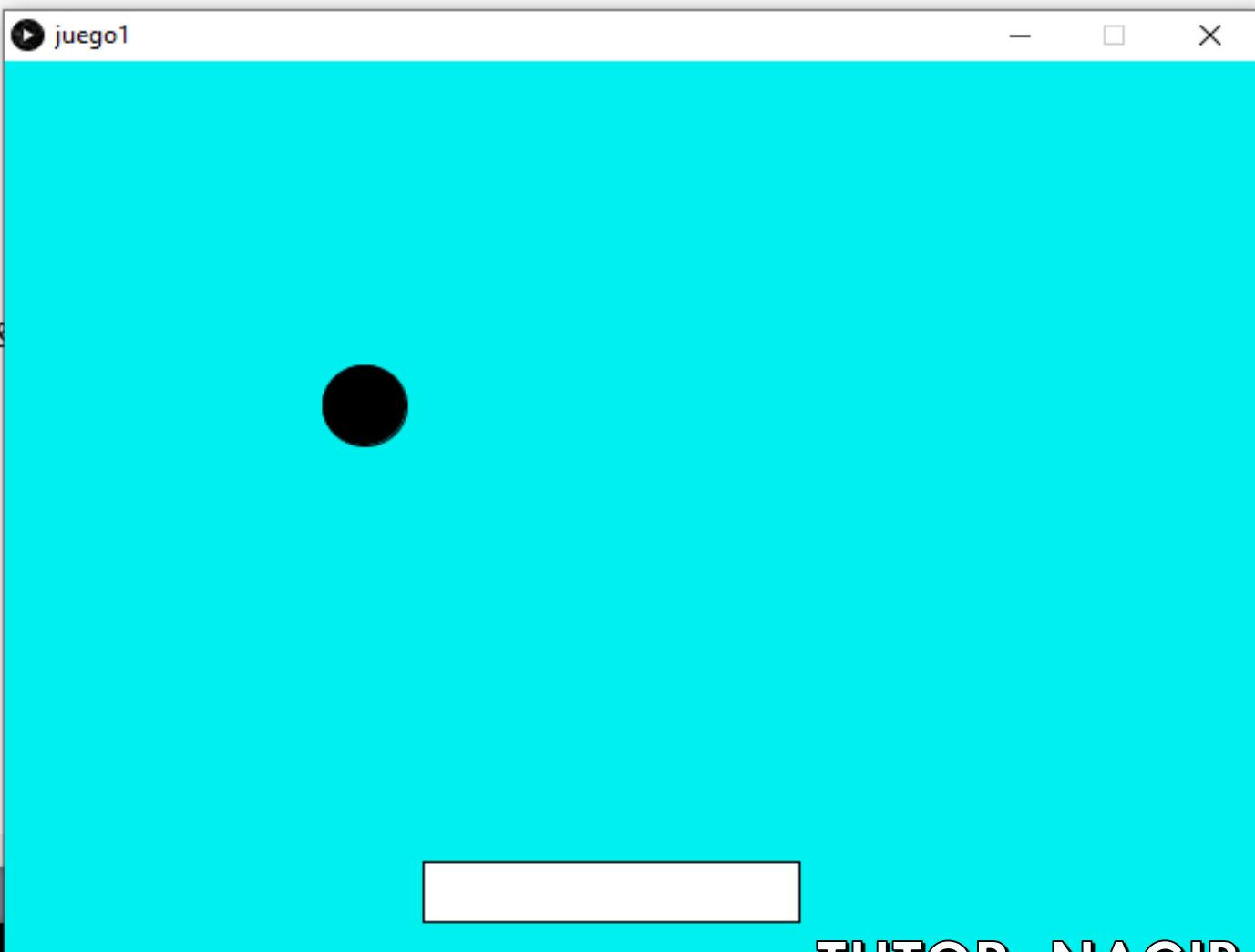
Realizar un videojuego interactuando botones físicos, arduino y Processing



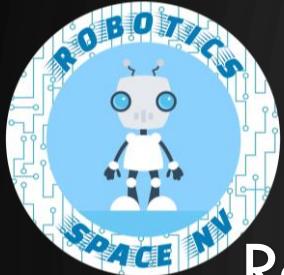
TUTOR: NAGIB LUIS VALLEJOS M.



EJERCICIO 4 - PRUEBAS DE FUNCIONAMIENTO

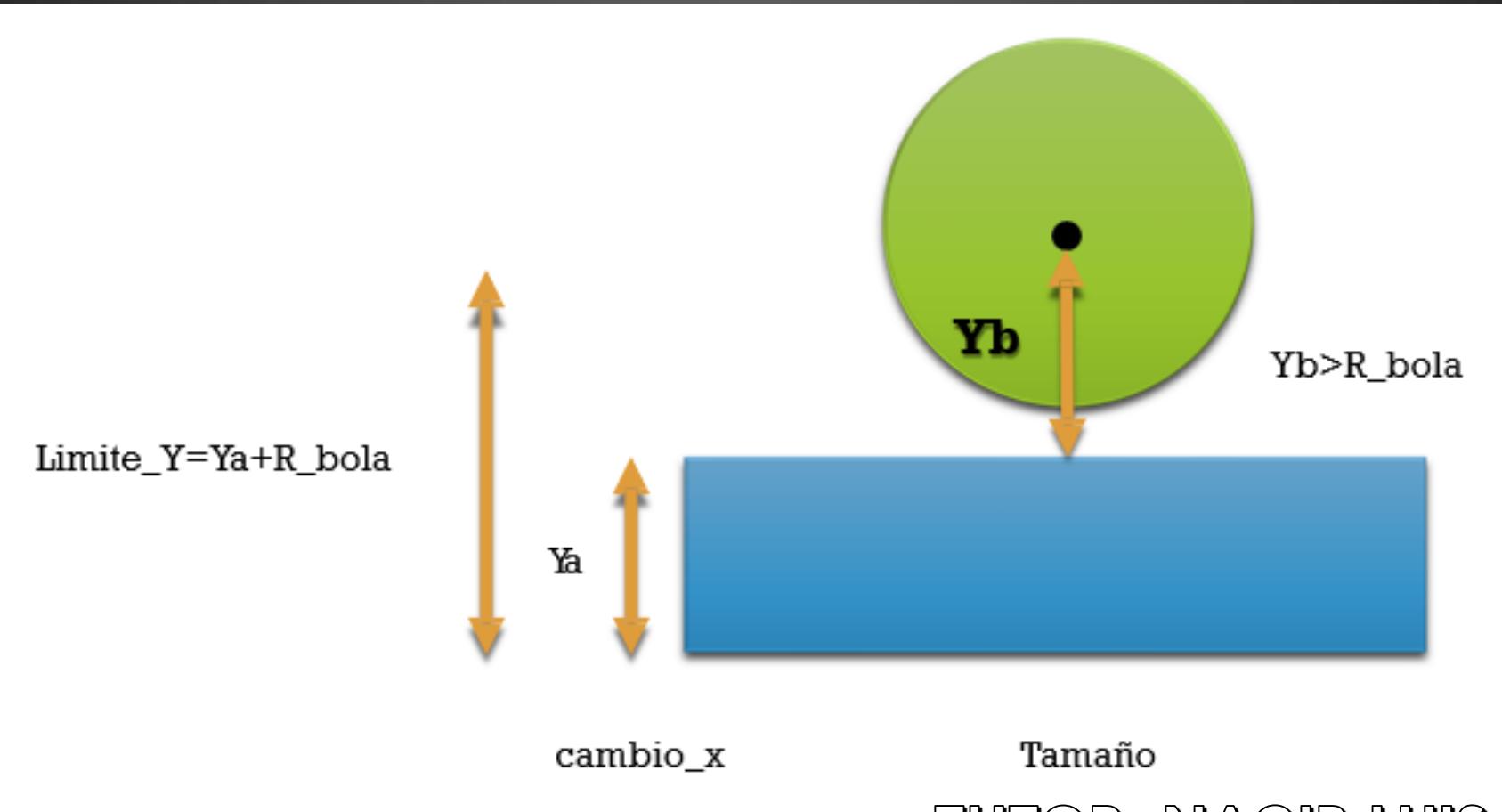


TUTOR: NAGIB LUIS VALLEJOS M.

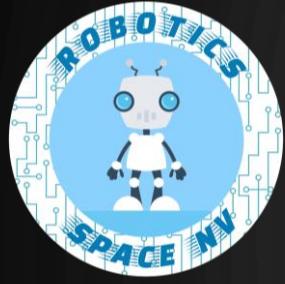


COLISIONES

Realizar un videojuego interactuando botones físicos, arduino y Processing



TUTOR: NAGIB LUIS VALLEJOS M.



EJERCICIO 4 - SOLUCIÓN

La solución a este ejercicio paso a paso se encuentra en el siguiente link, generado como un taller propio:

<https://youtu.be/HrsJwU1SxoQ>



TUTOR: NAGIB LUIS VALLEJOS M.

CONTACTOS



(+591) 63096640



robotics.space.nv@gmail.com



fb.me/RoboticsSpaceNV



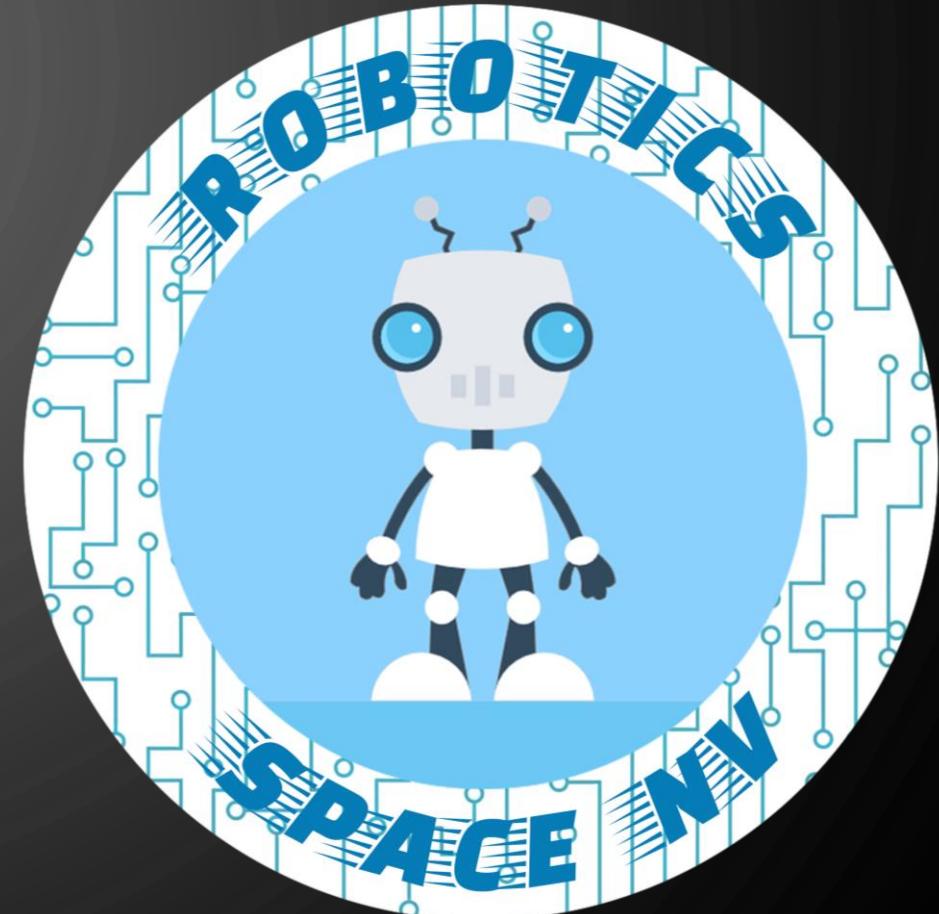
@NagibVallejos



Robotics Space NV



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



TUTOR: NAGIB LUIS VALLEJOS M.