

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA



DEPARTAMENTO DE INGENIERÍA CIVIL EN COMPUTACIÓN E INFORMÁTICA



Plan de Proyecto

“Ev3 Pascalito”

Alumno(os):	<ul style="list-style-type: none">- Denis Condori- Esteban Gutierrez- Ignacio Gallardo- Fernando Klinger- Martin Salinas
Asignatura:	Proyecto I
Profesor:	Humberto Urrutia

Historial de Cambios

Fecha	Versión	Descripción	Autor(es)
03/09/2023	1.0	Formulación del Proyecto	-Denis Condori -Fernando Klinger -Martin Salinas -Esteban Gutierrez -Ignacio Gallardo
06/12/2023	2.0	Etapa 2	-Denis Condori -Fernando Klinger -Martin Salinas -Esteban Gutierrez -Ignacio Gallardo

Tabla de Contenidos

1. Panorama General.....	4
Introducción:.....	4
Objetivos:.....	4
• Objetivo General:.....	4
• Objetivo Específico:.....	4
Restricciones:.....	5
Entregables:.....	5
2. Organización del Personal.....	5
Descripción y asignación de roles actuales.....	5
3. Planificación del Proyecto.....	6
• 3.1 Actividades:.....	6
• Asignación de tiempo:.....	8
La Carta Gantt fue creada con el propósito de planificar y gestionar de manera más eficaz nuestras actividades a lo largo del semestre. Además, se puede registrar el tiempo dedicado para cada tarea y compararlo con el tiempo estimado.....	8
• Gestión de riesgos.....	9
4. Planificación de los Recursos.....	10
• Costo Hardware:.....	10
• Costos de Software:.....	10
• Costo Recursos Humanos:.....	11
5. Análisis y diseño.....	12
5.1 Requerimientos.....	12
Requerimientos funcionales:.....	12
Requerimientos no funcionales:.....	12
5.2. Arquitectura:.....	12
5.3 Interfaz:.....	13
6. Implementación:.....	14
6.1 Física detrás del golpe de golf:.....	14
- Leyes de newton:.....	14
- Fundamentos de Projectiles o movimiento parabólico:.....	14
- Fricción:.....	15
6.2 Descripción de los programas:.....	16
• Servidor:.....	16
• Cliente:.....	17
6.3 Diagramas.....	19
Movimiento:.....	19
Golpe:.....	19
7. Resultados.....	20
7.1 Estado actual del proyecto.....	20
7.2 Problemas encontrados y Solución propuesta.....	20
8. Conclusión:.....	21
9. Referencias:.....	21

1. Panorama General

Introducción:

El proyecto denominado "PASCALITO", desarrollado como parte de la asignatura Proyectos I, se construyó utilizando componentes del kit LEGO EV3.

Es importante destacar que, para llevar a cabo este proyecto, se aprovecharán tanto los conocimientos adquiridos en la formación de los estudiantes de la carrera como la aplicación de los conceptos enseñados en los cursos de Taller de Programación I y II.

Objetivos:

- Objetivo General:

Diseñar y construir un robot utilizando el kit de Lego Mindstorms EV3 con la capacidad de desplazarse y golpear una pelota, imitando las acciones de un jugador de golf. Este robot será controlado por un usuario.

- Objetivo Específico:

- Crear un servidor que permita la conexión del cliente con el robot.
- Aplicar la teoría física del tiro parabólico.

Restricciones:

Se presentan diversas restricciones que deben superarse para lograr un desempeño exitoso en el desarrollo del robot. Estas limitaciones se detallan a continuación.

- **Tiempo:** fecha límite establecida para hacer el proyecto.
- **Set LEGO EV3:** Utilización de kit LEGO MINDSTORM EV3.
- **Documentación:** Todos los archivos redactados, serán subidos a Redmine.
- **Sistema operativo:** Utilización de sistema operativo Linux.

Entregables:

- Bitácoras semanales del avance realizado.
- Carta Gantt del detalle del proceso completo del proyecto.
- Informes.
- Presentaciones.
- Manual de usuario.

2. Organización del Personal

Se le designó una responsabilidad a cada miembro donde estarán a cargo de cumplir con el trabajo asignado.

Descripción y asignación de roles actuales.

ROL	DESCRIPCIÓN	ASIGNACIONES
Programador	Encargado de la creación del código para un correcto funcionamiento del robot y que este pueda cumplir con los objetivos del proyecto	- Denis Condori
Ensamblador	Encargado de la construcción del robot con el fin de que funcione óptimamente y cumpla con los objetivos señalados.	-Fernando Klinger
Documentador	Encargado de ingresar los avances realizados por el equipo a la plataforma "Redmine"	- Ignacio Gallardo
Diseñador	Encargado de diseñar una interfaz amigable con el usuario y logos del proyecto	-Esteban Gutierrez
Jefe de Grupo	Encargado de dirigir el proceso de planificación y ejecución del proyecto	-Martín Salinas

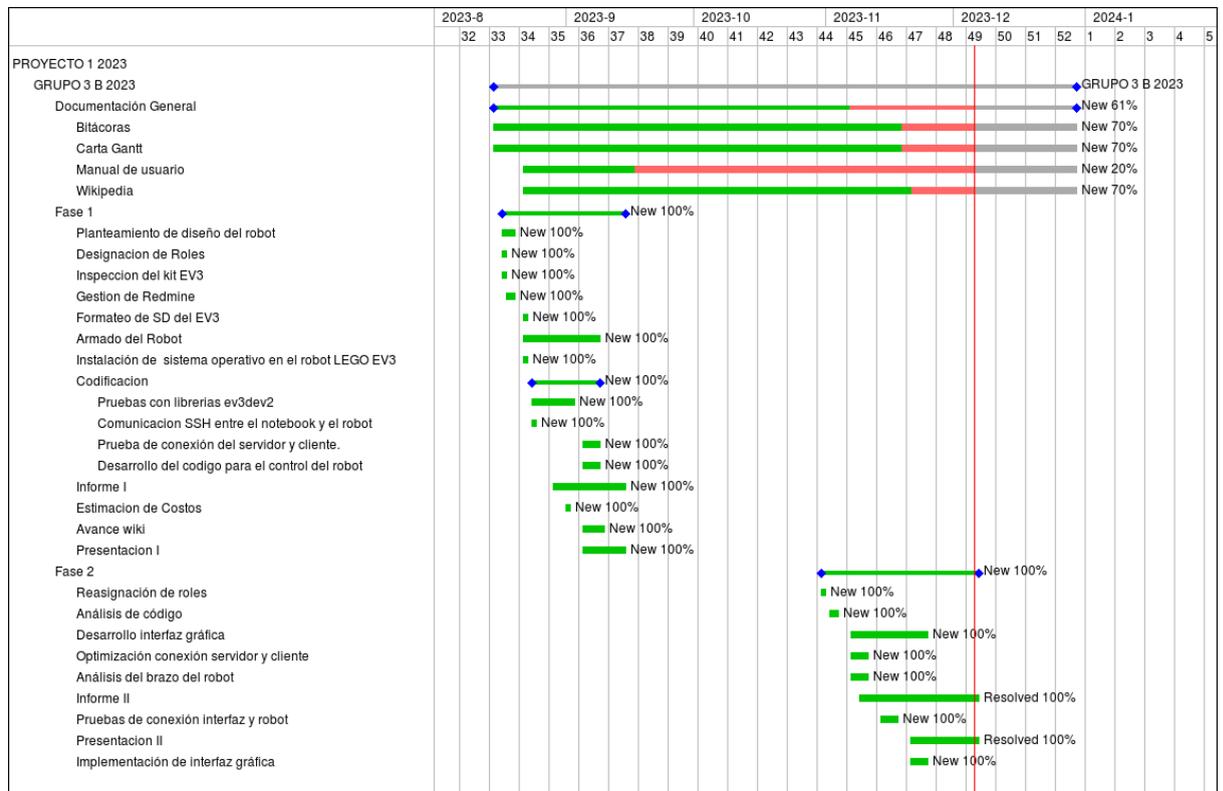
3. Planificación del Proyecto

- 3.1 Actividades:

Actividad	Descripción	Responsable	Involucrados
Bitácoras Semanales	Registro de todas las actividades que se desarrollan semanalmente.	- Ignacio Gallardo	- Martin Salinas - Denis Condori - Fernando Klinger - Esteban Gutierrez
Carta Gantt	Planificación de las actividades a lo largo del semestre.	- Esteban Gutierrez	- Martin Salinas - Ignacio Gallardo - Fernando Klinger - Denis Condori
Wiki	Se capturan y comparten ideas e información del proyecto.	- Ignacio Gallardo	- Esteban Gutiérrez
Organización	Designación de la actividad que estará encargado cada integrante.	- Martin Salinas	- Ignacio Gallardo
Contabilizar Piezas	Entrega del kit EV3 donde se contabilizaron las piezas.	- Esteban Gutierrez	- Martin Salinas - Denis Condori
Diseño del Robot	Búsqueda de ideas para el armado del robot	- Fernando Klinger	-Martin Salinas
Construcción del Robot	Armado de la base del robot.	- Martin Salinas	- Denis Condori - Martin Salinas - Ignacio Gallardo - Fernando Klinger
Instalación (SO) en la tarjeta SD	Encargado de instalar y comprobar el correcto funcionamiento del SO	- Ignacio Gallardo	-Ignacio Gallardo
Administrar Redmine	Subir y organizar documentos en	- Esteban Gutierrez	- Martin Salinas

	Redmine		
Informe I	Creación de informe I	- Ignacio Gallardo	- Fernando Klinger - Esteban Gutierrez - Denis Condori - Martin Salinas
Estimación de Costos	Calcular costo total del proyecto	- Denis Condori	- Denis condori - Martin Salinas
Presentación I	Creación de Presentación I	- Denis Condori	- Martin Salinas - Ignacio Gallardo - Fernando Klinger - Esteban Gutierrez
Programación	Programación de los movimientos y acciones	- Denis Condori	- Fernando Klinger - Esteban Gutierrez - Denis Condori - Martin Salinas
Implementación de la interfaz gráfica	Encargado de crear una interfaz funcional y amigable con el usuario	- Esteban Gutierrez	- Denis Condori - Martin Salinas - Ignacio Gallardo
Informe II	Redacción del informe del avance del proyecto.	- Martin Salinas	- Denis Condori - Ignacio Gallardo - Fernando Klinger - Esteban Gutierrez
Presentación II	Creación del material de apoyo sobre el avance del proyecto.	- Fernando Klinger	- Ignacio Gallardo - Esteban Gutierrez - Denis Condori - Martin Salinas

● Asignación de tiempo:



La Carta Gantt fue creada con el propósito de planificar y gestionar de manera más eficaz nuestras actividades a lo largo del semestre. Además, se puede registrar el tiempo dedicado para cada tarea y compararlo con el tiempo estimado.

- Gestión de riesgos

Niveles de impacto:

Bajo

Medio

Alto

Crítico

Riesgos	Probabilidad de concurrencia	Nivel de impacto	Acción remedial
Desarme por caída del robot	7%	Alto	Volver a construir el robot, de como estaba antes del desarme o caída.
Batería descargada	10%	Baja	Cargar la batería.
Rotura de pieza por caída del robot	10%	Alto	Pedir o comprar una nueva pieza, para reemplazar la rota
Pérdida de pieza	20%	Medio	Intentar encontrar la pieza perdida, o reemplazarla con una nueva
Incapacidad o inasistencia de un integrante	10%	Crítico	Justificar la inasistencia del integrante.
Daño o pérdida de tarjeta SD	5%	Alto	Comprar una nueva tarjeta SD
Escasez de piezas	2%	Medio	Comprar las piezas necesarias
Mala estimación del tiempo	5%	Alto	Reorganizarse como grupo con el tiempo perdido y restante
Reconstrucción del robot	40%	Medio	Implementar nuevas ideas

4. Planificación de los Recursos

- Costo Hardware:

Productos	Cantidad	Precio	Categoría
Arriendo Notebooks (4 meses)	5	\$1.200.000	Hardware
Kit Lego Mindstorm EV3	1	\$960.000	Hardware
Micro SD (8 GB)	1	\$8.000	Hardware
Dongle USB WIFI	1	\$10.142	Hardware
Router	1	\$26.000	Hardware
Control PS4	1	\$60.000	Hardware
Cable de carga PS4	1	\$4.000	Hardware
Gastos imprevistos	1	\$2.000.000	Empresa
Total		\$4.268.142	

- Costos de Software:

Producto	Cantidad	Precio
Python	5	GRATIS
Ev3dev	1	GRATIS
Whatsapp	5	GRATIS
Discord	5	GRATIS
Visual Studio Code	5	GRATIS
Linux	1	GRATIS
Licencia Office	5	\$60.000

- **Costo Recursos Humanos:**

Encargado	Personal	Horas trabajadas al mes	Valor hora por trabajador	Sueldo mensual	Sueldo total (4 meses)
Programador	1	40	\$16.000	\$ 640.000	\$ 2.560.000
Ensamblador	1	40	\$15.000	\$ 600.000	\$ 2.400.000
Jefe de grupo	1	40	\$19.000	\$ 760.000	\$ 3.040.000
Documentador	1	40	\$14.000	\$ 560.000	\$ 2.240.000
Diseñador	1	40	\$14.000	\$ 560.000	\$ 2.240.000
Costo Total					\$12.480.000

NOMBRE	COSTO TOTAL
Costos de Hardware	\$4.268.142
Costos de Software	\$60.000
Costos de Gestión	\$12.480.000
Costo total proyecto	\$16.808.142

5. Análisis y diseño

5.1 Requerimientos

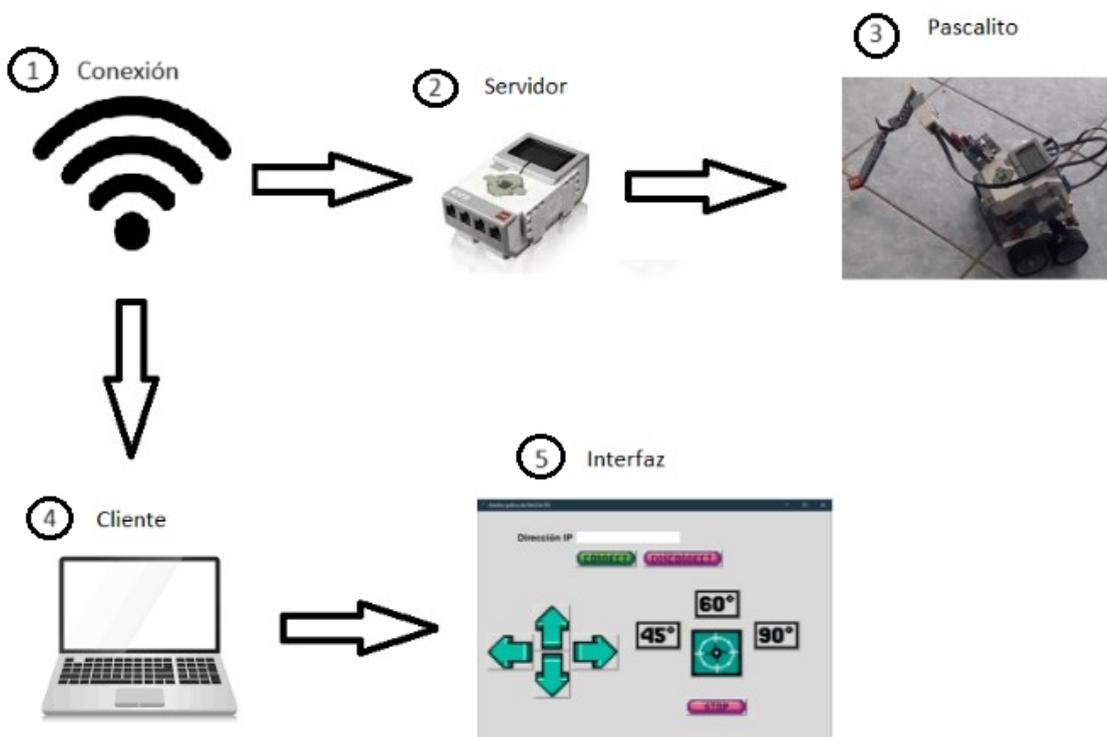
Requerimientos funcionales:

- El robot tiene que tener la capacidad de golpear una pelota con una estructura que imita a un palo.
- El robot debe tener la capacidad de moverse en todas las direcciones.
- El robot debe ser controlado por un usuario por medio del programa.
- Se requiere un servidor capaz de comunicar al programa con el usuario.

Requerimientos no funcionales:

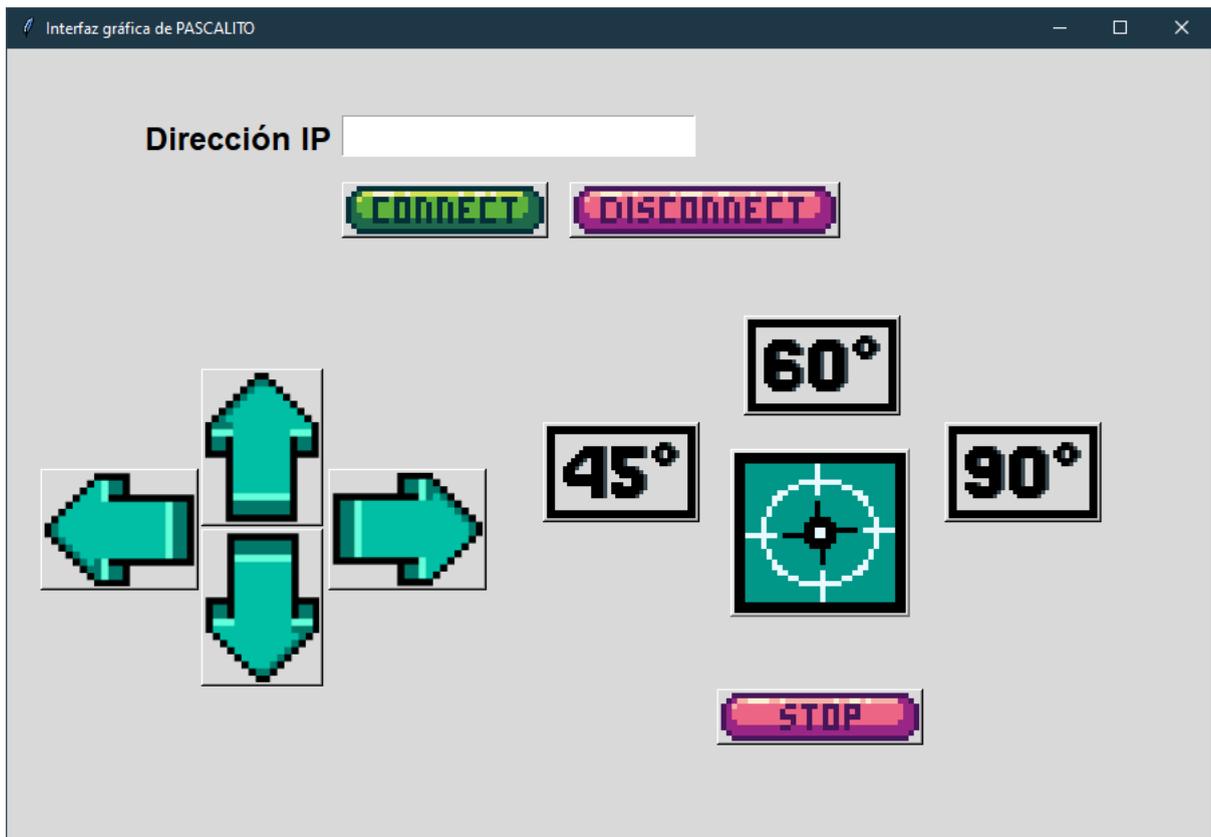
- La programación del software debe llevarse a cabo en el lenguaje Python y en el entorno del sistema operativo Linux.
- La interfaz debe ser eficiente y amigable con el usuario.

5.2. Arquitectura:



1. La conexión entre el servidor y el cliente será a través de la misma red wifi.
2. El servidor programado en Python se aloja en el ladrillo de comandos del ev3.
3. Robot Pascalito.
4. El cliente programado en Python estará en un notebook con el sistema operativo basado en Linux.
5. Interfaz gráfica del proyecto Pascalito programada usando la librería Tkinter.

5.3 Interfaz:



Direcciones de movimiento:

- Flechas para indicar las direcciones en las que el robot se desplazará, es decir, ir para adelante, para atrás, girar a la izquierda y girar a la derecha.

Ángulos del golpe:

- Botones de (45°, 60°, 90°) que tiene como función ingresar el ángulo del putter golf, para que la pelota obtenga un recorrido distinto.

Golpe a la pelota:

- Símbolo "ícono de mira" que logra pegarle a la pelota.

Conexión:

- Botón "Conectar" el cual vincula al cliente a través de la IP del servidor que está en el EV3.
- Botón de "Disconnect" que permite cerrar la conexión y el de "Stop" permite detener el robot.

6. Implementación:

6.1 Física detrás del golpe de golf:

- Leyes de newton:

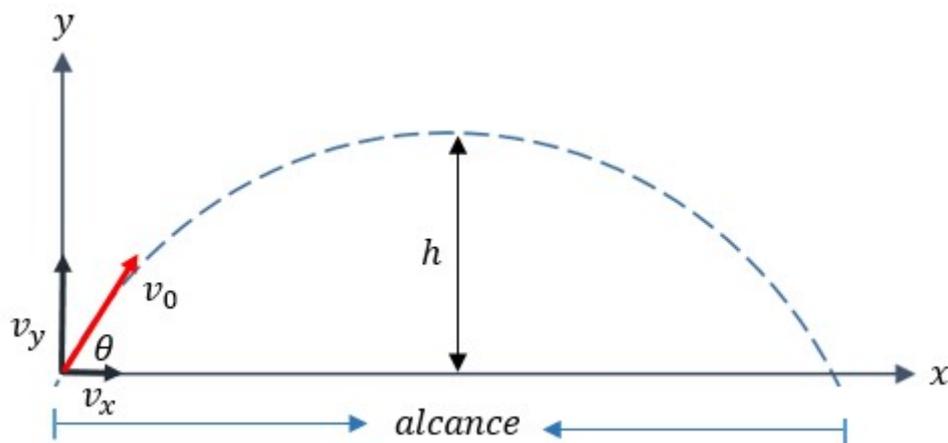
Primera ley de Newton (Inercia): Un objeto en reposo tiende a permanecer en reposo, y un objeto en movimiento tiende a permanecer en movimiento a una velocidad constante en línea recta, a menos que una fuerza neta actúe sobre él. Esto se aplica a la pelota de golf cuando está quieta en el tee y cuando se mueve a lo largo del recorrido.

Segunda ley de Newton (Fuerza y aceleración): La aceleración de un objeto es directamente proporcional a la fuerza neta que actúa sobre él, e inversamente proporcional a su masa. En el mini golf, la fuerza aplicada al golpear la pelota con el putter determina su aceleración y trayectoria.

Tercera ley de Newton (Acción y reacción): Por cada acción hay una reacción igual y opuesta. Cuando el putter golpea la pelota, la pelota aplica una fuerza igual y opuesta al putter.

- Fundamentos de proyectiles o movimiento parabólico:

La trayectoria de la pelota de golf sigue las leyes de la física de los proyectiles. El ángulo y la fuerza con los que golpeas la pelota afectan su trayectoria. Un ángulo y una fuerza precisos pueden permitir que la pelota alcance el hoyo.



Datos:

$$g = 9,8 \left(\frac{m}{s^2} \right); x_0 = 0 (m); y_0 = 0 (m); \theta = 45^\circ, 60^\circ, 90^\circ$$

Antes de plantear las ecuaciones de movimiento primero hay que descomponer la velocidad inicial (v_0) en los dos ejes:

$$\text{Velocidad inicial en } x (v_x): v_x = v_0 \cos(\theta)$$

Velocidad inicial en y (v_y): $v_y = v_o \sin(\theta)$

Ecuaciones de movimiento parabólico:

Movimiento en el eje X:

$$X = x_0 + v_x t$$

$$X = x_0 + v_o \cos(\theta)t$$

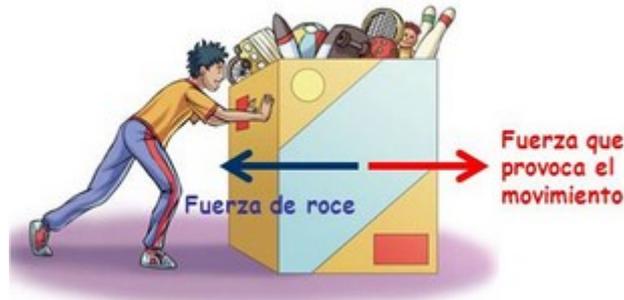
Movimiento en el eje Y:

$$Y = y_0 + v_y t - \frac{1}{2}gt^2$$

$$Y = y_0 + v_o \sin(\theta)t - \frac{1}{2}gt^2$$

- Fricción:

La fricción entre la pelota y la superficie del campo afecta su movimiento. Superficies más rugosas pueden disminuir la velocidad de la pelota, mientras que superficies más lisas permiten que la pelota ruede más fácilmente.



6.2 Descripción de los programas:

- Servidor:

```
1 import socket
2 from ev3dev2.motor import MediumMotor, OUTPUT_A, OUTPUT_B, OUTPUT_C, SpeedPercent, MoveTank
3
4 tankmoves = MoveTank(OUTPUT_B, OUTPUT_C)
5 brazo = MediumMotor(OUTPUT_A)
6
7 ip_host = input("Ingresa la ip que muestra el robot: ")
8 host = ip_host
9 port = 8000
10 server = socket.socket()
11 server.bind((host, port))
12 server.listen(5)
13 conexion, addr = server.accept()
14 print("Nueva conexion establecida")
15 print("conectado a ", addr)
16
17 running = True
18
19 while running:
20     mensaje = conexion.recv(1024).decode("utf-8")
21     print(mensaje)
22     if mensaje == 'arriba':
23         tankmoves.on(SpeedPercent(10), SpeedPercent(10))
24     elif mensaje == 'abajo':
25         tankmoves.on(SpeedPercent(-10), SpeedPercent(-10))
26     elif mensaje == 'izquierda':
27         tankmoves.on(SpeedPercent(20), SpeedPercent(-20))
28     elif mensaje == 'derecha':
29         tankmoves.on(SpeedPercent(-20), SpeedPercent(20))
30     elif mensaje == 'detener':
31         tankmoves.on(SpeedPercent(0), SpeedPercent(0))
32     elif mensaje == 'apagar':
33         running = False
34         conexion.close()
35         print("La conexion termino!")
36     elif mensaje == 'golpear':
37         brazo.on_for_degrees(SpeedPercent(30), 45)
38         brazo.on_for_degrees(SpeedPercent(100), -95)
```

- Cliente:

```
1 import tkinter as tk
2 import socket
3
4 def mover(direccion):
5     global mi_socket
6     if direccion == "arriba":
7         mi_socket.send("arriba".encode("utf-8"))
8     elif direccion == "abajo":
9         mi_socket.send("abajo".encode("utf-8"))
10    elif direccion == "izquierda":
11        mi_socket.send("izquierda".encode("utf-8"))
12    elif direccion == "derecha":
13        mi_socket.send("derecha".encode("utf-8"))
14
15 def conectar():
16    global mi_socket
17    # Función para manejar la conexión con la dirección IP ingresada
18    ip = entrada_ip.get()
19    host = str(ip)
20    port = 8000
21    mi_socket = socket.socket()
22    mi_socket.connect((host, port))
23
24 def apagar():
25    global mi_socket
26    mi_socket.send("apagar".encode("utf-8"))
27
28 def detener():
29    global mi_socket
30    mi_socket.send("detener".encode("utf-8"))
```

```
31
32 def angulo45():
33     global angulo
34     angulo = 45
35
36 def angulo60():
37     global angulo
38     angulo = 60
39
40 def angulo90():
41     global angulo
42     angulo = 90
43
44 def disparar():
45     global mi_socket
46     global angulo
47     if angulo == 45:
48         mi_socket.send("disparar45".encode("utf-8"))
49     elif angulo == 60:
50         mi_socket.send("disparar60".encode("utf-8"))
51     elif angulo == 90:
52         mi_socket.send("disparar90".encode("utf-8"))
53
54 # Crea la ventana principal
55 ventana = tk.Tk()
56 ventana.title("Interfaz gráfica de PASCALITO")
57 ventana.geometry("900x600")
58 ventana.config(bg="grey85")
```

```

58 ventana.config(bg="grey85")
59
60 # Etiqueta y entrada para la dirección IP
61 etiqueta_ip = tk.Label(ventana, text="Dirección IP", font=("Arial", 18, "bold"), bg="grey85")
62 etiqueta_ip.place(x=100, y=50)
63
64 entrada_ip = tk.Entry(ventana, font=("bold", 18))
65 entrada_ip.place(x=250, y=50)
66
67 # Creamos los botones para generar la conexión y para terminar la conexión
68 img_boton_conectar = tk.PhotoImage(file="connect.png")
69 boton_conectar = tk.Button(ventana, text="Conectar", image=img_boton_conectar, bg="grey85", command=conectar)
70 boton_conectar.place(x=250, y=100)
71
72 img_boton_desconectar = tk.PhotoImage(file="disconnect.png")
73 boton_desconectar = tk.Button(ventana, text="Desconectar", image=img_boton_desconectar, bg="grey85", command=apagar)
74 boton_desconectar.place(x=420, y=100)

```

```

76 # Botones de disparo
77 img_boton_45grados = tk.PhotoImage(file="45grados.png")
78 boton_45grados = tk.Button(ventana, text="45°", image=img_boton_45grados, bg="grey85", command=angulo45)
79 boton_45grados.place(x=400, y=280)
80
81 img_boton_60grados = tk.PhotoImage(file="60grados.png")
82 boton_60grados = tk.Button(ventana, text="60°", image=img_boton_60grados, bg="grey85", command=angulo60)
83 boton_60grados.place(x=550, y=280)
84
85 img_boton_90grados = tk.PhotoImage(file="90grados.png")
86 boton_90grados = tk.Button(ventana, text="90°", image=img_boton_90grados, bg="grey85", command=angulo90)
87 boton_90grados.place(x=700, y=280)
88
89 img_boton_disparar = tk.PhotoImage(file="shot.png")
90 boton_disparar = tk.Button(ventana, text="Disparar", image=img_boton_disparar, command=disparar)
91 boton_disparar.place(x=540, y=300)
92
93 # Boton para detener el movimiento
94 img_boton_stop = tk.PhotoImage(file="STOP.png")
95 boton_detener = tk.Button(ventana, text="Detener", image=img_boton_stop, bg="grey85", command=detener)
96 boton_detener.place(x=530, y=480)

```

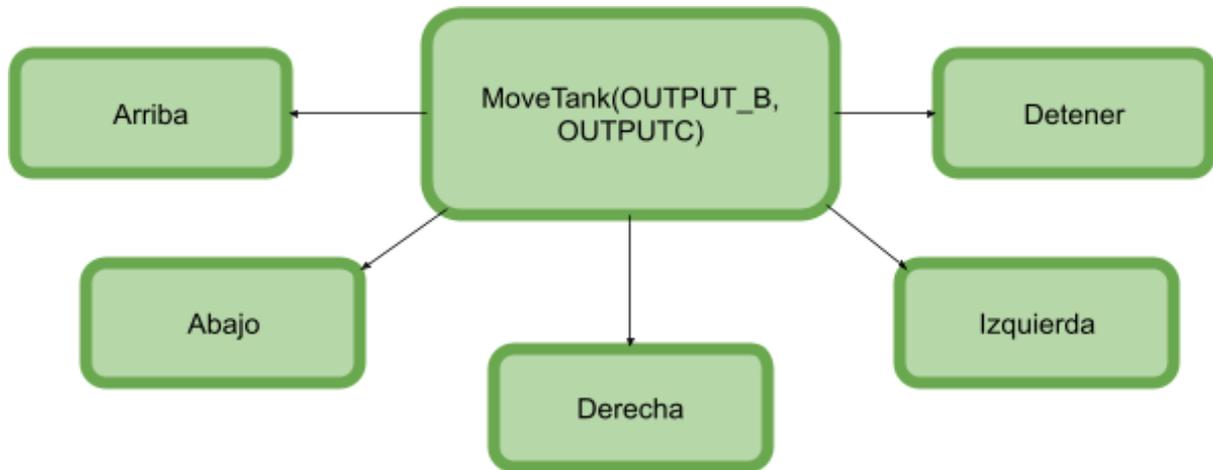
```

98 # Botones de movimiento
99 img_boton_arriba = tk.PhotoImage(file="flecha-up1.png")
100 boton_arriba = tk.Button(ventana, image=img_boton_arriba, bg="grey85", command=lambda: mover("arriba"))
101 boton_arriba.place(x=145, y=240)
102
103 img_boton_izquierda = tk.PhotoImage(file="flecha-izquierda.png")
104 boton_izquierda = tk.Button(ventana, text="Izquierda", image=img_boton_izquierda, bg="grey85", command=lambda: mover("izquierda"))
105 boton_izquierda.place(x=25, y=315)
106
107 img_boton_derecha = tk.PhotoImage(file="flecha-derecha.png")
108 boton_derecha = tk.Button(ventana, text="Derecha", image=img_boton_derecha, bg="grey85", command=lambda: mover("derecha"))
109 boton_derecha.place(x=240, y=315)
110
111 img_boton_abajo = tk.PhotoImage(file="flecha-abajo.png")
112 boton_abajo = tk.Button(ventana, text="Abajo", image=img_boton_abajo, bg="grey85", command=lambda: mover("abajo"))
113 boton_abajo.place(x=145, y=360)
114
115
116 # Iniciar el bucle principal
117 ventana.mainloop()

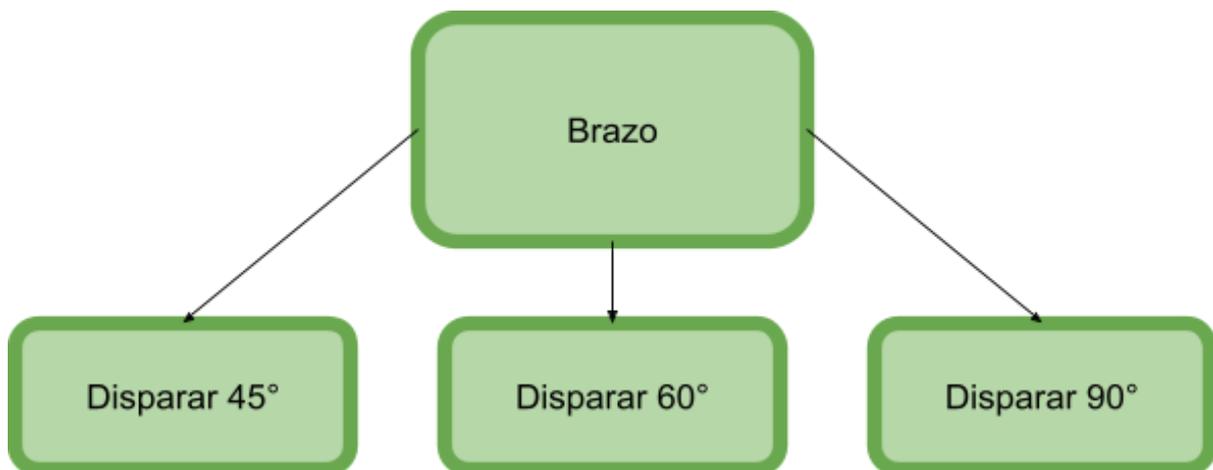
```

6.3 Diagramas

Movimiento:



Golpe:



7. Resultados

7.1 Estado actual del proyecto

En estos momentos, el proyecto cuenta con:

- Las últimas versiones del robot "Pascalito", es decir, se encuentra estructuralmente casi finalizado.
- Funciones de movimiento, el robot se desplaza de manera eficaz y tiene la capacidad de realizar golpes a la pelota.
- Interfaz gráfica, La programación de la interfaz gráfica se ha realizado empleando la librería "Tkinter", lo que habilita a los usuarios para ejecutar funciones desde dispositivos compatibles con Python. En la actualidad, la interfaz cumple sus funciones básicas, exceptuando el desarrollo de ángulos.
- Servidor. Ha sido desarrollado empleando la librería "socket" de Python, permitiendo la comunicación entre el host y el cliente.
- Conexión remota.

7.2 Problemas encontrados y Solución propuesta.

Problemas encontrados	Soluciones
Problema estructural debido a que se ladea el brazo.	Reforzar o cambiar la estructura del brazo de PASCALITO
Problemas con el cambio de ángulos de disparo.	Fabricar un tee de golf para distintos ángulos.
Problema de conexión del robot al reiniciar la conexión.	Hacer un debug en el código del servidor hasta poder arreglar el bug.
Falta de poder en el disparo.	Agregar algún mecanismo con engranajes para aumentar la potencia del disparo.

8. Conclusión:

En la segunda etapa del proyecto "Pascalito", nos enfrentamos a desafíos sustanciales que requerían la reconstrucción del robot en dos ocasiones. Estos desafíos abordaron problemas con la funcionalidad del robot, errores en las piezas y desequilibrios que impactaron su rendimiento. A pesar de estos obstáculos, nuestro equipo demostró una notable capacidad para resolver problemas y adaptarse a las circunstancias.

Esta experiencia nos impulsó a reevaluar y perfeccionar nuestra organización y enfoque de trabajo en equipo. Mediante una colaboración más estrecha y una comunicación mejorada, superamos las dificultades encontradas y avanzamos hacia la culminación exitosa del proyecto. Esta fase de ajuste no solo fortaleció nuestras habilidades técnicas, sino que también destacó la importancia de la flexibilidad y la adaptabilidad en entornos desafiantes, subrayando así la resiliencia y la capacidad de aprendizaje continuo de nuestro equipo

9. Referencias:

<https://cl.talent.com/es/salary>

<http://pomerape.uta.cl/redmine/projects/grupo-6-a-2022>