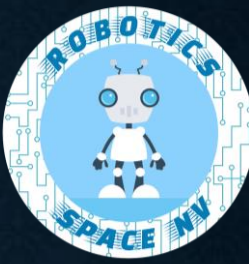


# VIDEOJUEGOS CON ARDUINO Y PROCESSING



EXPOSITOR: NAGIB LUIS VALLEJOS MAMANI



# PROCESSING

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

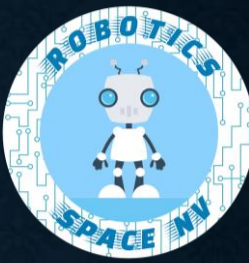
Processing cuenta con diferentes librerías que se encargan de integrar diferentes funciones como por ejemplo: Animaciones en 2D, 3D, aplicaciones visuales, creación de imágenes entre otros.



**EXPOSITOR: NAGIB LUIS VALLEJOS MAMANI**



# COMANDOS BÁSICOS



***draw()***: ejecuta un ciclo continuamente hasta que se detenga el programa (**en la programación viene después de setup()**).

***setup()***: en esta función se colocan los parámetros del entorno gráfico a crearse.

***size()***: función encargada del tamaño de la ventana que crea el programa (medido en píxeles).

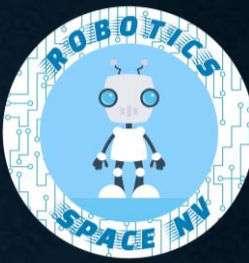
***width***: es el primer parámetro usado en el comando size, se refiere a la parte ancha del objeto.

***height***: es el segundo parámetro que se usa dentro del comando size, se refiere a la parte ancha del objeto.



EXPOSITOR: NAGIB LUIS VALLEJOS MAMANI

# PROCESSING



***rect()***: comando para crear un rectangulo

Ej: ***rect(a, b, c, d);***

***ellipse()***: comando para crear una circunferencia con coordenadas en X y Y

Ej: ***ellipse(x, y, dx, dy);***

***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)***

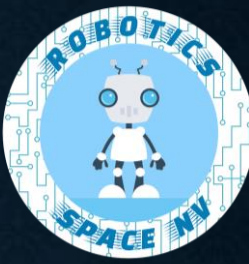
***text("escrito", x, y, z)***



EXPOSITOR: NAGIB LUIS VALLEJOS MAMANI



# COMANDOS BÁSICOS



***textFont()***: comando para colocar el tipo de fuente que va a usar el texto.

***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.

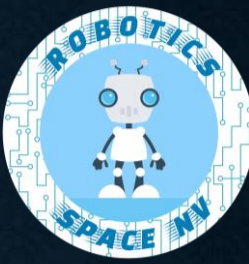
***noStroke()***: imagen o figura sin borde.



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

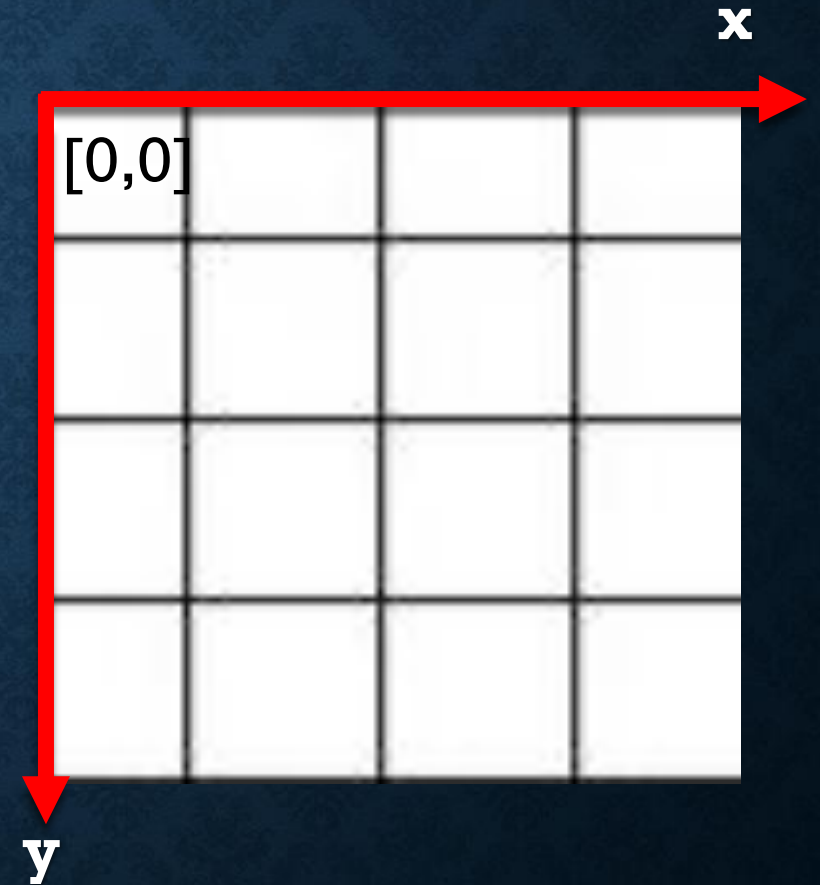
EXPOSITOR: NAGIB LUIS VALLEJOS MAMANI

# 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.

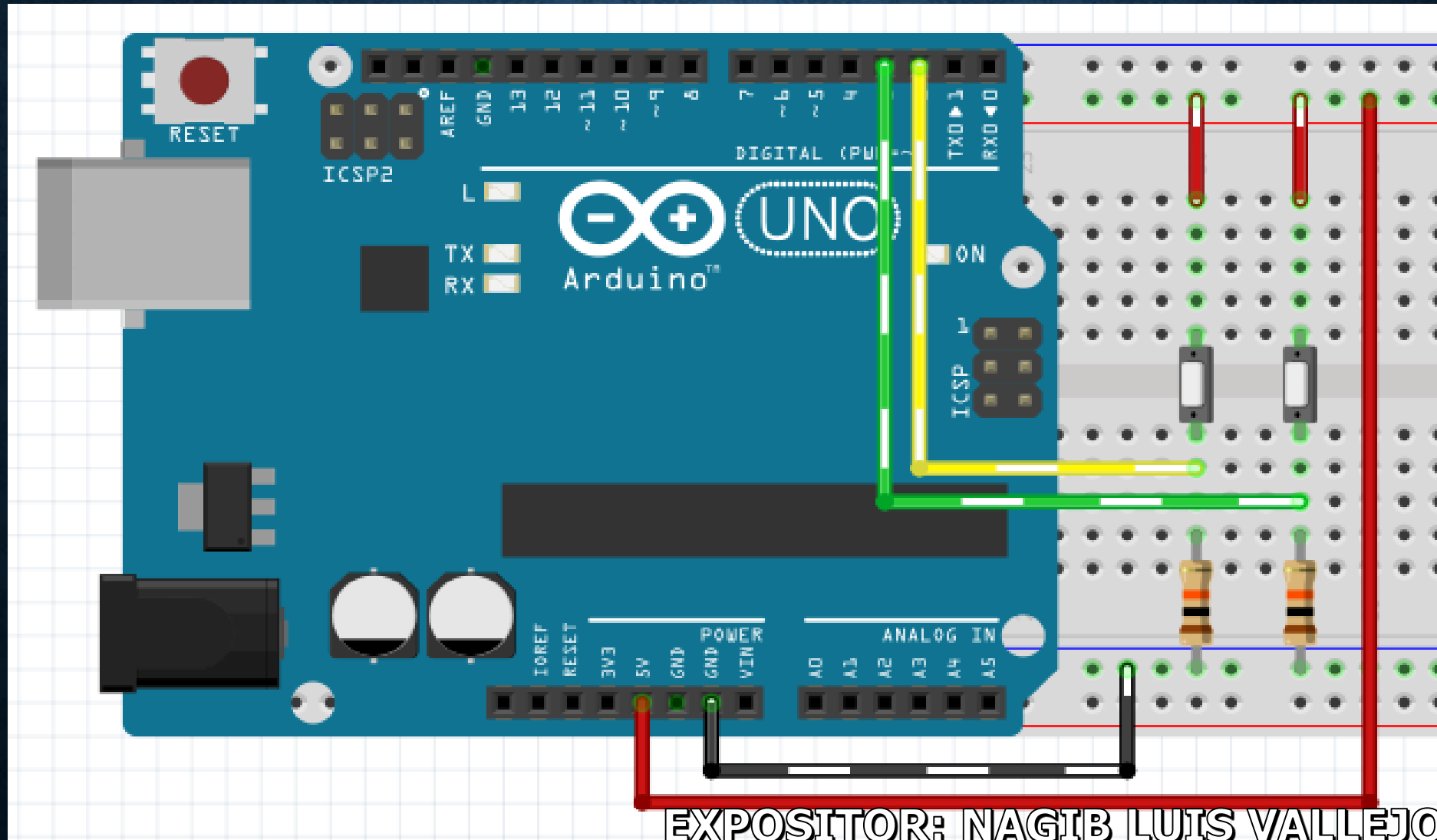
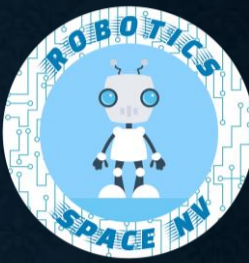


EXPOSITOR: NAGIB LUIS VALLEJOS MAMANI



# EJERCICIO – CIRCUITO

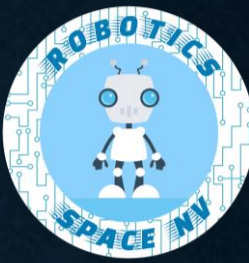
Interactuar entre arduino y Processing a través de pulsadores



EXPOSITOR: NAGIB LUIS VALLEJOS MAMANI

# EJERCICIO – ARDUINO

Interactuar entre arduino y Processing a través de pulsadores



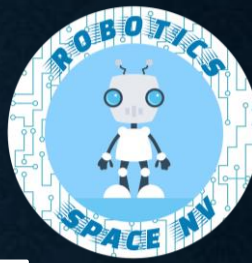
juego

```
1 int bi=2, bd=3, izq, der;
2 void setup() {
3     Serial.begin(9600);
4     pinMode(bi, INPUT);
5     pinMode(bd, INPUT);
6 }
7 void loop() {
8     izq=digitalRead(bi);
9     der=digitalRead(bd);
10    if(izq) {
11        Serial.println("1");
12        delay(20);
13    }
14    if(der) {
15        Serial.println("2");
16        delay(20);
17    }
18    if(izq==0 and der==0) {
19        Serial.println("3");
20        delay(20);
21    }
22 }
```

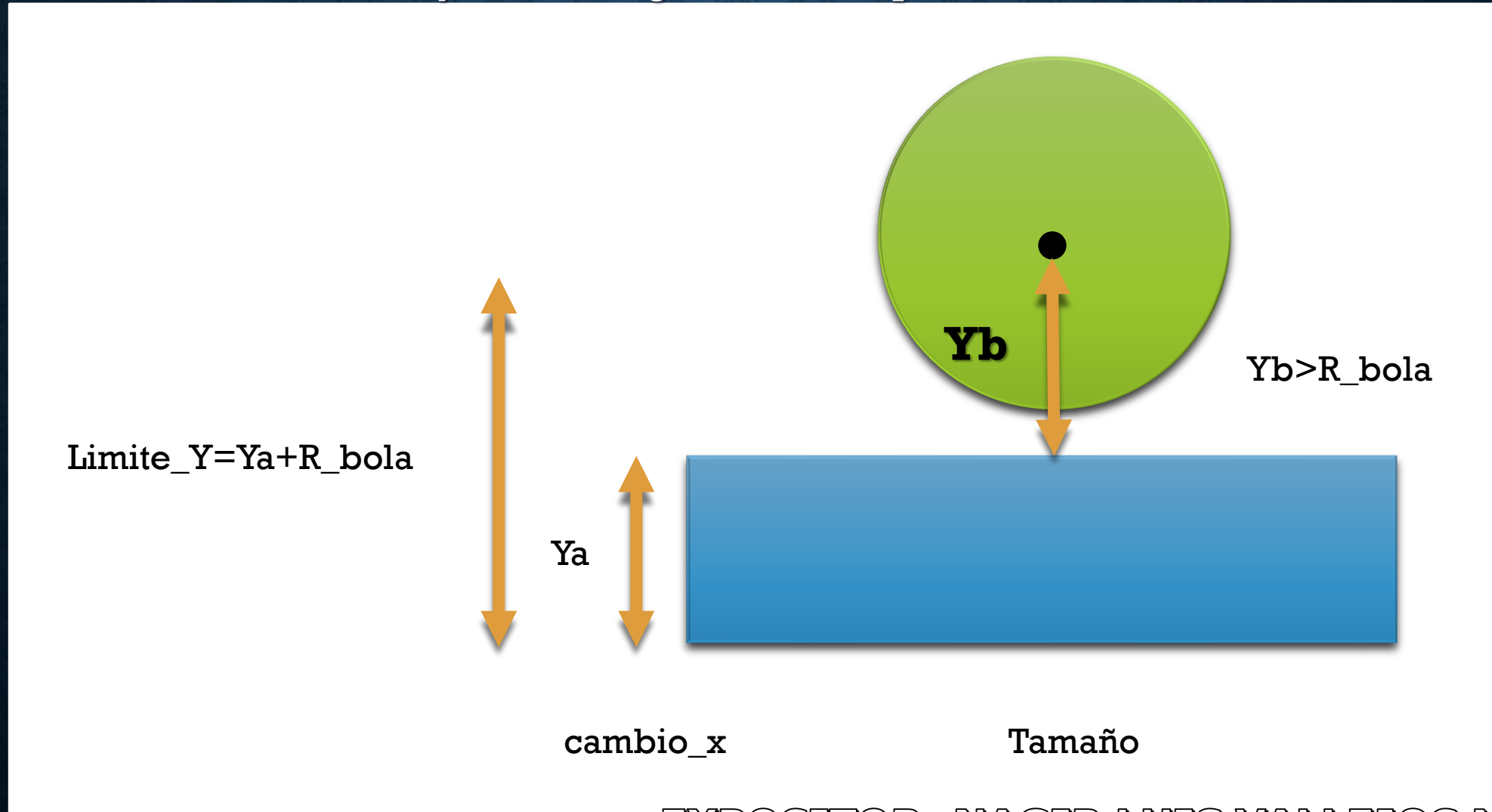
EXPOSITOR: NAGTB LUIS VALLEJOS MAMANI



# PROCESSING - COLISIONES



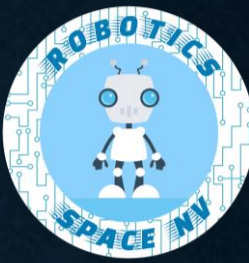
Interactuar entre arduino y Processing a través de pulsadores



EXPOSITOR: NAGIB LUIS VALLEJOS MAMANI

# EJERCICIO – PROCESSING

Interactuar entre arduino y Processing a través de pulsadores



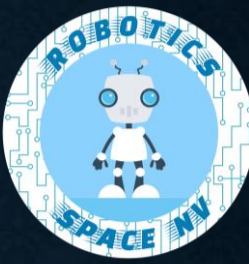
```
juego1 ▾
1 import processing.serial.*;
2 Serial puerto;
3 int x=100, y=100;
4 int velx=3, vely=3;
5 int PosicionX_Temporal;
6 String dato="";
7 float valor;
8 int cambio_x;
9 void setup(){
10     size(600,450);
11     puerto=new Serial(this,"COM6",9600);
12 }
13 void draw(){
14     if(puerto.available()>0){
15         dato=puerto.readStringUntil('\n');
16         if(dato==null){
17             return;
18         }
19         valor=float(dato);
20     }
21     if(valor==1){ //izq
22         PosicionX_Temporal-=10;
23     }
24     else if(valor==2){ //der
25         PosicionX_Temporal+=10;
26     }
27     background(0,240,240);
28     cambio_x=200+PosicionX_Temporal;
29     fill(255,255,255);
30     rect(cambio_x,400,180,30);
31     fill(0,0,0); //color pelotita
32     ellipse(x,y,40,40);
33     x+=velx;
34     if(x>580 || x<20){
35         velx=velx*-1;
36     }
37     y+=vely;
38     if(y>height || y<0){
39         vely=vely*-1;
40     }
41     if(y>=385 &&(x<cambio_x+180)&&(x>cambio_x)){
42         vely=-(vely+1);
43         velx+=1;
44     }
45     if(y>=450){
46         x=100;
47         y=100;
48         velx=3;
49         vely=3;
50     }
51 }
```

EXPOSITOR: NAGIB LUIS VALLEJOS MAMANI



# EJERCICIO – INTERFAZ

Interactuar entre arduino y Processing a través de pulsadores



EXPOSITOR: NAGIB LUIS VALLEJOS MAMANI

# CONTACTOS

Suscríbete



(+591) 63096640



robotics.space.nv@gmail.com



fb.me/RoboticsSpaceNV



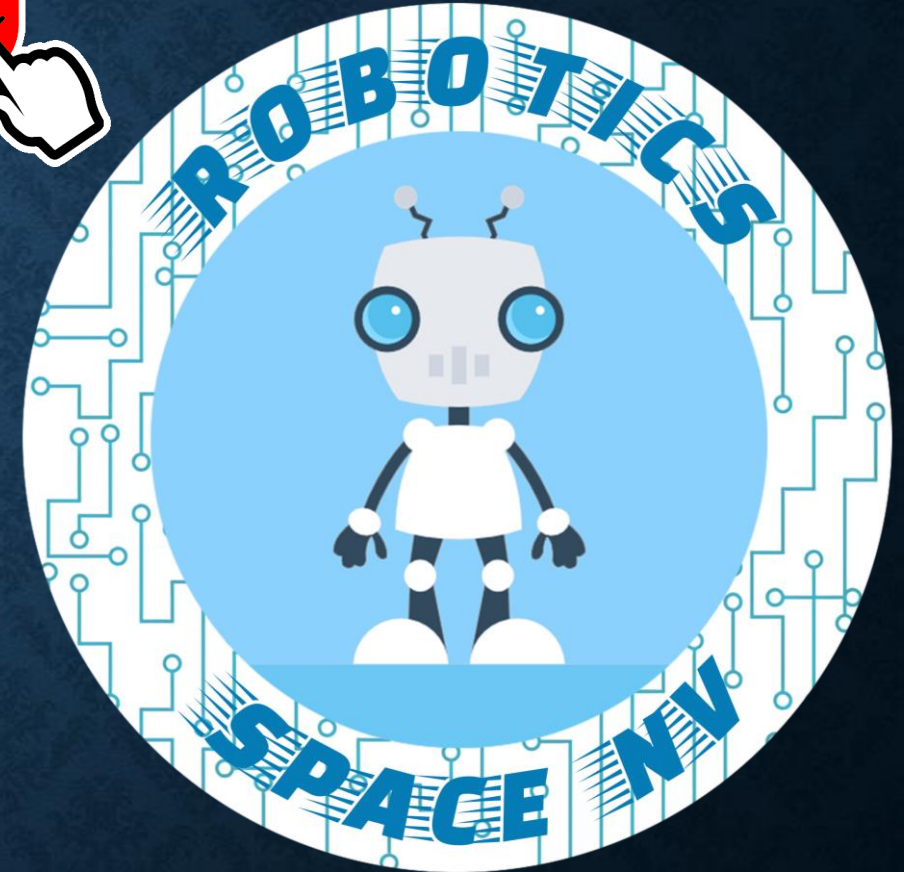
@NagibVallejos



Robotics Space NV



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



TUTOR:NAGIB LUIS VALLEJOS M.