



# Avance del Proyecto “MP3”

Proyecto I  
Integrantes: Fabian Orellana, Karen Correa,  
Iván Viscarra, Jhosep Marca  
Profesor: Humberto Urrutia





# Introducción

- En esta presentación se mostrarán los avances logrados en el proyecto hasta la fecha.
- Se expondrá el estado actual de la interfaz gráfica, los problemas que hemos sorteado, los programas que hemos utilizado y diagramas que nos ayudan a visualizar cómo funciona la conexión remota del robot.
- Además, se demostrará como en este proyecto están involucrados los conocimientos previos de mecánica clásica.





# Requerimientos



## Funcionales

- El robot debe desplazarse según la orden dada por la interfaz grafica.
- El robot debe disparar piezas de lego haciendo uso de las funciones implementadas en la interfaz grafica.
- El robot se debe conectar a un equipo mediante una conexión wifi.

## No Funcionales

- El robot debe tener una estructura estable para que al realizar las funciones solicitadas no derive en daños estructurales.
- Se debe programar la interfaz gráfica en lenguaje de programación Python.
- El robot debe estar construido con las piezas del Kit de LEGO MINDSTORMS EV3

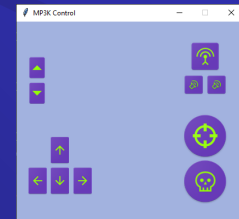


# Arquitectura

Se conecta el robot y el equipo a la misma red wifi.

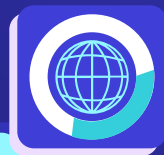
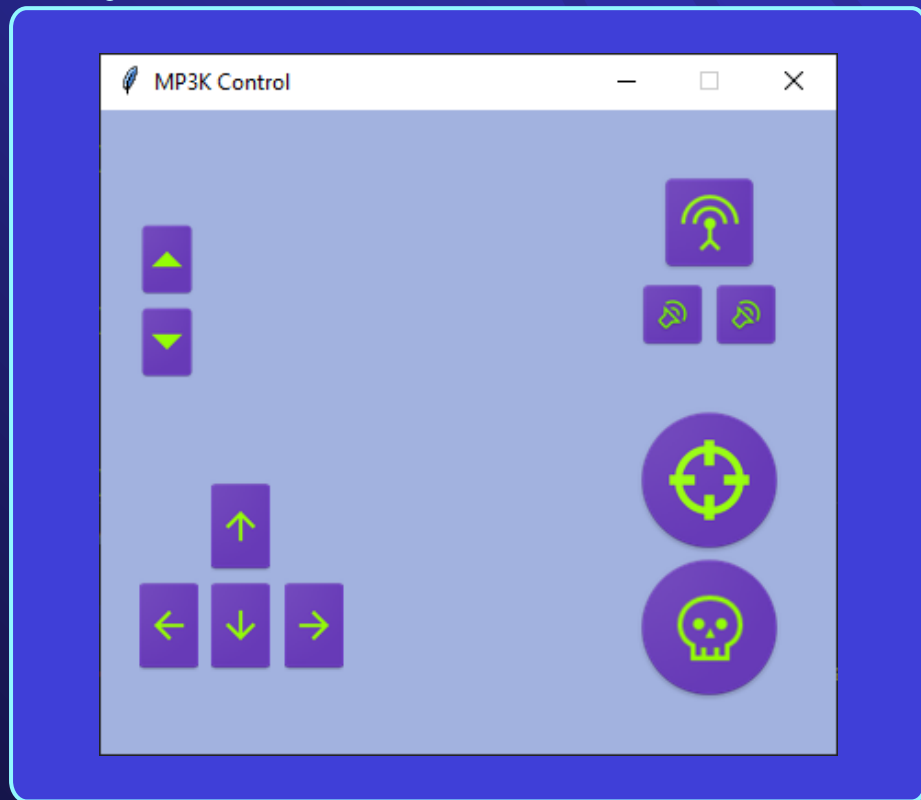
Se inicia el servidor para establecer la conexión remota.

Se inicia la interfaz gráfica para ejecutar las funciones del robot.



La interfaz se conecta al servidor del robot y el usuario puede utilizarla para controlar al robot.

# Interfaz Gráfica



# Fundamentos de Projectiles

**Movimiento Rectilíneo Uniforme (M.R.U):** En el eje horizontal

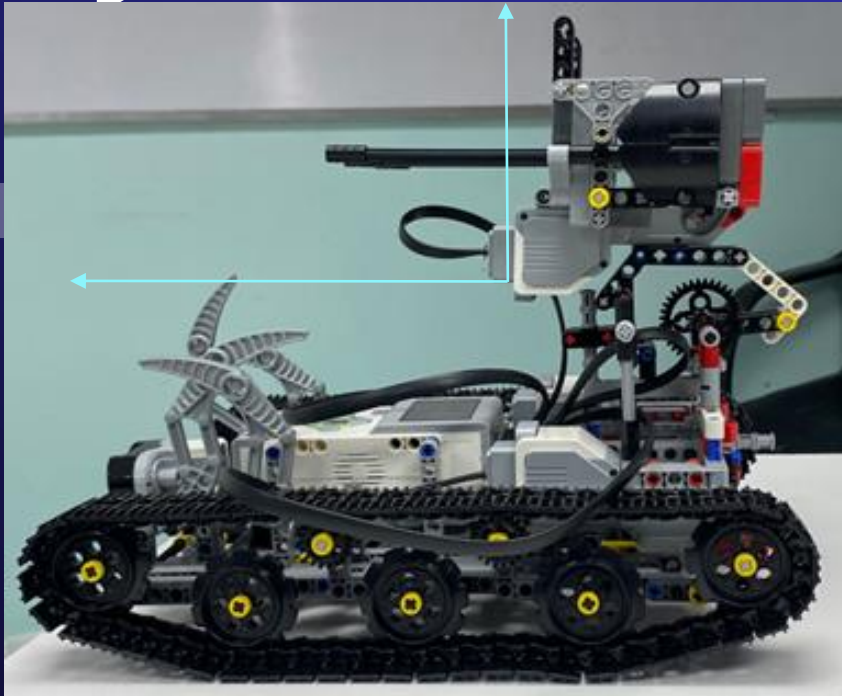
$$x = x_0 + v_x * t$$

$$v_x = v_0 * \cos\theta = cte$$

**Movimiento Rectilíneo Uniforme Acelerado (M.R.U.A):** En el eje vertical

$$y = y_0 + v_{0y} * t + \frac{1}{2} * a_y * t^2$$

$$v_y = v_0 * \sin\theta - g * t$$



$$\begin{aligned} \theta &= 30^\circ \\ X_0 &= 0 \\ Y_0 &= 0.19 \text{ [m]} \\ X_f &= 1.24 \text{ [m]} \\ Y_f &= 0 \\ X(t) &= X_0 + V_{0x} \cdot t \\ 1.24 &= 0 + V_0 \cdot \cos 30^\circ \cdot t \quad (1) \end{aligned}$$

$$\begin{aligned} Y(t) &= Y_0 + V_{0y} \cdot t - \frac{1}{2} g \cdot t^2 \\ 0 &= 0.19 + V_0 \sin 30^\circ \cdot t - 4.9 t^2 \quad (2) \end{aligned}$$

$$\begin{aligned} (1) \quad 1.24 &= 0 + V_0 \cos 30^\circ t \\ (2) \quad 0 &= 0.19 + V_0 \sin 30^\circ t - 4.9 t^2 \end{aligned}$$

$$\text{De } t = \frac{1.24}{V_0 \cos 30^\circ}$$

Reemplazando  $t$  en (2):

$$0 = 0.19 + V_0 \sin 30^\circ \cdot \frac{1.24}{V_0 \cos 30^\circ} - 4.9 \cdot \frac{1.24^2}{V_0^2 \cos^2 30^\circ}$$

$$0 = 0.19 + 0.72 - \frac{10.04}{V_0^2}$$

$$\frac{10.04}{V_0^2} = 0.91$$

$$\begin{aligned} \frac{10.04}{0.91} &= V_0^2 \\ 11.03 &= V_0^2 \\ 3.32 &= V_0 \end{aligned}$$

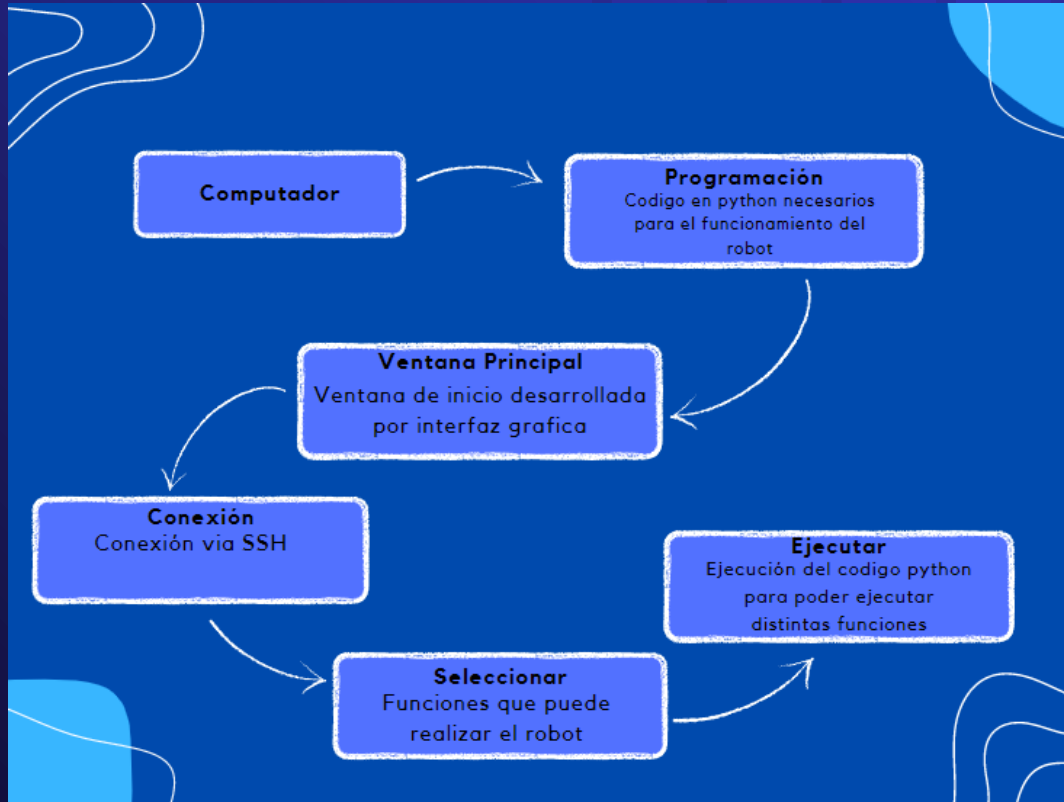
$$t = \frac{1.24}{3.32 \cos 30^\circ} = 0.43$$

Los datos obtenidos son:

- Tiempo: 0.43 [s]
- Distancia máxima: 1.24 [m]
- Altura: 0.19 [m]
- Ángulo:  $30^\circ$



# Diagramas





# Implementación

```
1 import socket
2 from Function import *
3
4 s = socket.socket()
5 print("Socket creado")
6 port = 8080
7 s.bind('', port)
8 print("El socket se creo con puerto: {}".format(port))
9 s.listen(5)
10 print("EL socket is listening...")
11 connect, addr = s.accept()
12 print("Se conecto a {}".format(addr))
13 while True:
14     rawByte = connect.recv(1)
15     char = rawByte.decode('utf-8')
16     if (char == 'w'):
17         moveUp()
18     if (char == 's'):
19         moveDown()
20 #Problema al mover a la derecha
21     if (char == 'd'):
22         moveRight()
23 #Problema al mover a la izquierda
24     if (char == 'a'):
25         moveLeft()
26     if (char == 'q'):
27         print("Terminando la sesion...")
28         break
29     if (char == 'p' or char == 'b'):
30         levantarTorreta(char)
31     if (char == 'z'):
32         disparar()
33     if (char == 'r'):
34         apuntar()
35     if (char == 'm' or char == 'n'):
36         sonido(char)
37     if (char == 'c'):
38         stop()
```

# Implementación

```
1 from ev3dev2.motor import LargeMotor, MediumMotor, OUTPUT_A, OUTPUT_B, OUTPUT_C, OUTPUT_D, MoveTank
2 from ev3dev2.sound import Sound
3 from ev3dev2.sensor import INPUT_4
4 from ev3dev2.sensor.lego import UltrasonicSensor
5
6 mT = LargeMotor(OUTPUT_A)
7 mI = LargeMotor(OUTPUT_B)
8 mD = LargeMotor(OUTPUT_C)
9 mL = MediumMotor(OUTPUT_D)
10
11 us = UltrasonicSensor(INPUT_4)
12 us.mode = "US-SI-CM"
13
14 spkr = Sound()
15
16 tankmoves = MoveTank(OUTPUT_B, OUTPUT_C)
17
18 def moveUp():
19     print("Moving up...")
20     tankmoves.on(100, 100)
21
22 def moveDown():
23     print("Moving down...")
24     tankmoves.on(-100, -100)
25
26 def moveRight():
27     print("Moving right...")
28     tankmoves.on(100, 1)
29
30 def moveLeft():
31     print("Moving left...")
32     tankmoves.on(1, 100)
33
```

```
34 def levantarTorreta(char):
35     print("Mover torreta")
36     if char == 'p':
37         mL.on_for_rotations(speed = 30, rotations = 0.2, block = True)
38     else:
39         mL.on_for_rotations(speed = 30, rotations = -0.2, block = True)
40
41 def disparar():
42     print("disparando")
43     mT.on_for_rotations(speed = 100, rotations = 1)
44
45 def apuntar():
46     print("distancia")
47     print(us.value()/10)
48
49 def sonido(char):
50     print("reproduciendo")
51     if char == 'm':
52         spkr.play_file("losvoyamatarfuerte.wav")
53     else:
54         spkr.play_file("firefire2fuerte.wav")
55 def stop():
56     tankmoves.stop()
```

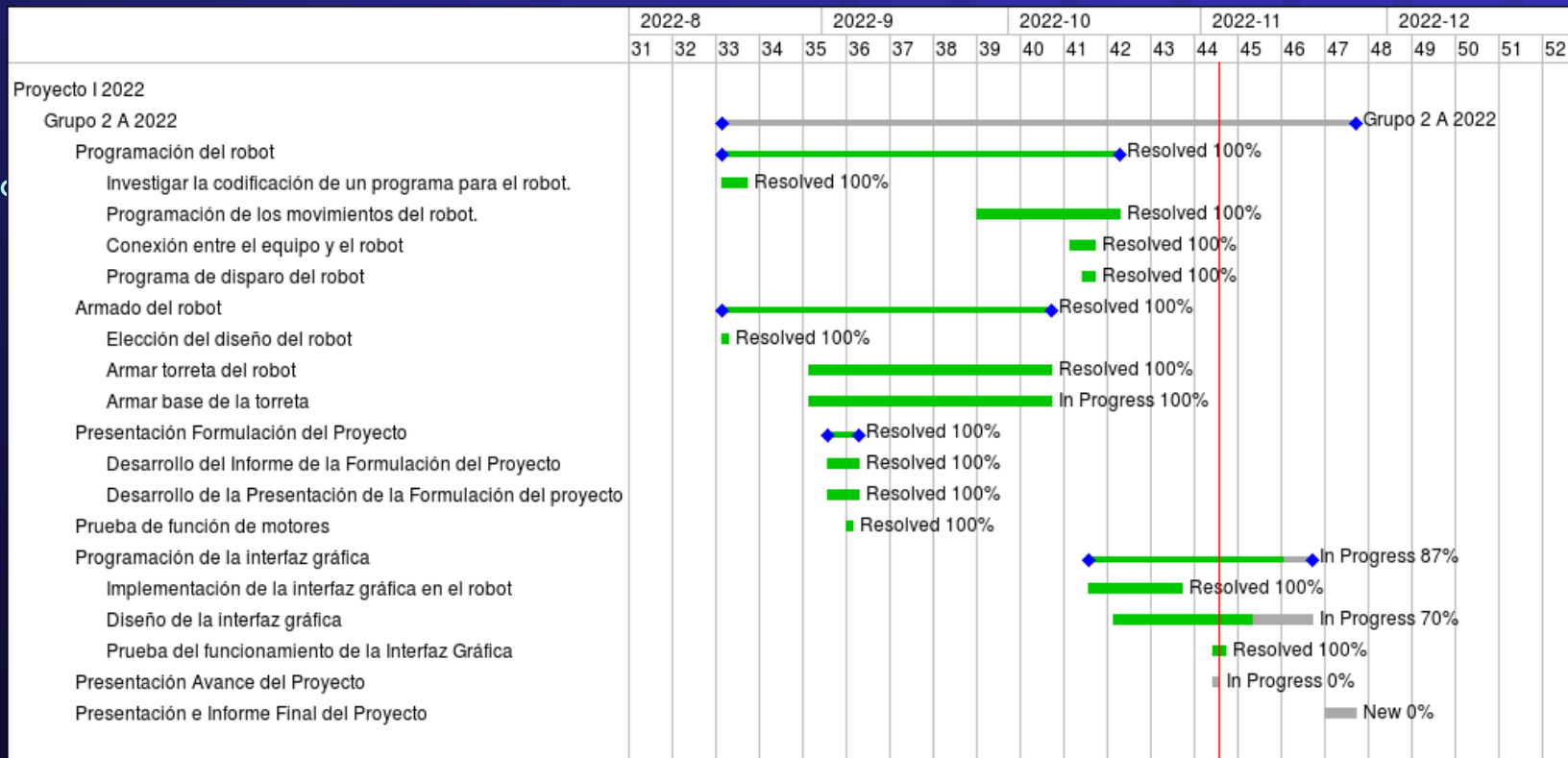
# Implementación

```
60 ventana = Tk()
61 ventana.title("MP3K Control")
62 ventana.resizable(False, False)
63 ventana.config(bg="#A2B2DF")
64
65
66 imagen_arriba = utl.leer_imagen("./imagenes/up_arrow.png", (50,50))
67 imagen_abajo = utl.leer_imagen("./imagenes/down_arrow.png", (50,50))
68 imagen_derecha = utl.leer_imagen("./imagenes/right_arrow.png", (50,50))
69 imagen_izquierda = utl.leer_imagen("./imagenes/left_arrow.png", (50,50))
70
71 imagen_arriba_trgl = utl.leer_imagen("./imagenes/up_trgl.png", (40,40))
72 imagen_abajo_trgl = utl.leer_imagen("./imagenes/down_trgl.png", (40,40))
73
74 imagen_conectar = utl.leer_imagen("./imagenes/conectar.png", (60,60))
75 imagen_skul = utl.leer_imagen("./imagenes/skul.png", (80,80))
76 imagen_apuntar = utl.leer_imagen("./imagenes/apuntar.png", (80,80))
77 imagen_sonido = utl.leer_imagen("./imagenes/sound_icon.png", (40,40))
78
79 btn_up = Button(None, image=imagen_arriba, width=30 ,bd=0 ,bg="#A2B2DF", repeatdelay=50,repeatinterval=50, command = backward,
80                 activebackground="#A2B2DF")
81 btn_up.place(x=60, y=200)
82
83 btn_conectar = Button(ventana, image=imagen_conectar,bd=0, bg="#A2B2DF", command=lambda:[conectar(ipAddress,port)],
84                       activebackground="#A2B2DF").place(x=300, y=30)
85
86 btn_down = Button(ventana, image=imagen_abajo, width=30 ,bd=0 ,bg="#A2B2DF", repeatdelay=50,repeatinterval=50, command = forward,
87                  activebackground="#A2B2DF")
88 btn_down.place(x=60, y=254)
89
90 btn_left = Button(ventana, image=imagen_izquierda, width=30 ,bd=0 ,bg="#A2B2DF", repeatdelay=50,repeatinterval=50, command = right,
91                  activebackground="#A2B2DF")
92 btn_left.place(x=21, y=254)
93
94 btn_right = Button(ventana, image=imagen_derecha, width=30 ,bd=0 ,bg="#A2B2DF", repeatdelay=50,repeatinterval=50, command = left,
95                   activebackground="#A2B2DF")
96 btn_right.place(x=100, y=254)
```

# Implementación

```
98 btn_disparo = Button(ventana, image=imagen_skul,bd=0, bg="#A2B2DF", repeatdelay=550,repeatinterval=550, command =disparar,
99                        activebackground="#A2B2DF")
100 btn_disparo.place(x=290, y=240)
101
102 btn_apuntar = Button(ventana, image=imagen_apuntar,bd=0, bg="#A2B2DF", command =apuntar,
103                     activebackground="#A2B2DF")
104 btn_apuntar.place(x=290, y=160)
105
106 btn_turret_up= Button(ventana,image=imagen_arriba_trgl,bd=0, bg="#A2B2DF", command =levantarTorreta,
107                      activebackground="#A2B2DF")
108 btn_turret_up.place(x=15, y=60)
109
110 btn_turret_down= Button(ventana, image=imagen_abajo_trgl,bd=0, bg="#A2B2DF", command =bajarTorreta,
111                        activebackground="#A2B2DF")
112 btn_turret_down.place(x=15, y=105)
113
114 btn_sound= Button(ventana, image=imagen_sonido,bd=0, bg="#A2B2DF",
115                 activebackground="#A2B2DF", command = sonido1).place(x=290, y=90)
116
117 btn_sound2= Button(ventana, image=imagen_sonido,bd=0, bg="#A2B2DF",
118                  activebackground="#A2B2DF", command = sonido2).place(x=330, y=90)
119
120 btn_turret_up.bind('<<ButtonRelease-1>>',on_release)
121 btn_turret_down.bind('<<ButtonRelease-1>>',on_release)
122 btn_disparo.bind('<<ButtonRelease-1>>',on_release)
123
124 btn_up.bind('<<ButtonRelease-1>>',on_release)
125 btn_down.bind('<<ButtonRelease-1>>',on_release)
126 btn_left.bind('<<ButtonRelease-1>>',on_release)
127 btn_right.bind('<<ButtonRelease-1>>',on_release)
128
129 clientSocket = socket.socket()
130 ipAddress = "192.168.71.153"
131 port = 8080
132
133 utl.centrar_ventana(ventana,400,350)
134 ventana.mainloop()
```

# Estado Actual del Proyecto



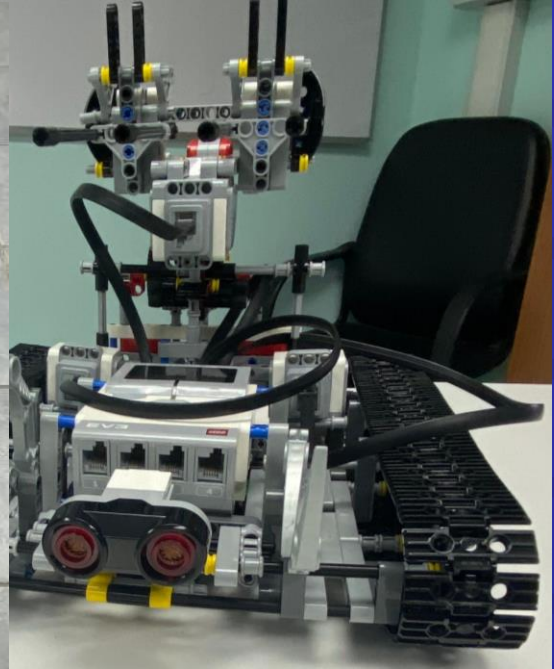
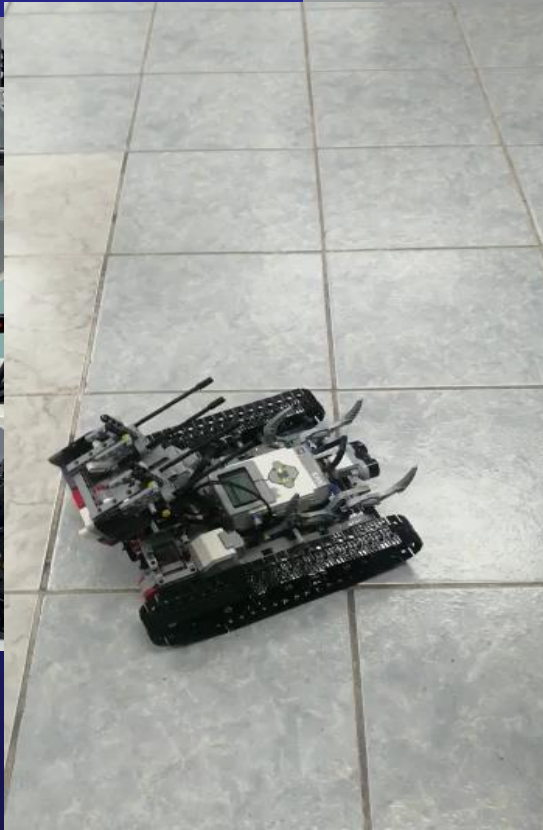
# Estado Actual del Proyecto

La estructura del robot ya está en su versión final, es estable al desplazarse de un punto a otro, su inclinación de posicionamiento para realizar los disparos funciona correctamente.

La información en la plataforma redmine está en constante actualización, subiendo las bitácoras, informes y fotos del avance del proyecto.

La interfaz gráfica está en un estado avanzado de su desarrollo.

# Estado Actual del Proyecto



# Problemas y Solución

1

## Cumplir con las fechas de la Carta Gantt

Reagendar las fechas de actividades programadas e ir avanzando de antemano con las que se pueda para compensar los plazos estipulados y no retrasar las entregas de avances.

## 2 Falta de Piezas

Buscar alternativas que cumplan las mismas funciones que las piezas que no están a disposición.

3

## Estabilidad de la estructura

Buscar distintas piezas que aporten la estabilidad a la torreta para que, al realizar los momentos debidos, no se ocasione ningún percance.





# Conclusión

- Durante el transcurso de este proyecto se reforzaron las habilidades de trabajo en equipo, las cuales son de importancia para avanzar en el proyecto de forma más rápida y facilitar el proceso de encontrar la solución a problemas.
- Se reforzó la programación con Python, a la cual se le sumaron nuevas técnicas para programar refiriéndose al trabajo con librerías e interfaz gráfica.
- Se aprendió a evadir ágilmente los distintos problemas que fueron apareciendo a medida que se realizaba el proyecto.