

**UNIVERSIDAD DE TARAPACÁ**



**FACULTAD DE INGENIERÍA**

**DEPARTAMENTO DE INGENIERÍA CIVIL EN COMPUTACIÓN E  
INFORMÁTICA**



**Plan de Proyecto  
“RobotFloo”**

**Alumno(os): René Ayca  
Claudio Carvajal  
Álvaro Lovera  
Giorgio Rojas  
Israel Tebes**

**Asignatura: Proyecto I**

**Profesor: Humberto Urrutia**

### Historial de Cambios

<b>Fecha</b>	<b>Versión</b>	<b>Descripción</b>	<b>Autor(es)</b>
05/11/2024	1.0	Formulación del Proyecto	René Ayca
11/11/2024	1.1	Ingreso de datos	Todos
13/11/2024	1.2	Revisión de datos	René Ayca
17/11/2024	1.3	Ultima revisión	Todos

## Tabla de Contenidos

<b>1. Panorama General</b>	4
1.1. <i>Introducción</i>	4
1.2. <i>Objetivos</i>	4
1.2.1. <i>Objetivo General</i>	4
1.2.2. <i>Objetivo Específico</i>	4
1.3. <i>Restricciones</i>	5
1.4. <i>Entregables</i>	5
<b>2. Organización del Personal</b>	6
2.1. <i>Descripción de los Roles</i>	6
2.2. <i>Personal que cumplirá los Roles</i>	6
2.3. <i>Mecanismos de Comunicación</i>	6
<b>3. Planificación del Proyecto</b>	7
3.1. <i>Actividades</i>	7
3.2. <i>Asignación de Tiempo</i>	8
3.3. <i>Gestión de Riesgos</i>	9
<b>4. Planificación de los Recursos</b>	9
4.1. <i>Hardware</i>	9
4.2. <i>Software</i>	9
4.3. <i>Estimación de Costos</i>	10
<b>5. Análisis y Diseño</b>	11
5.1. <i>Especificación de Requerimiento</i>	11
5.2. <i>Arquitectura</i>	11
5.3. <i>Interfaz</i>	12
<b>6. Implementación</b>	14
6.1. <i>Fundamentos Físicos</i>	14
6.2. <i>Descripción de los programas</i>	16
6.3. <i>Diagramas</i>	21
<b>7. Resultados</b>	21
7.1. <i>Estado Actual del Proyecto</i>	21
7.2. <i>Problemas Encontrados y Solución Propuesta</i>	22
<b>8. Conclusión</b>	22
<b>9. Referencias</b>	23

## 1. Panorama General

### 1.1. *Introducción*

Estos informes están dedicados a recopilar la organización y procedimientos que se llevarán a cabo en la asignatura proyecto 1, donde en equipo a través del robot Ev3 Mindstorm y con un computador de sistema operativo Linux se tendrá que programar a través del lenguaje Python sus funcionamientos, así también se creará una arquitectura para el robot que cumpla las necesidades.

En este semestre el robot deberá cumplir con trasladar bolas de ping pong, donde la arquitectura del robot propuesta deberá recoger las bolas del suelo y cargarlas para descargarlas a un diferente sector.

Este primer informe se centrará en la organización, donde se mostrará los roles asignados a los integrantes del equipo, los horarios de trabajo y recursos utilizados.

### 1.2. *Objetivos*

#### 1.2.1. Objetivo General

Desarrollar y programar un robot que pueda trasladar bolas de ping pong desde un lugar a otro

#### 1.2.2. Objetivo Específico

- Aprender a construir con el Set de Lego Mindstorms Ev3 para la creación del robot.
- Proponer arquitecturas que cumplan con las necesidades del robot.
- Armar el robot de acuerdo a la arquitectura propuesta.
- Investigar y aprender sobre los import que ofrece Python acerca del Ev3 Mindstorm.
- Programar una interfaz con Tkinter Python para mandar instrucciones al Ev3 Mindstorm.

## Proyecto I Plan de Proyecto Avance

### 1.3. *Restricciones*

- ◊ Solo se programa en Python.
- ◊ Se necesita hacer todo en el sistema operativo Linux.
- ◊ Solo se debe utilizar la plataforma Redmine para los documentos y avance del proyecto.
- ◊ Se debe utilizar el Set de Lego Mindstorms EV3.
- ◊ Solo una hora y cuarenta minutos tres veces al día para avanzar en equipo.
- ◊ Disponibilidad de tiempo de los integrantes del equipo.
- ◊ Cantidad de integrantes limitada a solo 5.
- ◊ Disponibilidad del robot para codificar y probar.

### 1.4. *Entregables*

**Bitácoras:** Registro que documenta avances, problemas y soluciones a lo largo del desarrollo de un proyecto.

**Carta Gantt:** Una herramienta gráfica utilizada para la planificación y gestión de proyectos. Consiste en una tabla que visualiza las tareas o actividades de un proyecto a lo largo de una línea de tiempo, mostrando el inicio, la duración y el final de cada tarea mediante barras horizontales.

**Informe de Formulación:** Documento que describe detalladamente el proceso de creación y planificación de un proyecto antes de su ejecución. Su objetivo es presentar un análisis estructurado de cómo se va a llevar a cabo dicho proyecto, cuáles son sus objetivos, recursos necesarios, metodologías, cronogramas, y presupuestos.

**Manual de Usuario:** Documento que proporciona instrucciones detalladas sobre el uso, operación y mantenimiento de un producto, Está diseñado para ayudar a los usuarios a comprender cómo utilizar adecuadamente un producto, resolver problemas comunes y maximizar su funcionalidad.

**Presentaciones:** Exposición estructurada de información, ideas o propuestas, que se realiza con el objetivo de comunicar un mensaje claro a una audiencia.

## 2. Organización del Personal

En esta sección se mostrará la organización de los roles que van a ejercer los integrantes del equipo, Así también el método el cual nos comunicaremos para el desarrollo del proyecto.

### 2.1. Descripción de los Roles

Jefe de proyecto: Responsable de planificar, ejecutar y supervisar un proyecto desde su inicio hasta su finalización.

Ensamblador: Responsable del armado del robot, gestiona piezas y modelo congruente a armar.

Programador: Responsable de la programación sobre los funcionamientos del robot, resuelve errores e ideas a implementar al robot.

Documentador: Responsable de a través de un informe transcribir los procesos que se llevan a cabo en el proyecto

Diseñador: Responsable del logo y diseños que se llevan a cabo del proyecto.

### 2.2. Personal que cumplirá los Roles

Rol	Responsable	Involucrados
Jefe de proyecto	René Ayca	René Ayca
Ensamblador	Claudio Carvajal	Claudio Carvajal Álvaro Lovera Israel Tebes
Diseñador	Álvaro Lovera	Álvaro Lovera Claudio Carvajal
Programador	René Ayca	René Ayca Giorgio Rojas
Documentador	René Ayca	René Ayca Israel Tebes

### 2.3. Mecanismos de Comunicación

Utilizaremos los siguientes medios de comunicación principales: WhatsApp, para la mensajería, aprovechando los grupos que permite crear; y Discord, que se usará para las reuniones, utilizando tanto sus canales de texto como de voz.

### 3. Planificación del Proyecto

#### 3.1. Actividades

Nombre	Descripción	Responsable	Producto
Trabajar en el primer diseño del robot	Se inicia un primer diseño piloto para ir conociendo sobre una posible estructura del robot	Claudio Carvajal Álvaro Lovera Israel Tebes	Prototipo inicial del robot
Crear Bitácoras Etapa 1	Se crearán bitácoras de la etapa N°1 sobre la organización del equipo y armado del robot	René Ayca	Prototipo inicial del robot
Creación de un nuevo diseño del robot y generar fotos del avance	Se desarma el prototipo anterior y se crea un nuevo diseño del robot, probando la movilidad y se tomas fotos del avance	Claudio Carvajal Álvaro Lovera René Ayca Israel Tebes	Se termina el modelo de robot 1.1
Trabajar en la preparación y programación del robot y avance del diseño del robot	Se conecta el robot a internet y se empiezan a preparar códigos para comprobar si el robot responde a las instrucciones de programación y se sigue avanzando en el diseño del robot	René Ayca Claudio Carvajal Álvaro Lovera Israel Tebes	Se termina el modelo 1.2 del robot
Supervisar avance en la carta Gantt y se continúa con el armado del robot	Se empieza con los detalles de la carta Gantt y el resto continúa armando el robot y creando una garra para agarrar objetos	Rene Ayca Claudio Carvajal Álvaro Lovera Israel Tebes	Se termina el modelo 1.3 del robot.
Crear Bitácoras Etapa 2	Se crearán bitácoras de la etapa N°2 sobre el avance del equipo y programación del	Rene Ayca	Modelo de robot 1.3

## Proyecto I Plan de Proyecto Avance

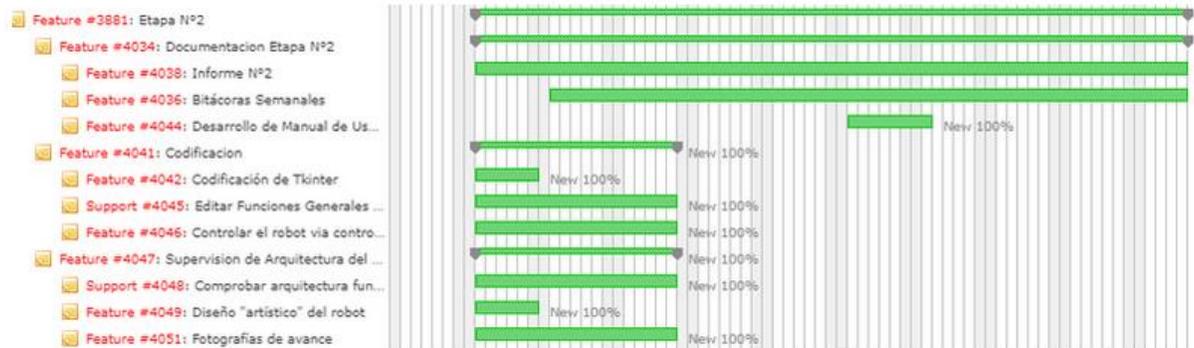
	<b>robot</b>		
Se finaliza el armado de robot	Se termina el diseño del robot y la garra, se comprueba la movilidad del robot, dándolo por finalizado y solo se mejoraran detalles	Claudio Carvajal Álvaro Lovera Israel Tebes	Término del modelo del robot
Crear Bitácora Etapa 3	Se crearán bitácoras de la etapa N°3 sobre el avance del proyecto y dando término a las actividades	Rene Ayca	Modelo de robot finalizado

### 3.2. Asignación de Tiempo

Los tiempos asignados están establecidos por etapas del proyecto, siendo todos estos los avanzados en clases.

Si es que se está atrasado en alguna sección se propone ir los sábados a terminar el avance según fecha establecida por Carta Gantt

## Carta Gantt



## Proyecto I Plan de Proyecto Avance

### 3.3. *Gestión de Riesgos*

A continuación, se presentará una tabla con los obstáculos que se ha enfrentado el proyecto en sus etapas iniciales. Los niveles en los cuales se resumirá el impacto de riesgo, se dividirán en cuatro tipos de daños:

1. Daño catastrófico: Las medidas a tomar en el caso son de forma inmediata, puede provocar que el proyecto se detenga indefinidamente.
2. Daño crítico: Se deben tomar medidas necesarias para resolver el riesgo, debido a que puede provocar que el proyecto se retrase en varias etapas.
3. Daño circunstancial: El riesgo se debe resolver en el momento, debido a que puede retrasar el desarrollo de una etapa base del proyecto.
4. Daño irrelevante: El riesgo no es de mayor importancia, es un detalle imprevisto que no necesita mucha atención y se puede resolver en cualquier momento.

## 4. **Planificación de los Recursos**

### 4.1. *Hardware*

- Tarjeta microSD
- Set lego Ev3 Mindstorm
- Notebooks
- Wifi dongle
- Adaptador microSD

### 4.2. *Software*

- Visual Studio Code
- Discord
- Linux (Ubuntu)
- Canva
- EV3 dev (ev3dev.org)
- WhatsApp
- Python

## Proyecto I Plan de Proyecto Avance

### 4.3. Estimación de Costos

#### Costo de Hardware:

Producto	Precio
Set Lego Mindstorm(EV3)	\$ 816.00
HP / Intel(R) core(tm) i5-10210u cpu @ 1.60ghz	\$799.990
HP / NOTEBOOK GAMER HP VICTUS 15-FA1013LA INTEL CORE I7 16 GB	\$ 1.200.000
HP / Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz	\$ 700.000
Notebook Toshiba Tecra Z40 C1410LA P/N PT463U-07P01Y	\$ 899.990
wifi dongle	\$5.200
Piezas extra	\$ 120.000
Micro SD	\$ 9.054
Adaptador microSD	\$ 2.000
<b>Total :</b>	<b>\$ 4.552.234</b>

#### Costo de Software:

Producto	Precio
Licencia de Canva / 5 meses	\$ 39.500
Licencia Microsoft Office	Gratis
Plan de internet (entel)	\$ 71.358
<b>Total :</b>	<b>\$ 110.858</b>

#### Costo de Trabajador:

Rol	Personas (lideres)	Valor horas trabajadas	Horas	Costo Final (por 5 meses)
Jefe de proyecto	1	5.000	30	750.000
Programador	1	3.000	30	450.000
Ensamblador	1	2.500	30	375.000
Diseñador	1	1.200	30	180.000
Documentador	1	2.200	30	330.000
<b>Total:</b>				<b>2.085.000</b>

Total de Costo:

Costo Hardware	\$ 4.552.234
Costo Software	\$ 110.858
Costo Empleados	\$ 2.085.000
Total :	\$ 6.748.092

## 5. Análisis y Diseño

### 5.1. Especificación de Requerimiento

Requerimientos funcionales:

- El robot y la interfaz gráfica se pueden conectar mediante la ip del robot.
- La interfaz gráfica tiene botones para que el robot pueda realizar sus funciones como: desplazarse, agarrar y soltar la bola.
- El robot a través de un mando de Wii puede realizar sus funciones de trabajo.

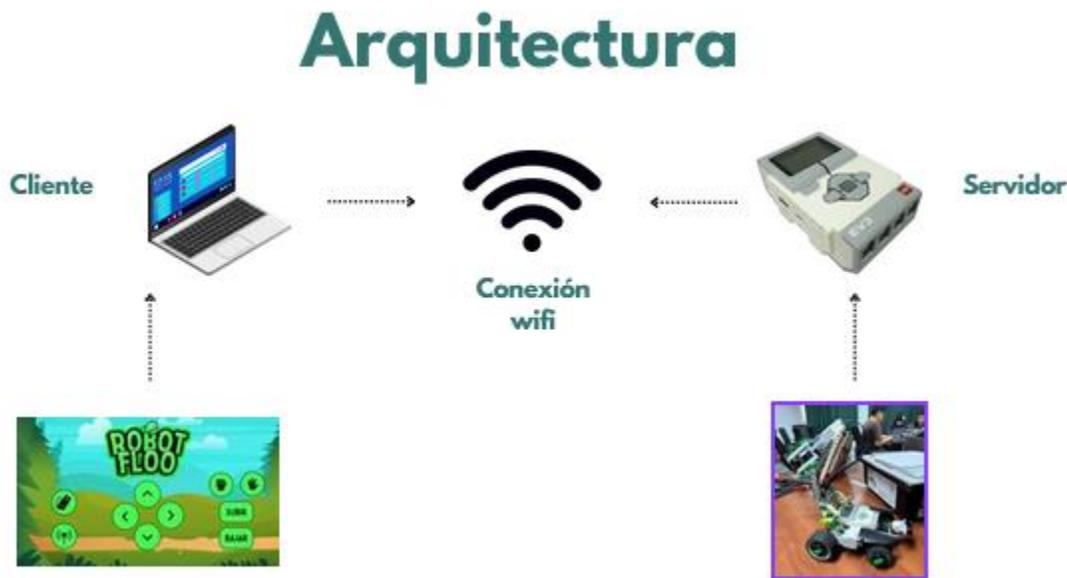
Requerimientos no funcionales:

- A través de un socket el robot puede realizar un servidor donde se puede comunicar con el cliente (usuario).
- La interfaz gráfica manda las instrucciones generadas por la interfaz al robot para que realice su función seleccionada.
- Se desarrolla a través de Python los códigos principales del robot como el servidor, las funciones y la interfaz gráfica.

### 5.2. Arquitectura

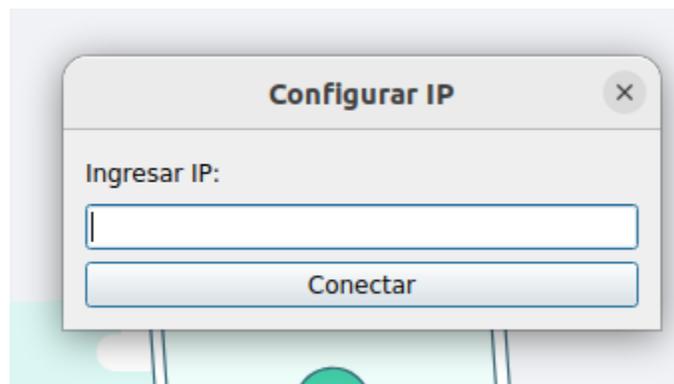
1. Ambos dispositivos deben estar conectados a la misma red wifi para establecer la comunicación.
2. Encargado de la conexión remota del cliente con el robot el cual permanece en espera de alguna conexión entrante.
3. Mediante una PC la interfaz se conectará al servidor del robot y el usuario podrá controlarlo.

4. El robot ejecuta la acción recibida por parte del usuario y procede a realizarla de acuerdo con las instrucciones recibidas.
5. Interfaz gráfica que el usuario usará para controlar al robot.



### 5.3. Interfaz

Al iniciar el programa de conectividad entre servidor del robot y el cliente se abre esta primera ventana donde el usuario ingresaría la ip correspondiente del robot para que este puede transmitir las señales de movimiento que se le indique. Para ingresar a la siguiente ventana el usuario tendría que presionar el botón conectar donde el programa si la conexión fue exitosa ira directamente a la ventana de funcionamiento, si no, esta mostrara un mensaje de error de conexión



## Proyecto I Plan de Proyecto Avance

Esta es la ventana de movimientos generales del robot ambientada a su temática la cual se resaltan colores verdes.

En esta ventana habrá 10 botones las cuales los 4 de al medio serán los que den movilidad al robot llamando a las funciones de movimiento:

- Adelante()
- Atrás()
- Izquierda()
- Derecha()

Estos tienen la particularidad que normalmente se espera de un control, de que al soltar el botón de la interfaz (ya no lo está presionando) este se detiene automáticamente.

También para el lado derecho hay 4 botones de acciones del robot que permiten funciones como agarrar, soltar, subir y bajar, ya que nuestro robot está dedicado a realizar estas acciones de uso. Se llama a su funcionamiento a través de:

- Subir()
- Bajar()
- Agarrar()
- Soltar()

Para el lado izquierdo están las conexiones del robot donde habrá 2 botones, uno permite el manejo del robot a través de un mando de Wii y el otro transmite música ambientada en la temática. Estos son llamados a funcionar a través de:

- Control\_Wii()
- Music()



## 6. Implementación

### 6.1. Fundamentos Físicos

Para este proyecto lo mas importante es poder ejecutar la función principal del robot, el cual es poder mover una bola de ping pong de un lugar a otro.

Por lo que se decidió investigar mediante nuestra arquitectura cual es el torque ejercido y cual es el peso que no podríamos levantar.

Datos del Robot:

- Peso total = 0,261 kg
- Gravedad = 9.8 m/s<sup>2</sup>
- Largo del brazo = 0,105 m
- Grados de abertura (1): 130°
- Grados de abertura (2): 45°
- Bola: 0,27 kg

Ocuparemos el peso total que es el peso del brazo, el de la bola, la gravedad y el largo del brazo

Para encontrar el torque ocuparemos esta fórmula de torque donde:

- $b$  : es el largo del brazo del robot
- $F$  : es la fuerza ejercida
- $T$  : que tan fácil es girar el cuerpo mediante una fuerza

$$\tau = bF$$

Remplazaremos  $F$  por  $W = m(\text{peso}) \times g(\text{gravedad})$

También se sabe a través del manual del robot de ejercen los siguientes torques

Especificaciones del Large Motor:

- Velocidad: 160–170 RPM (revoluciones por minuto).
- Torque en funcionamiento: 20 N·cm.
- Torque de bloqueo (stall torque): 40 N·cm.

Especificaciones del Medium Motor:

- Velocidad: 240–250 RPM (revoluciones por minuto).
- Torque en funcionamiento: 8 N·cm.
- Torque de bloqueo (stall torque): 12 N·cm.

## Proyecto I Plan de Proyecto Avance

Para empezar el calculo primero debemos de calcular las Fuerzas ejercidas a través del peso del brazo y la pelota:

$$W1 = 0,347 \times 9,8 = 3.401$$

$$W2 = 0,27 \times 9,8 = 2.646$$

Después remplazaremos todo en la formula para ver si el torque ejercido será menor a 8N (ya que utilizamos el médium motor), lo cual demostraría que el brazo del robot si podrá alzarlo.

$$T = (W1 + W2) \times 0.105$$

$$T = 6,047 \times 0,105 = 0,635 \text{ N cm}$$

Como el torque calculado  $T = 6,35$  es menor al torque funcional  $T = 8$ , el brazo del robot es capaz de levantar la bola de ping pong

Ósea que entre mas peso tenga el objeto levantado, habrá un mayor acercamiento al limite que nos permite poder subir objetos

## Proyecto I Plan de Proyecto Avance

### 6.2. Descripción de los programas

Servidor:

El servidor del programa fue hecho por medio del lenguaje de programación Python, donde mediante de la importación socket se pudo hacer un servidor donde el cliente podrá conectarse con el robot para dirigir funciones, estas funciones señalas irán a otro programa de funciones accedida gracias a la importación del programa de funciones.

```
1 import socket
2 from control 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     if (char == 'd'):
21         moveRight()
22     if (char == 'a'):
23         moveLeft()
```

```
24     if (char == 'i'):
25         getUp()
26     if (char == 'o'):
27         getDown()
28     if (char == 'k'):
29         catch()
30     if (char == 'l'):|
31         drop()
32     if (char == 'q'):
33         print("Terminando la sesion...")
34         break
35     if (char == ' '):
36         stop()
37     if (char == 'm'):
38         music()
```

## Proyecto I Plan de Proyecto Avance

### Funciones:

Para el funcionamiento del robot se importaron las librerías de ev3 para poder utilizar correctamente sus componentes, siendo los motores las mas importantes para el funcionamiento general del robot.

```
1 from ev3dev2.motor import LargeMotor, MediumMotor, OUTPUT_A, OUTPUT_B, OUTPUT_C$
2 import time
3 from ev3dev2.sound import Sound
4
5 ma = LargeMotor(OUTPUT_A)
6 mb = LargeMotor(OUTPUT_B)
7 mc = MediumMotor(OUTPUT_C)
8 md = MediumMotor(OUTPUT_D)
9 tankmoves = MoveTank(OUTPUT_A,OUTPUT_B)
10 def moveUp():
11     print("Moving up...")
12     tankmoves.on(50,50)
13 def moveDown():
14     print("Moving down...")
15     tankmoves.on(-50,-50)
16 def moveRight():
17     print("Moving right...")
18     tankmoves.on(50,0)
19 def moveLeft():
20     print("Moving left...")
21     tankmoves.on(0,50)
22 def getUp():
23     print("upping")
24     mc.on(SpeedPercent(25))
```

```
25     time.sleep(1)
26     mc.off()
27 def getDown():
28     print("dowwing")
29     mc.on(SpeedPercent(-25))
30 time.sleep(1)
31 mc.off()
32 def catch():
33     print("caching")
34     md.on(SpeedPercent(9))
35     time.sleep(1)
36     md.off()
37 def drop():
38     print("dropping")
39     md.on(SpeedPercent(-9))
40 time.sleep(1)
41 md.off()
42 def stop():
43     print("Stopping...")
44     tankmoves.off()
45 def music():
46     print("danceeee")
47     sound = Sound()
48     sound.play_file('/home/robot/fernan_arreglado.wav')|
```

## Proyecto I Plan de Proyecto Avance

### Interfaz:

Para la interfaz de control del robot de utilizo la interfaz grafica llamada PyQt5 ya que esta nos permitía la creación de botones redondos, además de un manejo mas amplio de las imágenes.

Esta interfaz permite la conexión al robot a través de la ip de este donde el servidor creado por el socket se pueden mandar funciones desde la interfaz al robot

```
1 import sys
2 import socket
3 import threading
4 import evdev
5 from PyQt5.QtWidgets import QApplication, QLabel, QLineEdit, QPushButton, QVBoxLayout, QDialog, QMessageBox, QMainWindow
6 from PyQt5.QtGui import QPixmap
7
8 # Crear socket del cliente global
9 clientSocket = None
10 main_window = None # Declarar la ventana principal como global
11
12 def enviar_comando(comando):
13     global clientSocket
14     try:
15         clientSocket.send(bytes([ord(comando)]))
16     except (BrokenPipeError, OSError):
17         QMessageBox.warning(None, "Error de conexión", "Parece que la conexión con el robot se ha perdido.")
18         clientSocket.close()
19
20 # Comandos de movimiento
21 def adelante(): enviar_comando('w')
22 def atras(): enviar_comando('s')
23 def derecha(): enviar_comando('d')
24 def izquierda(): enviar_comando('a')
25 def Subir(): enviar_comando('o')
26 def Bajar(): enviar_comando('i')
27 def Agarrar(): enviar_comando('k')
28 def Soltar(): enviar_comando('l')
29 def Musica(): enviar_comando('m')
30 def detener(): enviar_comando(' ') # Enviar un espacio para detener
```

```
31
32 def wii_control():
33     try:
34         device = evdev.InputDevice('/dev/input/event15') # Asegúrate de que esta ruta sea la correcta
35         print(f"Dispositivo conectado: {device.name}")
36
37         for event in device.read_loop():
38             if event.type == evdev.ecodes.EV_KEY:
39                 if event.value == 1: # Botón presionado
40                     # Lógica de control del Wii Remote
41                     if event.code == evdev.ecodes.BTN_A: Subir()
42                     elif event.code == evdev.ecodes.BTN_B: Bajar()
43                     elif event.code == evdev.ecodes.KEY_LEFT: izquierda()
44                     elif event.code == evdev.ecodes.KEY_RIGHT: derecha()
45                     elif event.code == evdev.ecodes.KEY_UP: adelante()
46                     elif event.code == evdev.ecodes.KEY_DOWN: atras()
47                     elif event.code == evdev.ecodes.BTN_2: Agarrar()
48                     elif event.code == evdev.ecodes.BTN_1: Soltar()
49                     elif event.code == evdev.ecodes.BTN_MODE: Musica()
50                 elif event.value == 0: # Botón soltado
51                     detener()
52     except OSError as e:
53         print(f"Error al conectar con el Wii Remote: {e}")
54         QMessageBox.critical(None, "Error", "No se pudo conectar al Wii Remote.")
55
56 def activar_wii_control():
57     t = threading.Thread(target=wii_control, daemon=True)
58     t.start()
59
```

## Proyecto I Plan de Proyecto Avance

```
60 def show_ip_window():
61     global clientSocket # Usar el socket global
62     # Crear ventana de configuración de IP
63     ip_window = QDialog()
64     ip_window.setGeometry(100, 100, 300, 100)
65     ip_window.setWindowTitle("Configurar IP")
66
67     # Crear widgets
68     ip_label = QLabel("Ingresar IP:")
69     ip_entry = QLineEdit(ip_window)
70
71     connect_button = QPushButton("Conectar", ip_window)
72
73     # Configurar layout
74     layout = QVBoxLayout()
75     layout.addWidget(ip_label)
76     layout.addWidget(ip_entry)
77     layout.addWidget(connect_button)
78     ip_window.setLayout(layout)
79
80     # Establecer el puerto como una variable interna
81     port = 8080 # Puerto predeterminado
82
83     def connect():
84         global clientSocket # Usar el socket global
85         address = ip_entry.text()
86         try:
87             conectar(address, port, ip_window)
88         except ValueError:
89             QMessageBox.warning(ip_window, "Error", "Por favor, ingrese una dirección IP válida.")
90
91     connect_button.clicked.connect(connect)
92     ip_window.exec_() # Mostrar la ventana de configuración de IP
93
94 def conectar(address, port, ip_window):
95     global clientSocket # Usar el socket global
96     # Crear socket del cliente
97     clientSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
98     try:
99         clientSocket.connect((address, port))
100        QMessageBox.information(ip_window, "Mensaje", f"Cliente conectado al robot: {address}:{port}")
101        ip_window.accept() # Aceptar la conexión y cerrar la ventana de IP
102
103        # Intentar mostrar la ventana principal
104        show_main_window() # Llamada a la ventana principal después de la conexión
105
106    except socket.error:
107        QMessageBox.warning(ip_window, "Conexión fallida", f"No se ha logrado la conexión, verifique la IP {address}")
108
109 def show_main_window():
110     global main_window # Definir main_window como global
111     main_window = QMainWindow()
112     main_window.setGeometry(100, 100, 1200, 675)
113     main_window.setWindowTitle("Botones Circulares y Ovalados con Fondo")
114
115     # Crear un widget central
116     central_widget = QWidget(main_window)
117     main_window.setCentralWidget(central_widget)
```

## Proyecto I Plan de Proyecto Avance

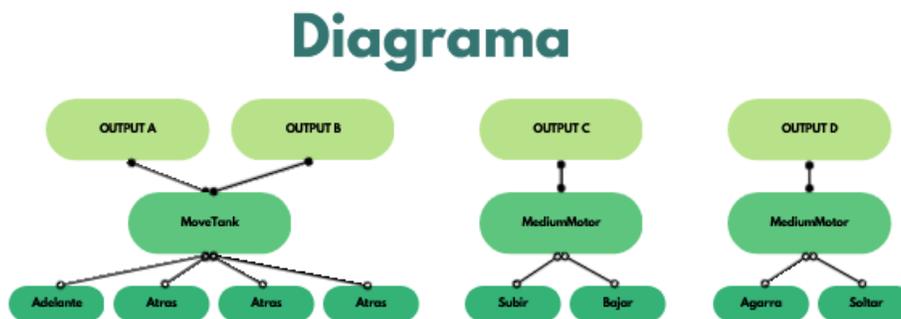
```
119 # Configurar la imagen de fondo
120 background_label = QLabel(main_window)
121 background_pixmap = QPixmap("/home/proyecto1/Descargas/WhatsApp Image 2024-10-30 at 19.47.20.jpeg") # Asegúra
122 background_label.setPixmap(background_pixmap)
123 background_label.setScaledContents(True) # Ajustar el contenido al tamaño del QLabel
124 background_label.setGeometry(0, 0, 1200, 675) # Establecer el tamaño del QLabel
125
126 # Crear botones circulares en posiciones únicas
127 button1 = QPushButton("", main_window)
128 button1.setGeometry(167, 334, 100, 100) # Botón 1
129 button1.setStyleSheet("border-radius: 50px; background-color: rgba(255, 0, 0, 0.5);") # Rojo
130 button1.pressed.connect(activar_wii_control)
131 button1.released.connect(detener)
132
133 button2 = QPushButton("", main_window)
134 button2.setGeometry(450, 400, 100, 100) # Botón 2
135 button2.setStyleSheet("border-radius: 50px; background-color: rgba(0, 255, 0, 0.5);") # Verde
136 button2.pressed.connect(izquierda)
137 button2.released.connect(detener)
138
139 button3 = QPushButton("", main_window)
140 button3.setGeometry(650, 400, 100, 100) # Botón 3
141 button3.setStyleSheet("border-radius: 50px; background-color: rgba(0, 0, 255, 0.5);") # Azul
142 button3.pressed.connect(derecha)
143 button3.released.connect(detener)
144
145 button4 = QPushButton("", main_window)
146 button4.setGeometry(550, 300, 100, 100) # Botón 4
147 button4.setStyleSheet("border-radius: 50px; background-color: rgba(255, 255, 0, 0.5);") # Amarillo
148 button4.pressed.connect(sdelanta)
149
150
151
152
153
154
155
156
157
158 button8.setGeometry(1027, 253, 100, 100) # Botón 8
159 button8.setStyleSheet("border-radius: 50px; background-color: rgba(255, 165, 0, 0.5);") # Naranja
160 button8.clicked.connect(Agarrar)
161
162 # Crear dos botones ovalados
163 oval_button1 = QPushButton("Ovalado 1", main_window)
164 oval_button1.setGeometry(925, 379, 155, 100) # Tamaño y posición del botón ovalado 1
165 oval_button1.setStyleSheet("border-radius: 50px; background-color: rgba(0, 128, 255, 0.5);") # Azul claro con
166 oval_button1.clicked.connect(Subir)
167
168 oval_button2 = QPushButton("Ovalado 2", main_window)
169 oval_button2.setGeometry(925, 495, 155, 100) # Tamaño y posición del botón ovalado 2
170 oval_button2.setStyleSheet("border-radius: 50px; background-color: rgba(255, 0, 128, 0.5);") # Rosa con trans
171 oval_button2.clicked.connect(Bajar)
172
173 # Asegúrate de que la etiqueta de fondo esté debajo de los botones
174 background_label.lower()
175
176 # Mostrar la ventana principal
177 main_window.show()
178
179 # Código principal
180 app = QApplication(sys.argv)
181
182 # Mostrar la ventana de configuración de IP primero
183 show_ip_window()
184
185 sys.exit(app.exec_())
186
187
188
```

### 6.3. Diagramas

Los principales movimientos y funciones del robot son a través de los botones, donde el output A y output B son los motores que desplazan al robot hacia Adelante, Atrás, Izquierda, derecha gracias al move tank el cual hace que se muevan simultáneamente

El output C a través del médium motor genera los movimientos del brazo del robot para subir y bajar la agarra

El output D a través del médium motor genera los movimientos del brazo del robot para agarra y soltar objetos



## 7. Resultados

### 7.1. Estado Actual del Proyecto

El proyecto lleva echo o en proceso las siguientes funciones:

- Arquitectura del robot.
- Conexión con el robot a través de wifi
- Conexión por servidor con el robot
- Creación de diseños temáticos con el robot (logo, interfaz, diseño).
- Creación a través de Python una interfaz gráfica para manejar el robot.
- Conteo y registro del procedimiento del proyecto a través de bitácora.
- Actualización de la Carta Gantt
- Implementación de un control de mando de Wii para mandar funciones de movimiento al robot.

7.2. *Problemas Encontrados y Solución Propuesta*

PROBELMAS	SOLUCIONES
El robot cada vez que avanzaban los días se iba descargando más rápido aun así lo cargáramos completamente.	Comprar pilas 6 pilas para que el robot pueda estar por un largo tiempo sin cargar o cargarlo completamente para solo utilizarlo en momentos claves.
Cuando la agarra acedia el rango de movimiento del robot causando su rompimiento.	Establecer en el código del robot una restricción de este movimiento a una sola vez después de presionarse
La interfaz tkinter no era adecuado para nuestro diseño de interfaz (botones redondos).	Se eligió la interfaz gráfica de Python PyQt5, ya que esta era la más adecuada para nuestra interfaz diseñada
La agarra del robot no podía agarrar objetos porque su arquitectura no se lo permitía	Se modifíco esta parte para que pueda la agarra pueda abrirse a un ángulo más amplio
Se quería implementar un control para el robot a través de un mando de Wii, pero las importaciones generales de esta como pygames o pywiid no asían una conexión exitosa	A través de evdev (lector de pulsaciones del mando) se puedo hacer que a través del mando de Wii el robot ejecute funciones de movimiento.

**8. Conclusión**

En el proyecto de desarrollo de robot se planificó y organizó cuidadosamente, donde los objetivos y/o tareas fueron distribuidas por todos los miembros del grupo. Con esto había una mejor gestión del tiempo y se podía llegar a los objetivos más rápidos, usando como recurso principal el uso de sets LEGO MINDSTORMS EV3.

A lo largo del proceso se abordaron aspectos importantes como el diseño piloto del robot, la asignación de recursos necesarios y la implementación de una metodología colaborativa y eficiente entre los miembros del equipo. Sin embargo, surgen algunos desafíos, especialmente para proyectos académicos, como la necesidad de ajustar los presupuestos de personal para que se ajusten a un marco más realista.

El equipo ha demostrado un fuerte compromiso con el proyecto y ha creado una base adecuada para futuras etapas de desarrollo. Sin embargo, es importante mejorar áreas como la justificación detallada de los costos y la documentación de las lecciones aprendidas para garantizar el éxito continuo del proyecto. En general, este informe representa un progreso significativo y sienta las bases para construir robots funcionales que cumplan los objetivos originales.

## 9. Referencias

EDUP N8508GS Wifi Adapter. (s/f). Indiamart.com. Recuperado el 8 de septiembre de 2024, de <https://www.indiamart.com/proddetail/edup-n8508gs-wifi-adapter-12990733673.html>

LEGO Education Set Base EV3. (s/f). ARQUIMED EDUCACION ESCOLAR. Recuperado el 8 de septiembre de 2024, de <https://www.arquimed.cl/educacion/product/set-base-ev3/>

(S/f). Intel.la. Recuperado el 8 de septiembre de 2024, de <https://www.intel.la/content/www/xl/es/products/sku/195436/intel-core-i510210u-processor-6m-cache-up-to-4-20-ghz/specifications.html>

(S/f). Winpy.cl. Recuperado el 8 de septiembre de 2024, de [https://www.winpy.cl/venta/notebook-toshiba-tecra-z40-c1410la/?srsltid=AfmBOoplLLKxv\\_ew6dvWVxOmNydWreuANGMDfxxAPDkCCSQQER5SODWU](https://www.winpy.cl/venta/notebook-toshiba-tecra-z40-c1410la/?srsltid=AfmBOoplLLKxv_ew6dvWVxOmNydWreuANGMDfxxAPDkCCSQQER5SODWU)

(S/f). Canva.com. Recuperado el 8 de septiembre de 2024, de [https://www.canva.com/es\\_419/](https://www.canva.com/es_419/)

(s/f). Microsoft.com. Recuperado el 8 de septiembre de 2024, de <https://www.microsoft.com/es-cl/microsoft-365/microsoft-office>

Entel: Telefonía Móvil, Internet Hogar Fibra y Celulares. (s/f). Personas. Recuperado el 8 de septiembre de 2024, de <https://www.entel.cl/>

(S/f). Lego.com. Recuperado el 17 de noviembre de 2024, de [https://www.lego.com/cdn/cs/set/assets/bltbef4d6ce0f40363c/LMSUser\\_Guide\\_LEGO\\_MINDSTORMS\\_EV3\\_11\\_Tablet\\_ENUS.pdf](https://www.lego.com/cdn/cs/set/assets/bltbef4d6ce0f40363c/LMSUser_Guide_LEGO_MINDSTORMS_EV3_11_Tablet_ENUS.pdf)

de Desarrollo Tecnológico UTA, U. (s/f). *Intranet - Universidad de Tarapacá*. Uta.cl. Recuperado el 17 de noviembre de 2024, de <https://portal.uta.cl/sign-in?redirectURL=%2Falumno%2Fclases%2F2024%2F2%2FIN056%2F1%2F58%2F108>

*Overview - grupo 1 A - redmine*. (s/f). Uta.Cl. Recuperado el 17 de noviembre de 2024, de <http://pomerape.uta.cl/redmine/projects/grupo-1-a?jump=welcome>