



UNIVERSIDAD DE TARAPACÁ  
*Universidad del Estado*

Ingeniería@  
Computación e Informática

# GOLFFENHEIMER

Alumnos: Bruno Améstica  
Jorge Cáceres  
Ignacio Garrido  
Katalina Oviedo  
Fernando Pizarro

Asignatura: Proyecto I  
Profesor: Humberto Urrutia

# CONTENIDO

- 01** INTRODUCCIÓN
- 02** ANÁLISIS Y DISEÑO
- 03** IMPLEMENTACIÓN
- 04** RESULTADOS
- 05** CONCLUSIÓN



# INTRODUCCIÓN

# OBJETIVOS

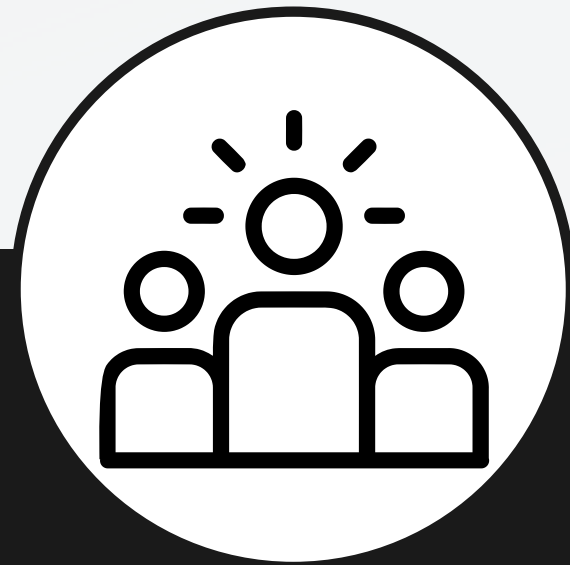
## Objetivo General

- Configurar un robot utilizando el set LEGO MINDSTORMS EV3, que sea capaz de golpear una pelota de golf a través de una interfaz gráfica programada en Python que permita controlar todas y cada una de las acciones del robot.

## Objetivos Especificos

- Diseñar software efectivo para el control del robot.
- Construir un diseño de robot útil, que cumpla con lo requerido.
- Realizar pruebas para evaluar el funcionamiento y el rendimiento del robot, para identificar y corregir posibles errores o mejoras.

# CAMBIO DE ROLES



Jefe de grupo

Fernando Pizarro



Programador

Katalina Oviedo



Documentador

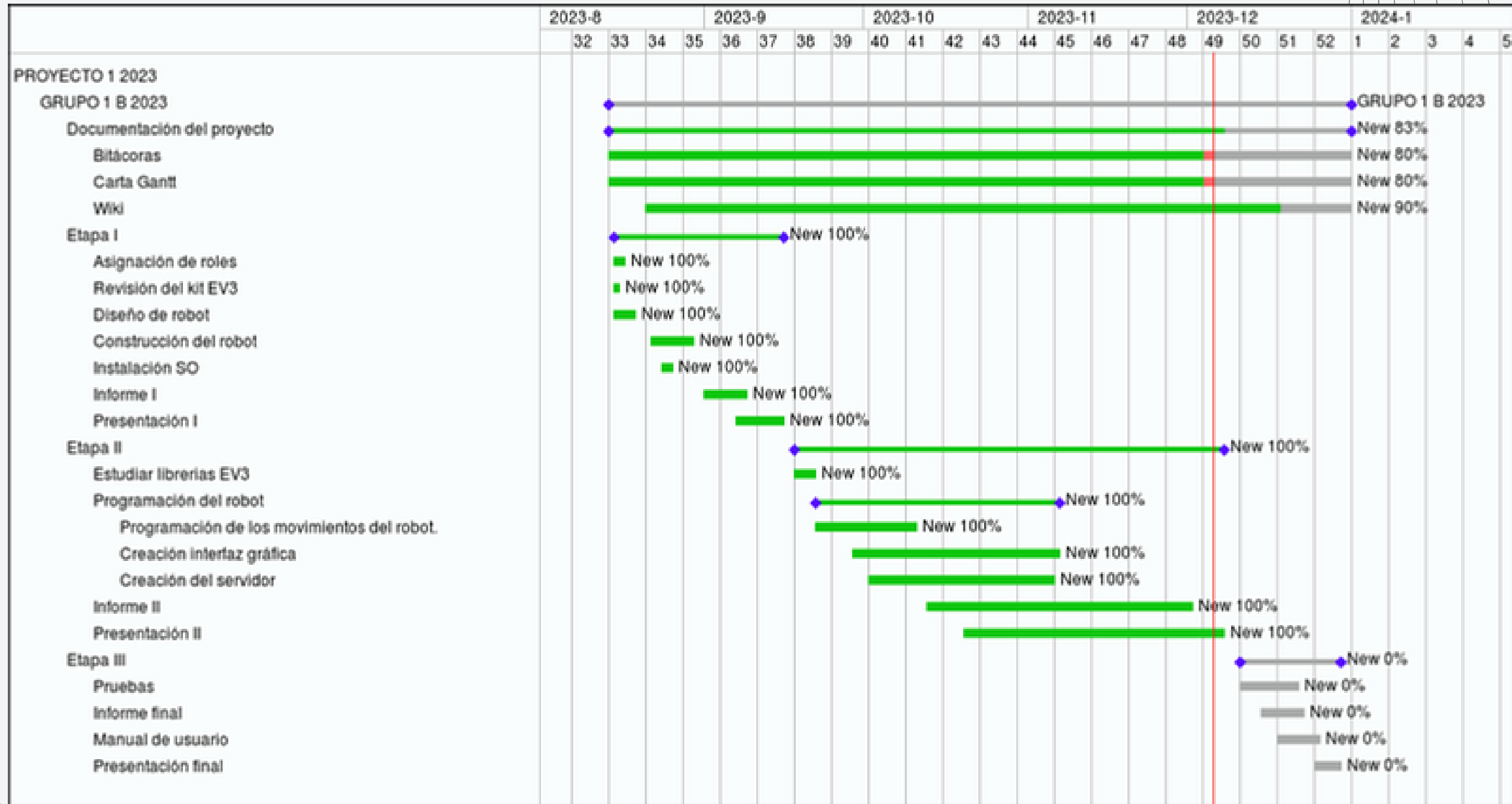
Fernando Pizarro  
Bruno Améstica



Diseñador

Jorge Cáceres  
Ignacio Garrido

# CARTA GANTT



# COSTO TOTAL

<b>Costos</b>	<b>Costo total</b>
Costos de Hardware	\$2.257.980
Costos de Software	\$0
Costos de Gestión	\$5.082.000
Costo total del proyecto	\$7.339.980





**ANÁLISIS Y  
DISEÑO**



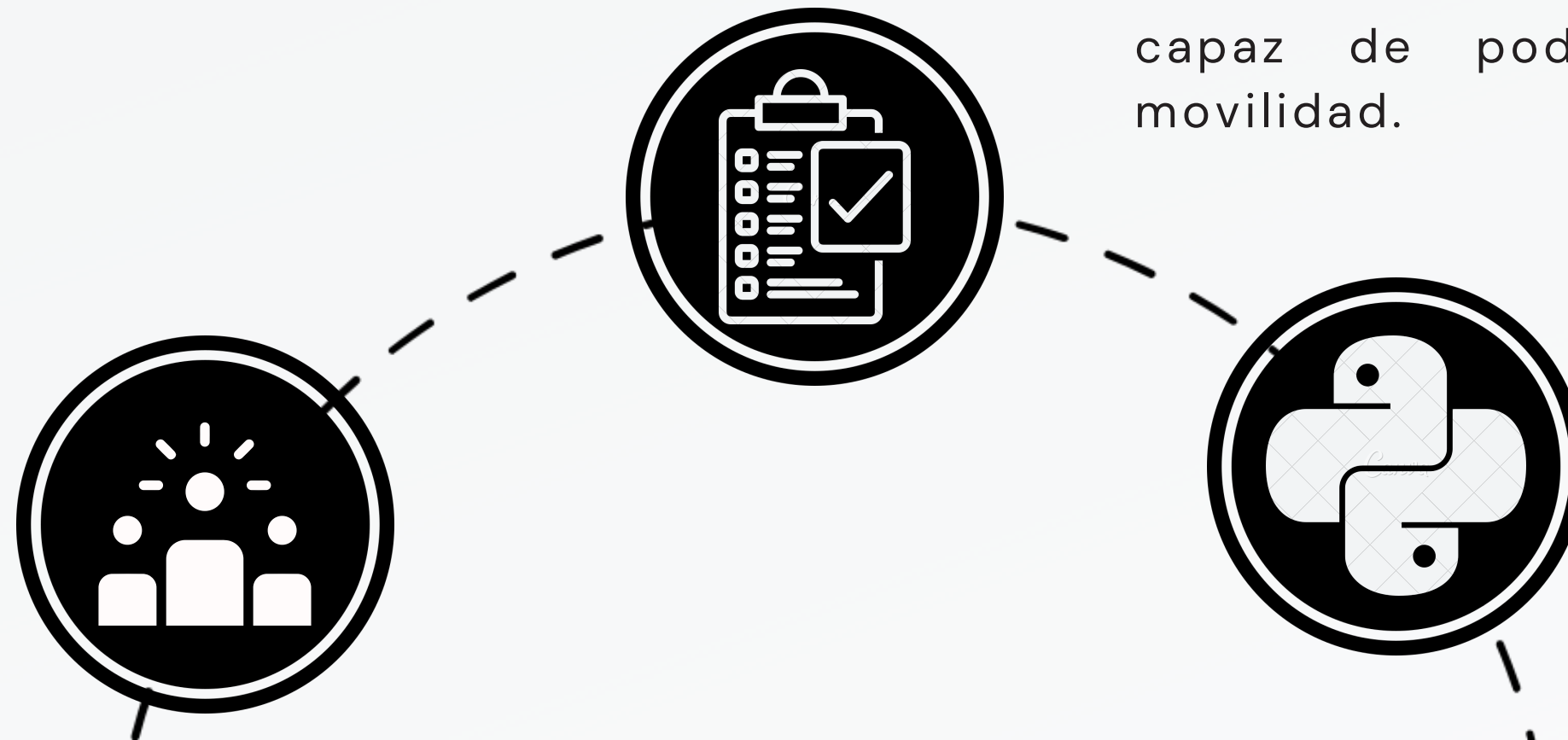
# REQUERIMIENTOS

## Funcionales

- El robot debe ser controlado mediante la implementación de una interfaz gráfica.
- El robot debe tener la capacidad de moverse.
- El robot debe tener la capacidad de golpear una pelota de golf.

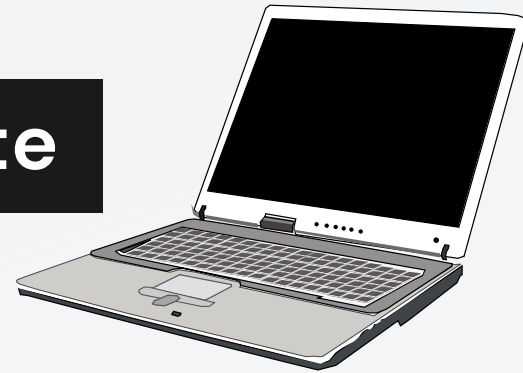
## No Funcionales

- La elaboración del código debe ser con el lenguaje de programación Python.
- La interfaz gráfica debe contener los movimientos del robot
- El robot debe ser construido con pieza de legos kit Mindstorm EV3 Education.
- La estructura del robot debe ser estable y capaz de poder realizar todo tipo de movilidad.



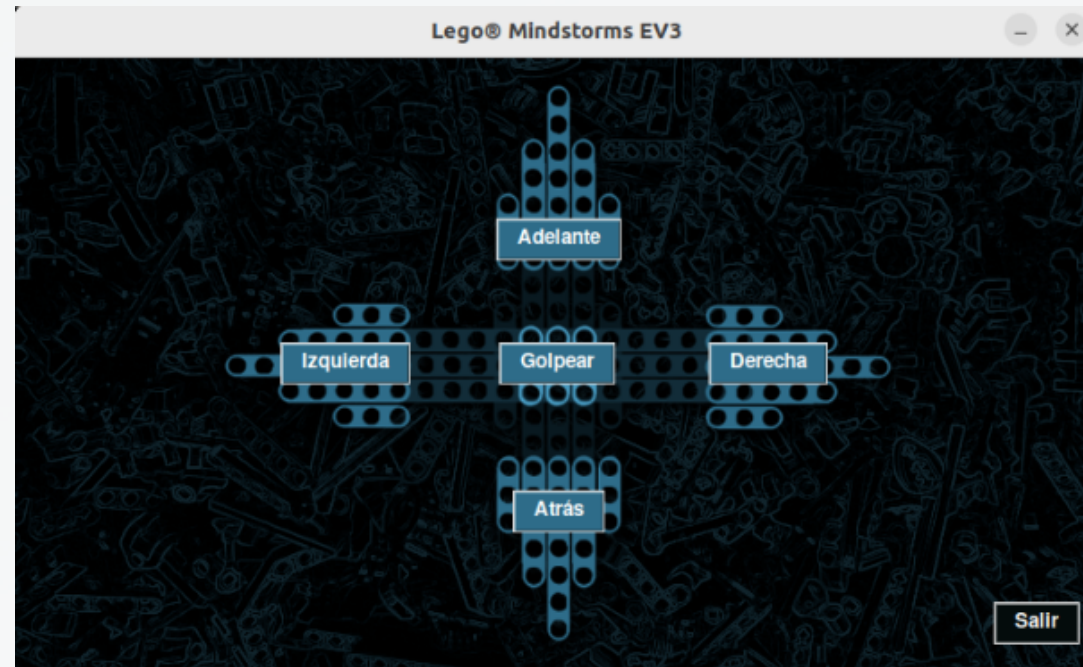
# ARQUITECTURA

Cliente



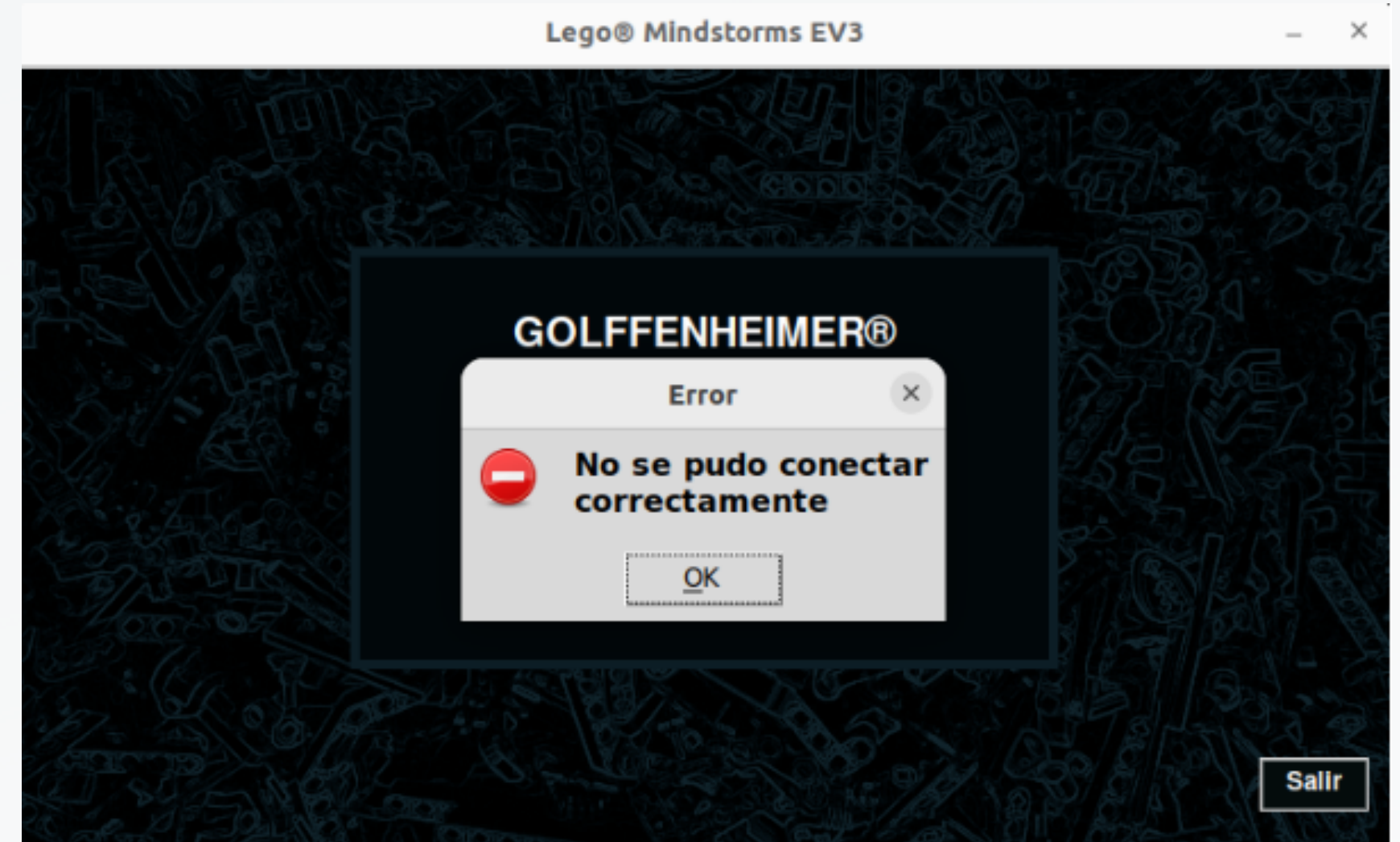
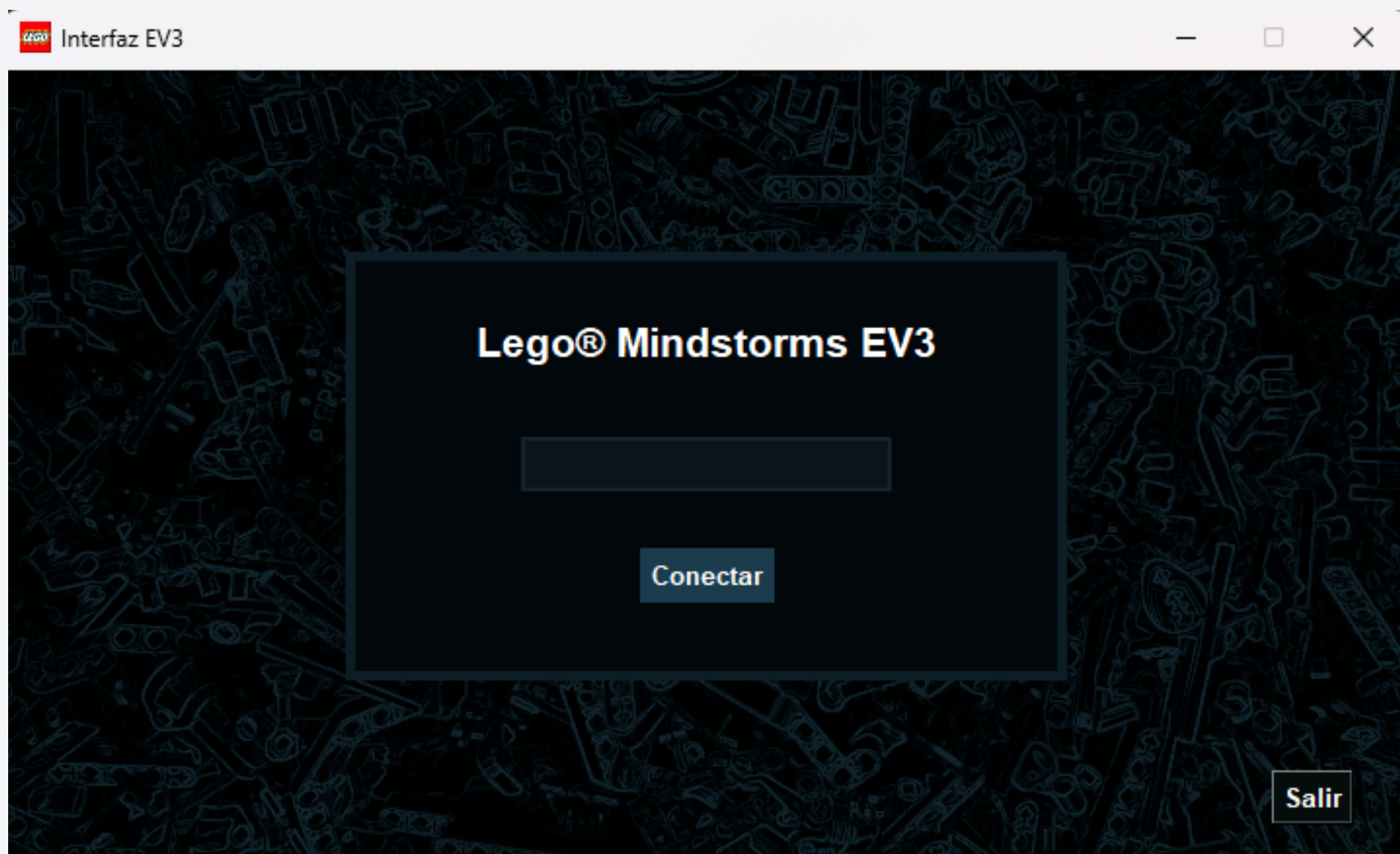
Conexión Wifi

Servidor



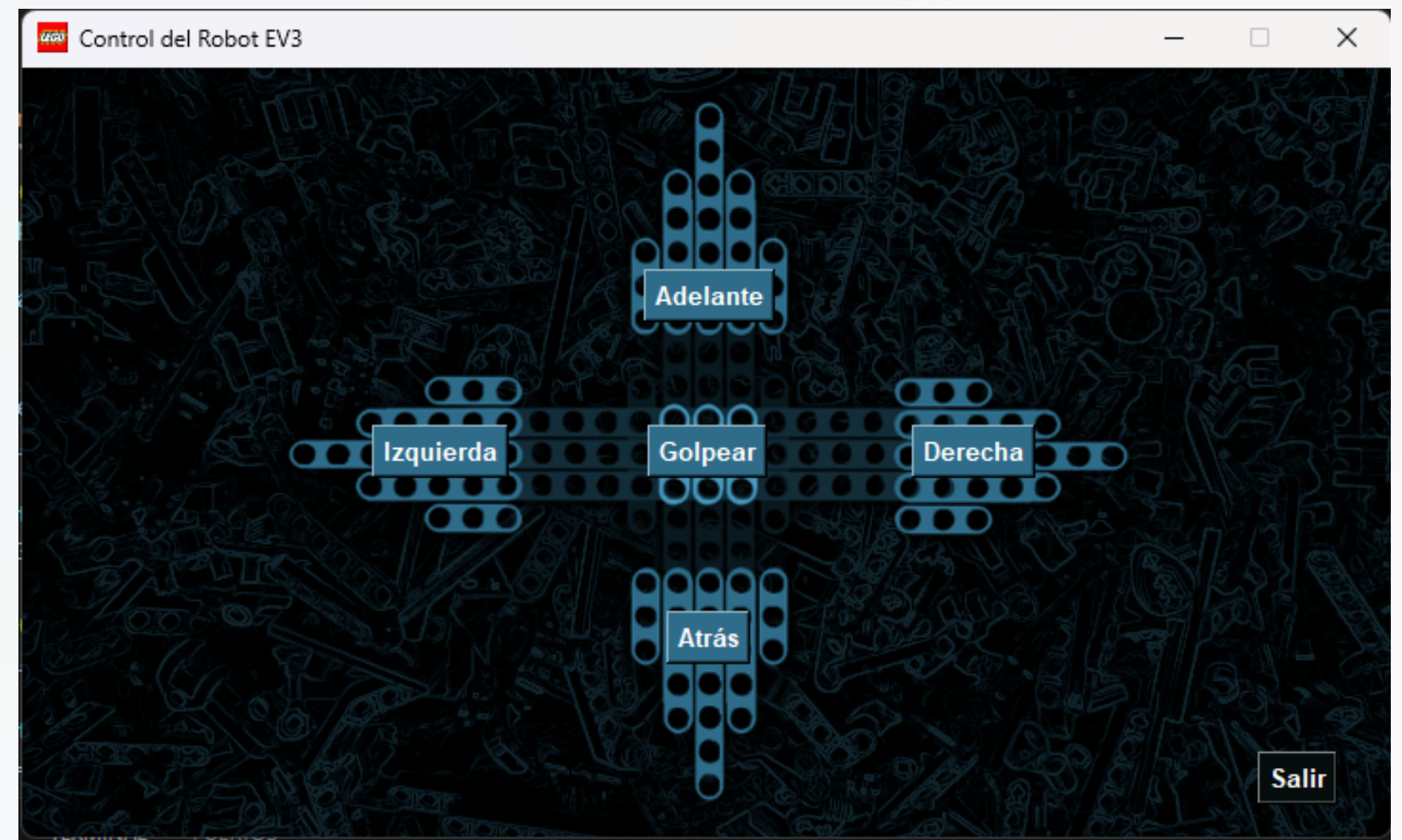
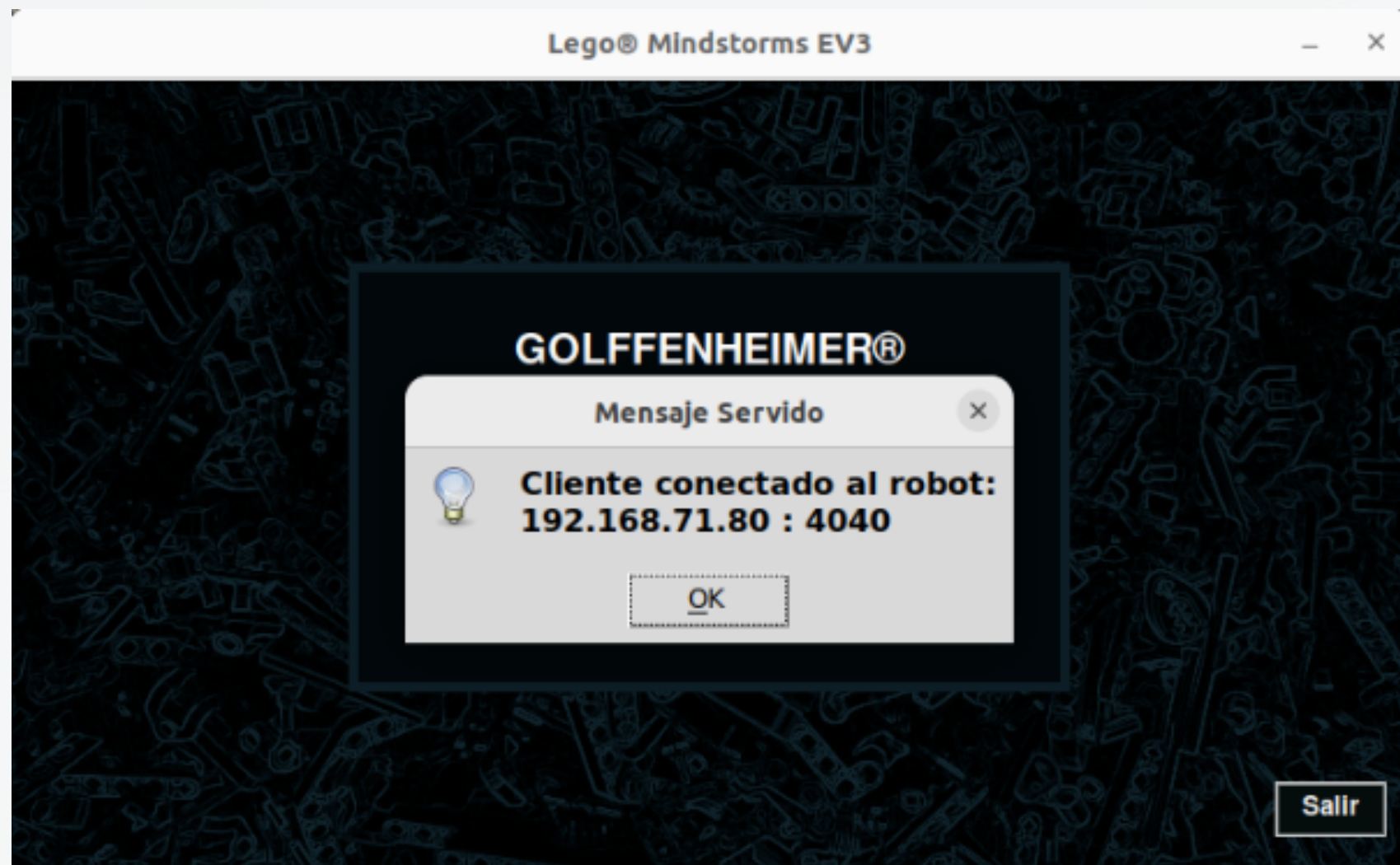


# INTERFAZ GRÁFICA





# INTERFAZ GRÁFICA



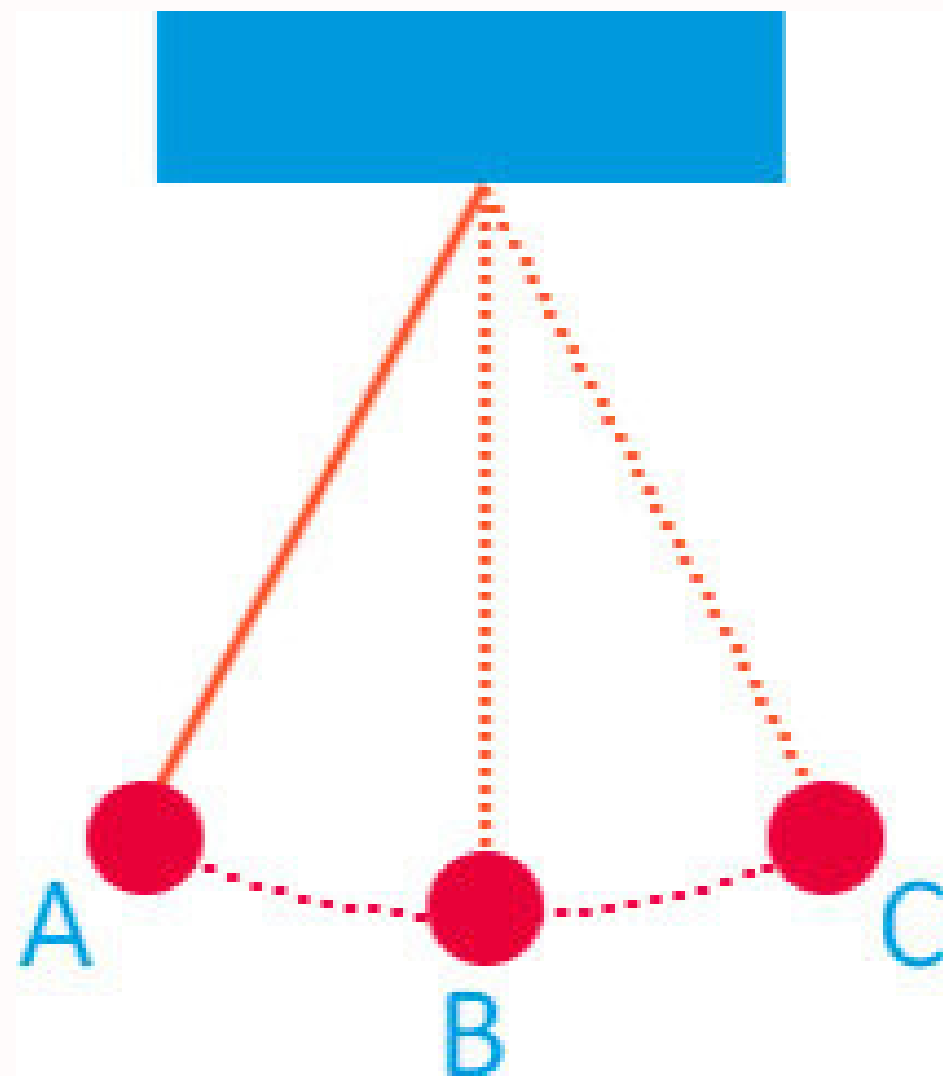


# IMPLEMENTACIÓN



# FUNDAMENTOS DE PROYECTILES.

## MOVIMIENTO DEL BRAZO DEL ROBOT



- Energía potencial:  
 $U = m \times g \times h$
- Energía cinética:  
 $K = \frac{1}{2} \times m \times v^2$

Datos:

$$h = 0.5m$$

$$g = 9.8 \frac{m}{s^2}$$

$$K_i + U_i = K_f + U_f$$

(i) Despejando  $v$  y reemplazando los datos:

$$K_i + U_i = K_f + U_f$$

$$U_i = K_f$$

$$m \times g \times h = \frac{1}{2} \times m \times v^2$$

$$v = \sqrt{2gh}$$

$$v = 3.1 \frac{m}{s}$$



Luego, se calcula la velocidad de la pelota al momento del impacto.

Datos:

$$m_b = 0.1 \text{ kg}$$

$$M_p = 0.003 \text{ kg}$$

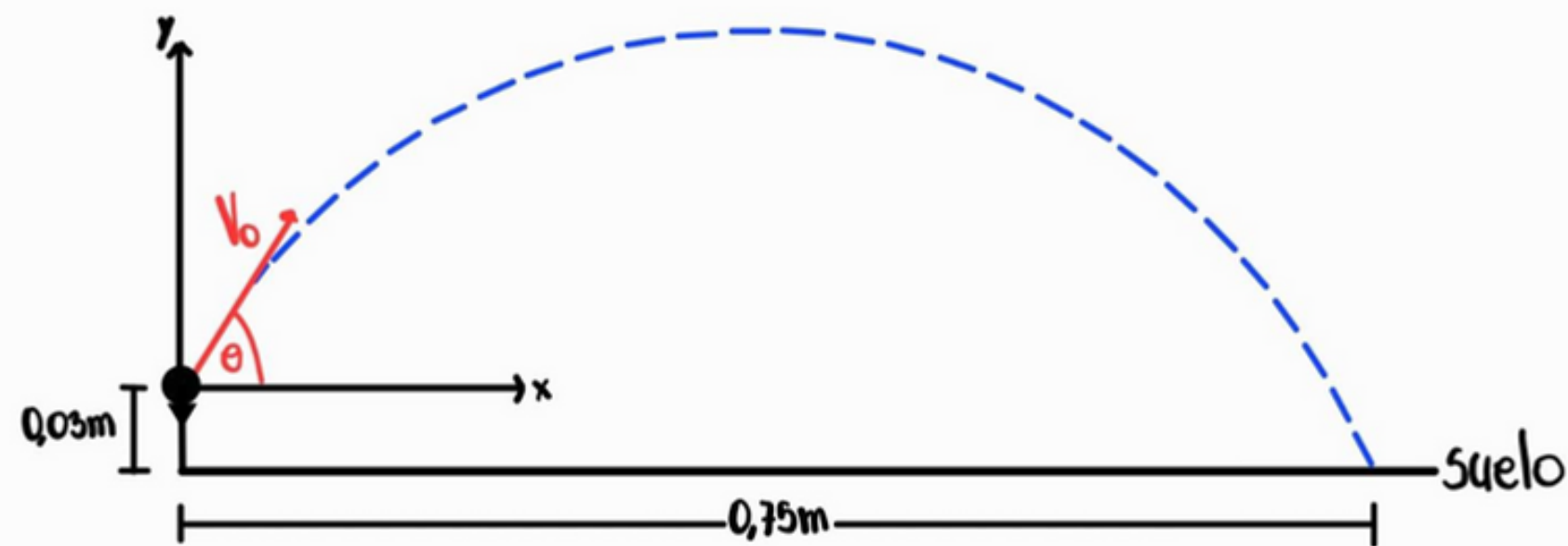
$$m_b \times v_{ib} + M_p \times v_{ip} = v(m_b + M_p)$$

$$m_b \times v_{ib} = v(m_b + M_p)$$

$$v = \frac{m_b \times v_{ib}}{(m_b + M_p)}$$

$$v = 3 \frac{\text{m}}{\text{s}}$$

## MOVIMIENTO DE LA PELOTA



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

$$x_f = x_0 + v_{0x} \times t$$

$$v_{0x} = v_0 \cos \theta = \text{cte}$$

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

$$y_f = y_0 + v_{0y} \times t + \frac{1}{2} \times g \times t^2$$

$$v_{0y} = v_0 \times \text{sen} \theta$$



Datos:

$$x_0 = 0m$$

$$y_0 = 0m$$

$$\theta = 45^\circ$$

$$x_f = 0.75m$$

$$y_f = -0.03m$$

$$v_0 = 3 \frac{m}{s}$$

$$g = -9.8 \frac{m}{s^2}$$

(i) Reemplazando y despejando  $t$ :

$$x_f = x_0 + v_{0x} \times t$$

$$0.75 = v_0 \cos 45^\circ \times t$$

$$t = \frac{0.75}{3 \times \cos 45^\circ}$$

$$t = \frac{0.75}{3 \times \cos 45^\circ}$$

$$t = 0.35s$$

El tiempo transcurrido desde que se golpea la pelota hasta que llega al suelo es de 0.35s

(ii) Reemplazando  $t$ :

$$y_f = y_0 + v_{0y} \times t + \frac{1}{2} \times g \times t^2$$

$$y_f = 3 \times \text{sen } 45^\circ \times 0.35 - 4.9 \times (0.35)^2$$

$$y_f = 3 \times \text{sen } 45^\circ \times 0.35 - 4.9 \times (0.35)^2$$

$$y_f = 0.14m$$

La altura máxima que alcanzará la pelota será de 0.14m cuando la velocidad inicial de la pelota es  $3 \frac{m}{s}$

# DESCRIPCIÓN DE LOS PROGRAMAS.

*Servidor*

```
1  import socket
2  from funciones import *
3
4  s = socket.socket()
5  print("Socket creado")
6  port = 4040
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
14 while True:
15     rawByte = connect.recv(1)
16     char = rawByte.decode('utf-8')
17     if (char == 'w'):
18         | adelante()
19     if (char == 'a'):
20         | izquierda()
21     if (char == 's'):
22         | atras()
23     if (char == 'd'):
24         | derecha()
25     if (char == 'r'):
26         | golpear()
27     if (char == ' '):
28         | stop()
```

# DESCRIPCIÓN DE LOS PROGRAMAS.

## Funciones

```
1  #!/usr/bin/env python3
2  from ev3dev2.motor import MediumMotor, OUTPUT_A, OUTPUT_C, OUTPUT_D, MoveTank, SpeedPercent
3
4  tank_drive = MoveTank(OUTPUT_A, OUTPUT_D)
5  stick = MediumMotor(OUTPUT_C)
6
7  def adelante():
8      print("Moving up...")
9      tank_drive.on(SpeedPercent(100), SpeedPercent(100))
10
11 def atras():
12     print("Moving down...")
13     tank_drive.on(SpeedPercent(-100), SpeedPercent(-100))
14
15 def derecha():
16     print("Moving right...")
17     tank_drive.on(SpeedPercent(-100), SpeedPercent(100))
18
19 def izquierda():
20     print("Moving left...")
21     tank_drive.on(SpeedPercent(100), SpeedPercent(-100))
22
23 def golpear():
24     stick.on_for_degrees(SpeedPercent(-100), 27000)
25
26 def stop():
27     tank_drive.stop()
```

# DESCRIPCIÓN DE LOS PROGRAMAS.

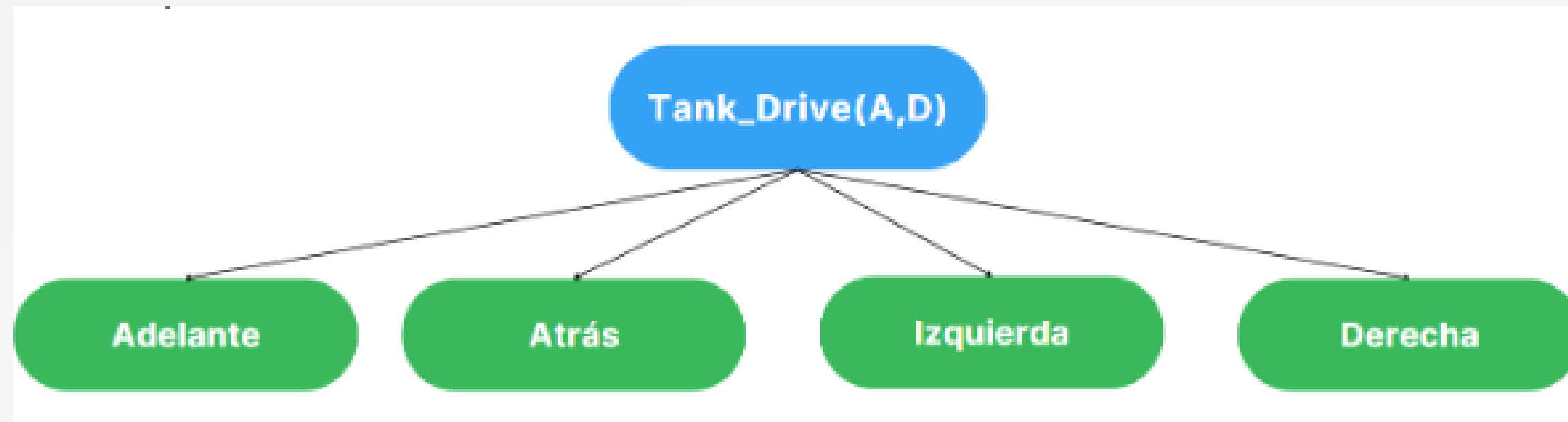
## Interfaz

```
1 import tkinter as tk
2 from tkinter import messagebox
3 import socket
4
5 def adelante():
6     | clientSocket.send(bytes([ord('w')]))
7
8 def atras():
9     | clientSocket.send(bytes([ord('s')]))
10
11 def derecha():
12     | clientSocket.send(bytes([ord('d')]))
13
14 def izquierda():
15     | clientSocket.send(bytes([ord('a')]))
16
17 def golpear():
18     | clientSocket.send(bytes([ord('r')]))
19
20 def on_release(event):
21     | clientSocket.send(bytes([ord(' ')]))
22
23 def abrirventana():
24     ip = entry.get()
25     try:
26         clientSocket.connect((ip,port))
27         messagebox.showinfo("Conexión exitosa","Cliente conectado al robot: {0} : {1}".format(ip,port))
28
29         ventana.destroy()
30         nueva_ventana = tk.Tk()
31         nueva_ventana.title("Lego® Mindstorms EV3")
32         nueva_ventana.geometry("720x405")
33         nueva_ventana.resizable(width=False, height=False)
34         #nueva_ventana.iconbitmap(icono)
35
36         canvas = tk.Canvas(nueva_ventana, width=720, height=405, highlightthickness=0)
37         canvas.pack()
```

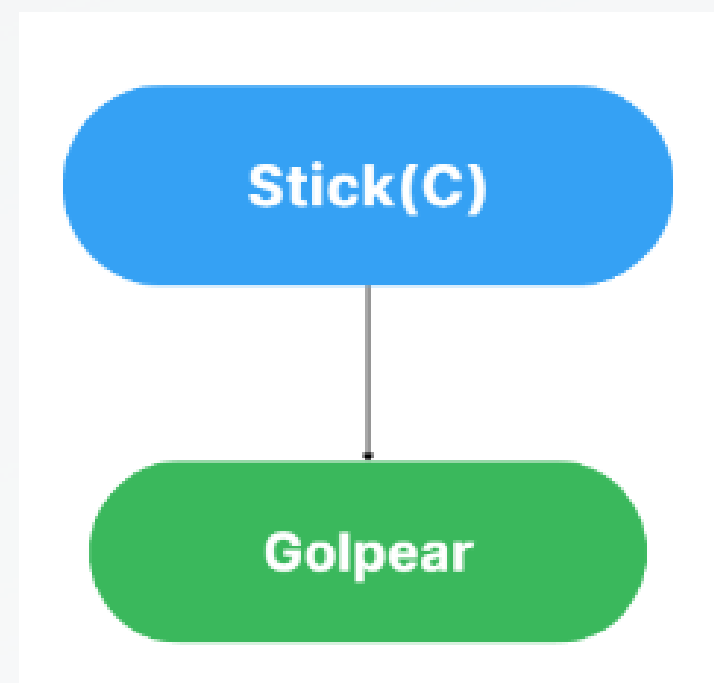
```
39     img = tk.PhotoImage(file="/home/proyecto1/Escritorio/interfaz/fondo2.png")
40     canvas.create_image(360, 202, anchor="center", image=img)
41
42     boton_adelante = tk.Button(nueva_ventana, text="Adelante", repeatdelay=50, repeatinterval=50, command=adelante, bg="#2e6c89", fg="white", font=("Helvetica", 10, "bold"))
43     boton_adelante.bind('<ButtonRelease-1>', on_release)
44     boton_adelante.place(x=361, y=120, anchor="center")
45
46     boton_atras = tk.Button(nueva_ventana, text="Atrás", repeatdelay=50, repeatinterval=50, command=atras, bg="#2e6c89", fg="white", font=("Helvetica", 10, "bold"))
47     boton_atras.bind('<ButtonRelease-1>', on_release)
48     boton_atras.place(x=361, y=300, anchor="center")
49
50     boton_izquierda = tk.Button(nueva_ventana, text="Izquierda", repeatdelay=50, repeatinterval=50, command=izquierda, bg="#2e6c89", fg="white", font=("Helvetica", 10, "bold"))
51     boton_izquierda.bind('<ButtonRelease-1>', on_release)
52     boton_izquierda.place(x=220, y=202, anchor="center")
53
54     boton_derecha = tk.Button(nueva_ventana, text="Derecha", repeatdelay=50, repeatinterval=50, command=derecha, bg="#2e6c89", fg="white", font=("Helvetica", 10, "bold"))
55     boton_derecha.bind('<ButtonRelease-1>', on_release)
56     boton_derecha.place(x=500, y=202, anchor="center")
57
58     boton_golpear = tk.Button(nueva_ventana, text="Golpear", command=golpear, bg="#2e6c89", fg="white", font=("Helvetica", 10, "bold"))
59     boton_golpear.place(x=360, y=202, anchor="center")
60
61     boton_salir = tk.Button(nueva_ventana, text="Salir", command=nueva_ventana.destroy, bg="#040c0c", fg="white", font=("Helvetica", 10, "bold"))
62     boton_salir.place(x=650, y=360)
63
64     nueva_ventana.mainloop()
65 except socket.error:
66     messagebox.showwarning("Conexión errónea", "No se ha logrado al conexión, verifique la ip {}".format(ip))
67     clientSocket.close()
68
69 ventana = tk.Tk()
70 ventana.title("Lego® Mindstorms EV3")
71 ventana.geometry("720x405")
72 ventana.resizable(width=False, height=False)
73 #icono = "home/proyecto1/Escritorio/interfaz/lego.ico"
74 #ventana.iconbitmap(icono)
```

```
76 canvas = tk.Canvas(ventana, width=720, height=405, highlightthickness=0)
77 canvas.pack()
78
79 img = tk.PhotoImage(file="/home/proyecto1/Escritorio/interfaz/fondo.png")
80 canvas.create_image(360, 202, anchor="center", image=img)
81
82 rect_width = 370
83 rect_height = 220
84 rect1 = canvas.create_rectangle(0, 0, rect_width, rect_height, fill="#0c1d25", outline="")
85 canvas.place(relx=0.5, rely=0.5, anchor="center")
86
87 canvas.coords(rect1, (canvas.winfo_reqwidth() - rect_width) / 2,
88                    (canvas.winfo_reqheight() - rect_height) / 2,
89                    (canvas.winfo_reqwidth() + rect_width) / 2,
90                    (canvas.winfo_reqheight() + rect_height) / 2)
91 rect_width = 360
92 rect_height = 210
93 rect2 = canvas.create_rectangle(0, 0, rect_width, rect_height, fill="#01070a", outline="")
94 canvas.place(relx=0.5, rely=0.5, anchor="center")
95 canvas.coords(rect2, (canvas.winfo_reqwidth() - rect_width) / 2,
96                    (canvas.winfo_reqheight() - rect_height) / 2,
97                    (canvas.winfo_reqwidth() + rect_width) / 2,
98                    (canvas.winfo_reqheight() + rect_height) / 2)
99
100 texto = canvas.create_text(360, 140, text="GOLFFENHEIMER®", font="Helvetica 16 bold", fill="white", anchor="center")
101
102 entry = tk.Entry(ventana, font=("Helvetica", 12), bg="#0c141b", fg="white", bd=2, relief="flat", highlightbackground="#15222e", highlightthickness=2)
103 entry.place(x=360, y=202.5, anchor="center")
104
105 boton = tk.Button(ventana, text="Conectar", command=abrirventana, bg="#1a3c4c", fg="white", font=("Helvetica", 10, "bold"), relief=tk.FLAT)
106 boton.place(x=360, y=260, anchor="center")
107
108 boton_salir = tk.Button(ventana, text="Salir", command=ventana.destroy, bg="#040c0c", fg="white", font=("Helvetica", 10, "bold"))
109 boton_salir.place(x=650, y=360)
110
111 clientSocket = socket.socket()
112 port = 4040
113
114 ventana.mainloop()
```

# DIAGRAMAS



A Y D SON LOS MOTORES CONECTADOS AL PUERTO OUTPUT\_A Y OUTPUT\_D RESPECTIVAMENTE.



C ES EL MOTOR CONECTADO AL PUERTO OUTPUT\_C.





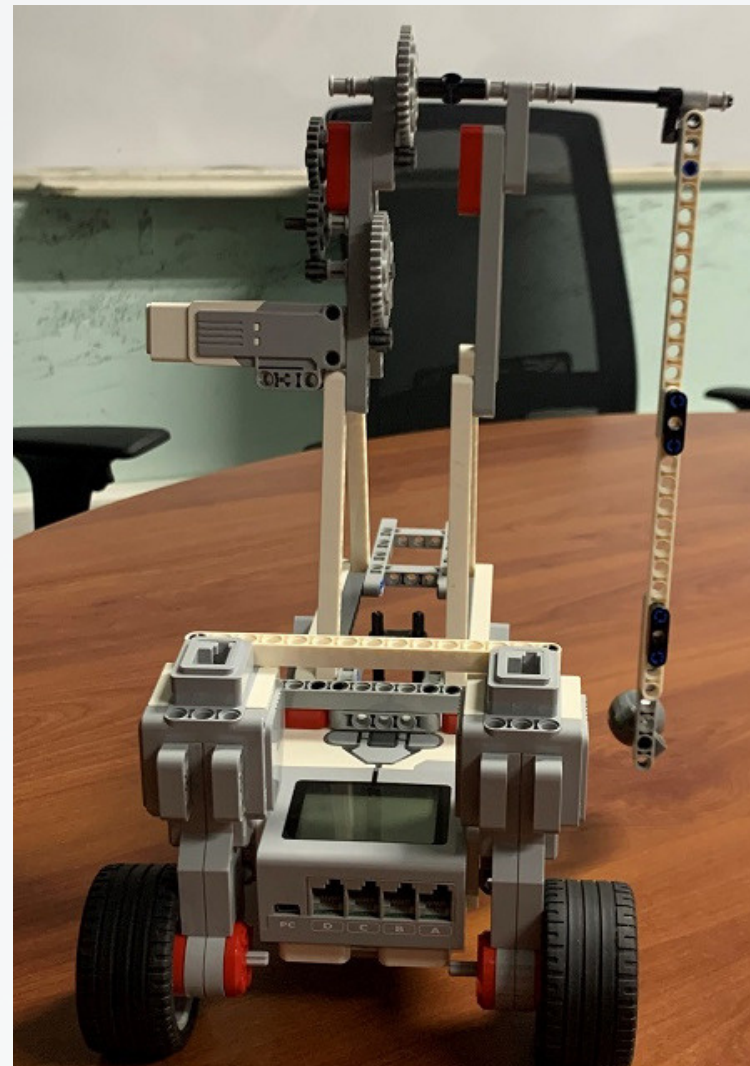
# RESULTADOS

# ESTADO ACTUAL DEL PROYECTO

01



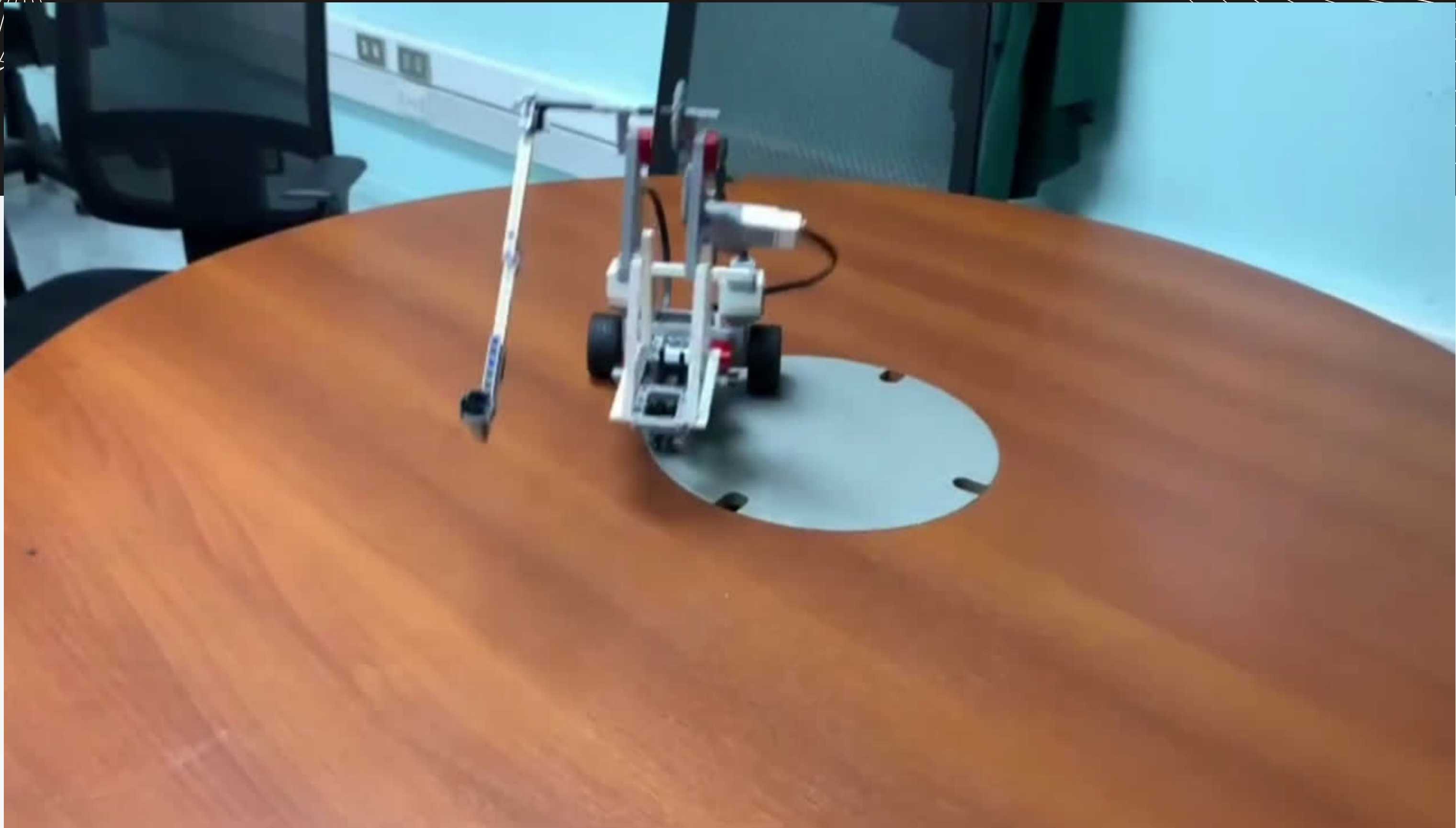
02



03







# PROBLEMAS ENCONTRADOS Y SOLUCIÓN PROPUESTA

## *Problemas*

1. Movimientos del robot.
2. Bloque EV3 no detecta los motores
3. El diseño del robot contaba con más de cuatro motores

## *Soluciones*

1. Se reinstaló el SO EV3Dev para limpiar la cantidad de archivos dentro del bloque, lo que provocaba el problema.
2. Se hicieron modificaciones al código para que fueran compatibles con nuestro diseño de robot.
3. Se rediseñó el robot para que ocupara dos motores grandes y uno mediano.



# CONCLUSIÓN

**GRACIAS POR SU ATENCIÓN**

