

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA

Departamento de Ingeniería en Computación e Informática



Avance del Proyecto “Scorpion Machine Gun”

Autor(es): -Alexis Yucra

-Francisco Silva

-Leandro Molina

-Sebastián Cayupi

-Arturo Rodríguez

Asignatura: Proyecto 1

Profesor(es): Leonel Alarcón Bravo

ARICA, 27 Octubre 2022

Historial de Cambios

Fecha	Versión	Descripción	Autor(es)
18/08/2022	1.0	Versión preliminar del formato	Alexis Yucra Francisco Silva Sebastián Cayupi Leandro Molina Arturo Rodríguez
25/08/2022	1.1	Revisión y Finalización del plan	Alexis Yucra Francisco Silva Sebastián Cayupi Leandro Molina Arturo Rodríguez
08/09/2022	1.2	Arreglos para la presentación	Alexis Yucra
15/09/2022	1.3	Arreglo para la formulación de proyectos	Alexis Yucra
02/11/2022	1.4	Actualización para la Presentación 2	Todos*

Tabla de Contenidos

1.	Panorama General	4
1.1	Introducción:.....	4
1.2	Objetivos:	4
1.2.1	Objetivo general.....	4
1.2.2	Objetivos Específicos:	4
1.3	Restricciones:	4
1.4	Entregables:	5
2.	Organización del Personal	5
2.1	Descripción de roles:.....	5
2.2	Personal que cumplirá los roles:	6
2.3	Mecanismos de Comunicación:	6
3.	Planificación del Proyecto.....	7
3.1	Actividades (nombre, descripción, responsable, producto):	7
3.2	Asignación de tiempo (carta Gantt Redmine):.....	8
3.3	Gestión de Riesgos (ver plantilla para el Tratamiento de los Riesgos):	8
4.	Planificación de los recursos.....	10
4.1	Recursos Hardware-Software requeridos.....	10
4.2	Estimación de costos (hardware, software, recursos humanos)	11
5.	Análisis y Diseño	12
5.1	Especificación de requerimientos	12
5.2	Arquitectura.....	13
5.3	Interfaz.....	13
6.	Implementación.....	14
6.1	Fundamentos de Proyectiles.....	14
6.2	Descripción de los programas.....	15
6.3	Diagramas	19
7.	Resultados.....	20
7.1	Estado actual del Proyecto.....	20
7.2	Problemas Encontrados y solución Propuesta.....	21
8.	Conclusión.....	21
9.	Referencia.....	22

1. Panorama General

1.1 Introducción:

En el presente informe se buscará poner en antecedentes acerca de cómo se trabajará a lo largo del proyecto, este consiste en el diseño, construcción y programación de un robot hecho con piezas del kit lego Mindstorm EV3, el cual como objetivo principal deberá ser capaz de disparar ligas y lograr tener un movimiento tipo tanque, es decir, dos ruedas largas con movimiento avanzar, retroceder en conjunto y por separado para permitir girar. El kit de lego viene con un minicomputador denominado Brick que servirá como cerebro del robot, a través de este se ejecutarán los programas encargados de guardar las funciones que lo controlan, y ser controlado vía computador inalámbricamente para decidir cuándo ejecutar dichas funciones a través de teclas.

1.2 Objetivos:

1.2.1 Objetivo general:

Construir un robot con el kit lego Mindstorm EV3 que sea capaz de realizar distintos movimientos y capaz de disparar ligas, además de desarrollar una interfaz capaz de controlar los movimientos y acciones del robot.

1.2.2 Objetivos Específicos:

Los objetivos específicos que han surgido en la creación del proyecto son los siguientes:

- Diseñar y construir un robot capaz de moverse y disparar ligas.
- Programar código Python capaz de controlar las acciones del robot.
- Diseñar interfaz para controlar los movimientos y acciones del robot.
- Entrega del producto final.

1.3 Restricciones:

A medida que íbamos avanzando en la construcción del robot y su configuración, Tuvimos las siguientes complicaciones:

- Falta de piezas del robot.
- Falta de experiencia a la hora de crear su algoritmo e interfaz.
- Programar en Python.

Las cuales fueron rápidamente resueltas ya que como dejamos a un encargado de ver las cosas faltantes del robot lo solicitamos y nos respondían rápidamente dándonos solución, Con la falta de experiencia a la hora de crear algoritmo la solución que le dimos fue juntarnos investigar más de cómo será programado, siguiendo ejemplos ya armados que salían en la página que se mostraba en el Manual del kit del robot.

1.4 Entregables:

Durante la realización del proyecto habrá una serie de documentos que entregar con el motivo de actualizar la información con respecto a los avances y organización del proyecto.

Informe de Formulación del Proyecto

- Presentación de Formulación del Proyecto
- Informe de Avance del Proyecto
- Presentación de Avance del Proyecto
- Informe Final del Proyecto
- Manual de usuario
- Presentación Final del Proyecto
- Producto Final

2. Organización del Personal

2.1 Descripción de roles:

Durante el desarrollo del proyecto se necesitó que los integrantes del grupo cumplan con cierto rol, con el propósito de tener una mejor administración en los tiempos de trabajo y comunicación obteniendo como resultado el cumplimiento de las tareas propuestas semanalmente respetando su tiempo de realización, Los roles que se seleccionaron son los siguientes:

- Programador: Es el que se encargara en la elaboración del algoritmo e interfaz para poder maniobrar al robot
- Líder de grupo: está al pendiente de todos los roles y mira el progreso del proyecto
- Ensamblador: Encargado de armar el robot y ver su funcionamiento.
- Documentador: encargo de tener la bitácora y el informe al día realizar fotos videos de los avances del proyecto.

- Diseñador: encargado de seleccionar el modelo que se basara el robot del proyecto.

2.2 Personal que cumplirá los roles:

Roles	Encargado	Descripción
Jefe de proyecto	Arturo Rodríguez	Representante del equipo de trabajo, toda decisión debe ser aprobada por el líder.
Ensamblador	Francisco Silva	Encargado de armar el modelo seleccionado por el grupo
Programador	Leandro Molina y Sebastián Cayupi	Desarrollar e implementar el algoritmo en Python, con el propósito de ejecutar todas las acciones solicitadas
Documentador	Alexis Yucra	Encargado de realizar los informes, presentaciones, bitácoras, videos manuales de usuario y wiki del proyecto
Diseñador	Alexis Yucra, Francisco Silva, Leandro Molina, Sebastián Cayupi y Arturo Rodríguez	Encargado de seleccionar el diseño del robot

2.3 Mecanismos de Comunicación:

Los medios de comunicación que se utilizarán para la elaboración del proyecto serán los siguientes:

- Discord (Software Libre): este software se utilizará para realizar reuniones externas de las clases presenciales para la organización del proyecto.
- WhatsApp (Software Libre): esta aplicación es utilizada para avisar la coordinación de reuniones y para avisar si es que uno falta.
- Zoom (Software Libre): este software se utilizará para en caso de emergencia por si se llega a caer discord.

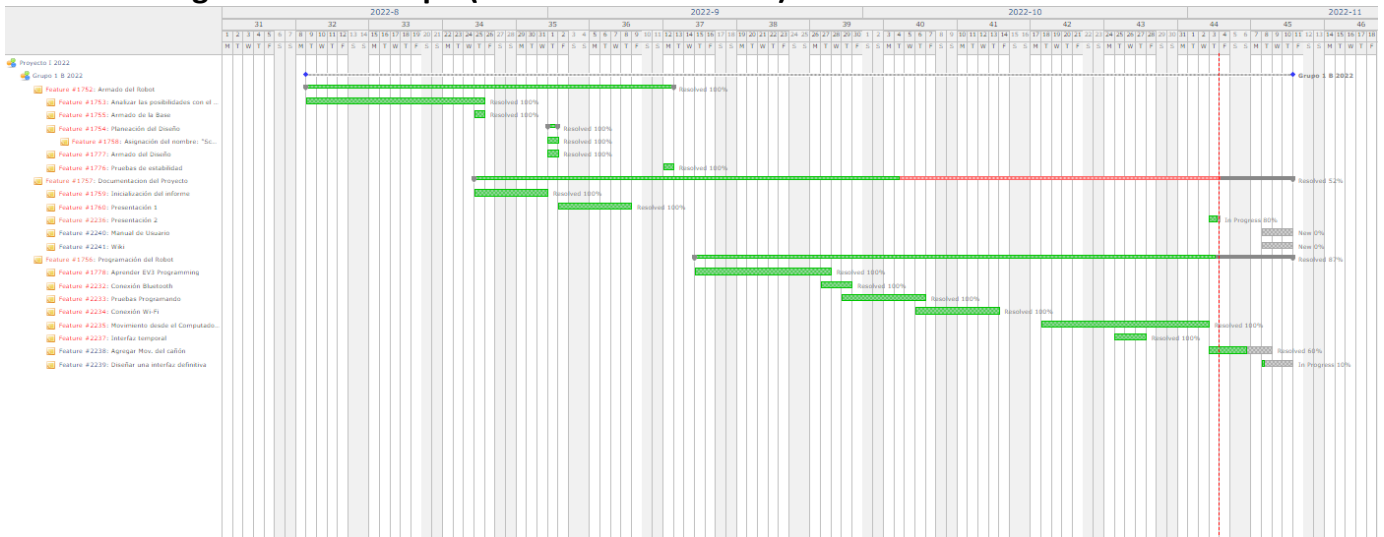
el objetivo de utilizar estos mecanismos de comunicación es más que todo para la organización que tendremos externas a las clases y justificar las ausencias de las reuniones, Donde también se comentan las ideas nuevas donde se verá si se aprueba o no por todos los integrantes.

3. Planificación del Proyecto

3.1 Actividades:

- Proyecto I 2022
 - Grupo 1 B 2022
 - Feature #1752: Armado del Robot
 - Feature #1753: Analizar las posibilidades con el ...
 - Feature #1755: Armado de la Base
 - Feature #1754: Planeación del Diseño
 - Feature #1758: Asignación del nombre: "Sc...
 - Feature #1777: Armado del Diseño
 - Feature #1776: Pruebas de estabilidad
 - Feature #1757: Documentacion del Proyecto
 - Feature #1759: Inicialización del informe
 - Feature #1760: Presentación 1
 - Feature #2236: Presentación 2
 - Feature #2240: Manual de Usuario
 - Feature #2241: Wiki
 - Feature #1756: Programación del Robot
 - Feature #1778: Aprender EV3 Programming
 - Feature #2232: Conexión Bluetooth
 - Feature #2233: Pruebas Programando
 - Feature #2234: Conexión Wi-Fi
 - Feature #2235: Movimiento desde el Computado...
 - Feature #2237: Interfaz temporal
 - Feature #2238: Agregar Mov. del cañón
 - Feature #2239: Diseñar una interfaz definitiva

3.2 Asignación de tiempo (carta Gantt Redmine):



Link: [Carta Gantt \(imagen\)](#) o ver en [Redmine](#)

3.3 Gestión de Riesgos (ver plantilla para el Tratamiento de los Riesgos):

RIESGOS	PROBABILIDAD DE OCURRENCIA	NIVEL DE IMPACTO	ACCIÓN REMEDIAL
Se necesitará cambiar las especificaciones del proyecto	40%	2	Se buscará información para crear un proyecto de menor dificultad
Uno o más miembros del grupo se enferman	20%	3	Se reorganizarán las tareas de los miembros afectados y así disminuir los riesgos
No se termina en el tiempo esperado	30%	2	Se buscará una solución que permita terminar el

			proyecto los más antes posible
Uno o más miembros dejan el proyecto	10%	1	Se reorganizarán las tareas
Falta de piezas/herramientas para el proyecto	20%	2	Se buscará información de como terminar el proyecto buscando otro tipo de herramientas
Catástrofes Naturales	40%	1	Se intentará resumir el proyecto lo más antes posible, sin poner en riesgo la vida y salud de los integrantes
Pérdida total de archivos o procesos	10%	1	Recrear todo lo perdido lo más fiel posible, basándonos en nuestra memoria
Falta de información para realizar el proyecto	50%	3	Informarse de forma completa antes de seguir con el proyecto

Factores de Riesgos:

TIPO DE RIESGO	INDICADORES POTENCIALES
Tecnología	No se cuenta con el software o herramientas necesarias
Salud	Algún integrante del grupo se contagia con COVID-19
Organización	Ocurre algún contratiempo que complica la realización del proyecto
Estimación	Las expectativas no son realistas y se requerirá cambios en el proyecto

4. Planificación de los recursos**4.1 Recursos Hardware-Software requeridos**

los recursos que se utilizaron para la elaboración de este proyecto son los siguientes:

- Hardware: Lego Mindstorms EV3, Tarjeta SD 8 GB.
- Software: Lenguaje Python, Visual Studio Code.

4.2 Estimación de costos (hardware, software, recursos humanos)

en este apartado se le mostrará detalladamente los recursos utilizados para llevar a cabo el proyecto:

Recursos utilizados:

a) Materiales:

Recursos	Valor de recursos	Cantidad	Precio total
Notebook MSI Modern	\$463.907	1	\$463.907
micro SD	\$5.900	1	\$5.900
Kit Lego Ev3	\$1.233.999	1	\$1.233.999
Notebook Asus Tuf Gaming	\$649.990	1	\$649.990
Adaptador USB inalámbrico TP-LINK TL-WN725N NANO	\$9.900	1	\$9.900

Monto total de los materiales: \$2.363.696

b) Estimación de costos (Trabajadores):

◆ Trabajo manual:

Función	Valor del trabajo	Cantidad de trabajadores	Horas trabajadas	Meses trabajados	Valor total
Programador	\$7.500	2	19	5	\$1.425.000
ensamblador	\$7.000	1	4	5	\$140.000
diseñador	\$7.000	5	4	5	\$700.000

Monto total de trabajo manual: \$2.265.000

◆ Trabajo por documento:

Función	Valor del trabajo	Cantidad de trabajadores	Cantidad de documentos	Valor total
Documentador	7.000	1	11	77.000

Monto total de trabajo por documento: \$56.000

Valor total del proyecto:

trabajo por documento valor total	trabajo manual valor total	Materiales valor total	Monto total del proyecto
\$77.000	\$2.265.000	\$2.363.696	4.705.696

5. Análisis y Diseño

5.1 Especificación de requerimientos

Especificación de requerimientos funcionales y no funcionales:

Funcionales:

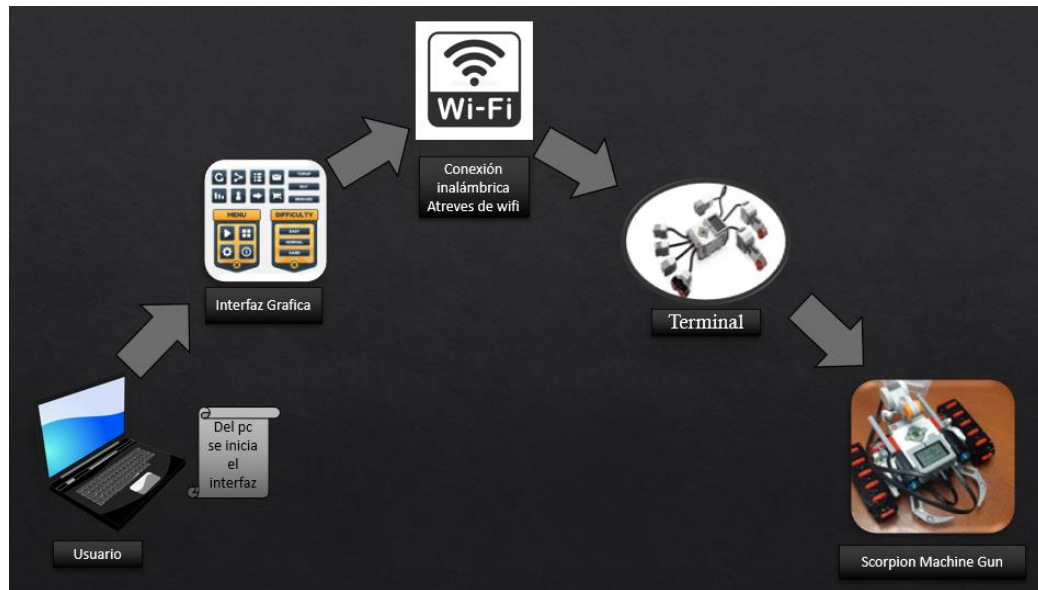
- Construcción del robot conectado vía wifi.
- Que el usuario pueda decidir los movimientos del robot.
- Implementación de una interfaz gráfica para la realización de los movimientos del robot donde se podrá controlar el robot, como por ejemplo el disparo del elástico del cañón que se pueda manipular de arriba para abajo o viceversa.

No Funcionales:

- La elaboración del algoritmo será a través de lenguaje de programación Python.

- Elaboración de un manual de usuario con el fin de explicar todas las funciones detalladamente.

5.2 Arquitectura



5.3 Interfaz



1. Botón encargado que el robot avance hacia delante.
2. Botón encargado que el robot gire hacia la izquierda.
3. Botón encargado que el robot retroceda.
4. Botón encargado que el gire a la derecha.
5. Botón encargado de conectar el robot al servidor y además puede desconectar al robot del servidor
6. Botón que realiza la acción de subir el cañón.
7. Botón que hace que el robot dispare las ligas.
8. Botón que realiza la acción de subir cañón.
9. Este botón despliega una pequeña guía de la interfaz.

6. Implementación

6.1 Fundamentos de proyectiles

El concepto físico que se presenta en nuestro proyecto a la hora de lanzar el elástico, se puede apreciar el lanzamiento de proyectil.

¿Qué es?

Un proyectil es cualquier objeto que se proyectará una vez que continúa en el movimiento por su propia inercia y es influenciado solamente por la fuerza hacia abajo de la gravedad.

Por definición, un proyectil tiene solamente una fuerza que actúa sobre él, esta es la fuerza de gravedad. Si hubiera alguna otra fuerza que actuara sobre un objeto, ese objeto no sería un proyectil. La gravedad actúa para influenciar el movimiento vertical del proyectil. El movimiento horizontal del proyectil es el resultado de la tendencia de cualquier objeto a permanecer en movimiento a velocidad constante.

Se denomina proyectil a cualquier objeto al que se le da una velocidad inicial y a continuación sigue una trayectoria determinada por la fuerza gravitacional que actúa sobre él y por la resistencia de la atmósfera. El camino seguido por un proyectil se denomina trayectoria.

El término proyectil se aplica por ejemplo a una bala disparada por un arma de fuego, a un cohete después de consumir su combustible, a un objeto lanzado desde un avión o en muchas actividades deportivas.

Se obtienen las ecuaciones cinemáticas del movimiento de un proyectil:

$$Y_m = \frac{V_o^2 \text{sen}^2 \alpha}{2g}$$

$$X_m = \frac{V_o^2 \text{sen} 2\alpha}{g}$$

$$t_v = \frac{2V_o \text{sen} \alpha}{g}$$

$$V = \sqrt{V_o^2 + V_y^2}$$

6.2 Descripción de los programas

- Visual Studio Code: Es un editor de código fuente. Incluye soporte para depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.
- PuTTY: Es un emulador de terminal gratuito que admite varios protocolos de red tal como SSH.

Implementación del servidor:

```

1  import socket
2  from main 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 == 'q') :
25         stop()
26     if (char=='i'):
27         levantarT()
28     if (char=='m'):
29         bajarT()
30     if (char=='k'):
31         disparar()

```

Funciones del robot:

```

3  from ev3dev2.motor import LargeMotor, MediumMotor, OUTPUT_A, OUTPUT_B, OUTPUT_C, OUTPUT_D, MoveTank
4
5  mC = MediumMotor(OUTPUT_D)
6  mI = LargeMotor(OUTPUT_B)
7  mD = LargeMotor(OUTPUT_C)
8  mA = LargeMotor(OUTPUT_A)
9
10 tankmoves = MoveTank(OUTPUT_B, OUTPUT_C)
11
12 def moveUp():
13     print("Moving up...")
14     tankmoves.on_for_seconds(left_speed=-100, right_speed=-100, seconds=0.1, brake=True, block=False)
15
16 def moveDown():
17     print("Moving down...")
18     tankmoves.on_for_seconds(left_speed=100, right_speed=100, seconds=0.1, brake=True, block=False)
19
20 def moveRight():
21     print("Moving right...")
22     tankmoves.on_for_seconds(left_speed=-100, right_speed=1, seconds=0.1, brake=True, block=False)
23
24 def moveLeft():
25     print("Moving left...")
26     tankmoves.on_for_seconds(left_speed=1, right_speed=-100, seconds=0.1, brake=True, block=False)
27
28 def levantarI():
29     print("Cannon moving up...")
30     mA.on_for_rotations(speed = 10, rotations = -0.05, brake=True, block=False)
31
32 def bajarI():
33     print("Cannon moving down")
34     mA.on_for_rotations(speed = 10, rotations = 0.05, brake=True, block=False)
35
36 def disparar():
37     print("disparando")
38     mC.on_for_seconds([speed = -100, seconds = 0.1], brake=True, block=False)
39
40 def stop():
41     tankmoves.stop()

```

Interfaz grafica:

```

1  from tkinter import *
2  import socket
3  from tkinter import messagebox
4
5  estado = False
6
7  def conectar(address, port):
8      global estado
9      if (estado == True):
10         botonEncendido.config(image=fotoApagado)
11         clientSocket.close()
12         estado = False
13     else:
14         botonEncendido.config(image=fotoEncendido)
15         clientSocket.connect((address, port))
16         messagebox.showinfo("Mensaje Servido", "Cliente conectado al robot: {0} : {1}".format(ipAddress, port))
17         estado=True
18

```

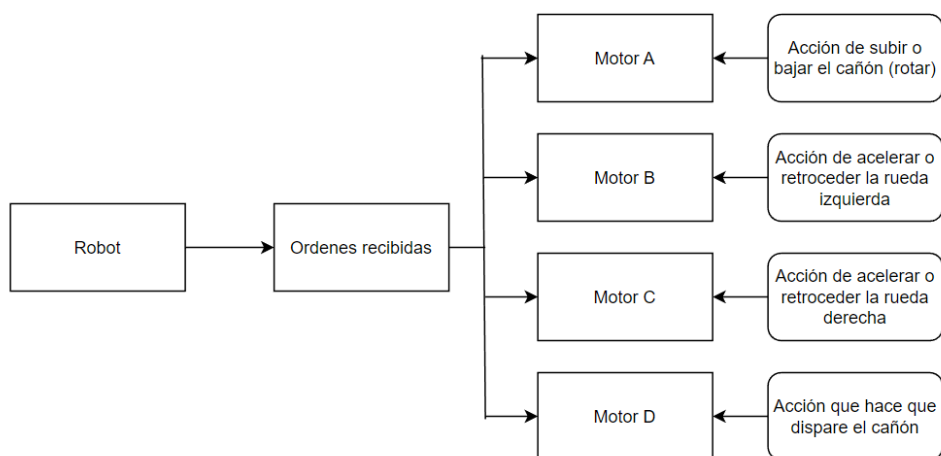


```
19 def avanzar():
20     clientSocket.send(bytes([ord('w')]))
21
22 def retroceder():
23     clientSocket.send(bytes([ord('s')]))
24
25 def girarIzquierda():
26     clientSocket.send(bytes([ord('a')]))
27
28 def girarDerecha():
29     clientSocket.send(bytes([ord('d')]))
30
31 def subirCañon():
32     clientSocket.send(bytes([ord('i')]))
33
34 def bajarCañon():
35     clientSocket.send(bytes([ord('m')]))
36
37 def disparar():
38     clientSocket.send(bytes([ord('k')]))
39
40 def on_release(event):
41     if event.char=='k':
42         print("dejar de disparar")
43     else:
44         clientSocket.send(bytes([ord('q')]))
45
46
47 def ventaInfo(event):
48     ventanaSecundaria = Toplevel(ventana)
49     ventanaSecundaria.geometry("605x300")
50
51     bg = PhotoImage(file = r"C:\Users\Leandro\Desktop\archivos\ev3\prueba\Pruebas5\Informacion.png")
52     label = Label(ventanaSecundaria, image=bg,)
53     label.place(x=0,y=0)
54     ventanaSecundaria.mainloop()
55
56
57
58 clientSocket = socket.socket()
59 ipAddress = "192.168.71.170"
60 port = 8080
```

```
62 ventana = Tk()
63 ventana.geometry('450x250')
64 ventana.title('Scorpion Machine Gun')
65 ventana.resizable(0,0)
66 ventana.config(bg='gray90')
67 ventana.bind('<w>',lambda event:avanzar())
68 ventana.bind('<a>',lambda event:girarIzquierda())
69 ventana.bind('<d>',lambda event:girarDerecha())
70 ventana.bind('<s>',lambda event:retroceder())
71 ventana.bind('<i>', lambda event:subirCañon())
72 ventana.bind('<m>', lambda event:bajarCañon())
73 ventana.bind('<k>', lambda event: disparar())
74 ventana.bind('<KeyRelease>', on_release)
75
76 fotoW = PhotoImage(file = r"C:\Users\Leandro\Desktop\archivos\ev3\prueba\Pruebas5\Pad-AR.png")
77 fotoA = PhotoImage(file = r"C:\Users\Leandro\Desktop\archivos\ev3\prueba\Pruebas5\Pad-I.png")
78 fotoD = PhotoImage(file = r"C:\Users\Leandro\Desktop\archivos\ev3\prueba\Pruebas5\Pad-D.png")
79 fotoS = PhotoImage(file = r"C:\Users\Leandro\Desktop\archivos\ev3\prueba\Pruebas5\Pad-AB.png")
80 fotoI = PhotoImage(file = r"C:\Users\Leandro\Desktop\archivos\ev3\prueba\Pruebas5\Subir.png")
81 fotoK = PhotoImage(file = r"C:\Users\Leandro\Desktop\archivos\ev3\prueba\Pruebas5\Disparar.png")
82 fotoM = PhotoImage(file = r"C:\Users\Leandro\Desktop\archivos\ev3\prueba\Pruebas5\Bajar.png")
83 fotoInfo = PhotoImage(file = r"C:\Users\Leandro\Desktop\archivos\ev3\prueba\Pruebas5\boton-de-informacion.png")
84 fotoEncendido = PhotoImage(file = r"C:\Users\Leandro\Desktop\archivos\ev3\prueba\Pruebas5\Encendido.png")
85 fotoApagado = PhotoImage(file = r"C:\Users\Leandro\Desktop\archivos\ev3\prueba\Pruebas5\Apagado.png")
86 fotologo = PhotoImage(file = r"C:\Users\Leandro\Desktop\archivos\ev3\prueba\Pruebas5\Scorpion M.G.png")
87 icono = PhotoImage("Escudo.ico")
88
89 ventana.iconbitmap(True, icono)
90
91 botonAvanzar = Label(ventana, image = fotoW, bg='gray90')
92 botonAvanzar.place(x=70, y=35)
93
94 botonGirarIzq = Label(ventana,image=fotoA, bg='gray90')
95 botonGirarIzq.place(x=20, y=100)
96
97 botonRetroceder = Label(ventana,image=fotoS, bg='gray90')
98 botonRetroceder.place(x=70, y=150)
99
100 botonGirarDer = Label(ventana,image=fotoD, bg='gray90')
101 botonGirarDer.place(x=110, y= 100)
102
103 botonBajarCañon = Label(ventana,image=fotoM, bg='gray90')
104 botonBajarCañon.place(x=360, y = 170)
105
```

```
106 botonSubirCañon = Label(ventana,image=fotoI, bg='gray90')
107 botonSubirCañon.place(x=360, y = 30)
108
109 botonDisparar = Label(ventana,image=fotoK, bg='gray90')
110 botonDisparar.place(x=360, y = 100)
111
112 botonInformacion = Label(ventana, image=fotoInfo, bg = 'gray90')
113 botonInformacion.place(x=430, y=10)
114
115 botonEncendido = Label(ventana, image=fotoApagado, bg='gray90')
116 botonEncendido.place(x=230, y=120)
117
118 logo = Label(ventana, image=fotologo, bg = 'gray90')
119 logo.place(x=220, y = 40)
120
121 botonEncendido.bind('<Button-1>', lambda event:conectar(ipAddress,port))
122 botonInformacion.bind('<Button-1>', ventaInfo)
123 botonAvanzar.bind('<Button-1>', lambda event:avanzar())
124 botonRetroceder.bind('<Button-1>', lambda event:retroceder())
125 botonGirarDer.bind('<Button-1>', lambda event: girarDerecha())
126 botonGirarIzq.bind('<Button-1>',lambda event: girarIzquierda())
127 botonSubirCañon.bind('<Button-1>', lambda event:subirCañon())
128 botonBajarCañon.bind('<Button-1>', lambda event:bajarCañon())
129 botonDisparar.bind('<Button-1>', lambda event:disparar())
130
131 ventana.mainloop()
132
```

6.3 Diagramas



7. Pruebas

7.1 Descripción de las pruebas realizadas

Pruebas de movimiento: verificar que las funciones de movimiento del robot se realicen exitosamente, la cual consisten en avanzar, girar izquierda, derecha disparar con el cañón subir y bajar cañón

Pruebas de disparo: consiste en realizar el disparo de elásticos con el cañón.

estabilidad del robot: consiste en que a la hora de realizar los movimientos no se desarme o se dañe.

7.2 Resultados de las pruebas

Pruebas de movimiento: los movimientos del robot como avanzar subir bajar el cañón todas funcionaron con éxito.

Pruebas de disparo: la prueba de disparo del elástico con el cañón fue ejecutada correctamente.

estabilidad del robot: se verifico la estabilidad ala hora de ejecutar cualquier función solicitada del interfaz, no generando ningún daño al robot y con una buena estabilidad.

8. Resultados

8.1 Estado actual del Proyecto

Estado actual del robot:

- El diseño del robot está finalizado sin problemas a la hora de manipular.
- Funciones definidas.
- Interfaz de comando.
- Movimiento del robot a la hora de girar
- La wiki del proyecto lista
- Diseño en la interfaz finalizada.
- Movimiento del cañón de subida y bajada y disparo

8.2 Problemas Encontrados y solución Propuesta

Problemas encontrados:

- Conectar la interfaz creada mediante tkinter a través de RPYC al ev3 en Lenguaje de Python
- Problema con el USB wifi para poder conectarnos con el robot ev3

Soluciones de los problemas:

- Conexión a través del programa socket, PuTTY
- Conseguir un USB compatible.

9. Conclusión

9.1 Conclusión general:

En la formulación del proyecto, a la hora de diseñar el robot es fundamental una buena comunicación con el equipo de trabajo ya que esto también influirá en la organización que se tendrá en el proyecto como la planificación de tareas a realizar que se serán acordadas por el equipo de trabajo.

Para la ejecución de cualquier tipo de proyecto es importante tener planes de emergencia en caso de que pase algún problema, para así brindar soluciones rápidas a dichos imprevistos.

Es necesario conocer bien las herramientas de trabajo que se estén utilizando en el proyecto.

9.2 Trabajo a futuro

Mejorar la interfaz, haciéndola mas llamativa visualmente

Implementar la conexión con un mando de consola al robot

Buscar distintas aplicaciones o modificaciones para crear nuevos usos

10. Referencia

Enlaces de los que se recopiló información de los productos e investigaciones:

1. <https://www.solotodo.cl/products/146440-msi-modern-14-b5m-ryzen-5-5500u-8gb-256gb-ssd-w11h>

2. <https://www.falabella.com/falabella-cl/product/16110275/Laptop-M415DA-EK963W-AMD-Ryzen-7-8GB-RAM-512GB-SSD-14-FHD/16110275>

3. https://articulo.mercadolibre.cl/MLC-984997712-mindstorm-ev3-core-set-45544-new-_JM#position=3&search_layout=stack&type=item&tracking_id=5b85e87d-9ae8-4de5-8be7-979235d32260

4. https://articulo.mercadolibre.cl/MLC-635353920-sony-memoria-micro-sdhc-8gb-clas10-cadaptador-70mb-ecoffice-_JM#position=3&search_layout=stack&type=item&tracking_id=4ec324ba-4667-4bb4-b7f1-cf1fad792243