

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA

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



**Plan de Proyecto
Lanzamiento de golf
“Princesita”**

**Alumno(os): Tihare Cabello Diaz
Cristian Huanca Calle
Melisa Huanca Laura
Liliana Gálvez Yanqui
Byron Santibáñez Fernández**

Asignatura: Proyecto I

Profesor: Humberto Urrutia López

09 – 2023

Historial De Cambios

Fecha	Versión	Descripción	Autor(es)
18/08/2023	1.0	Concepción del Documento	Tihare Cabello
23/08/2023	1.1	Recopilación de Datos	Melisa Huanca Tihare Cabello
02/08/2023	1.2	Versión Preliminar del Formato	Melisa Huanca
14/08/2023	1.3	Revisión y Finalización del Informe	Melisa Huanca Tihare Cabello Liliana Gálvez Cristian Huanca Byron Santibáñez
28/10/2023	2.0	Revisión y análisis del informe anterior	Cristian Huanca
13/11/2023	2.1	Creación del informe II	Cristian Huanca
25/11/2023	2.2	Formato del informe II	Cristian Huanca Melisa Huanca
08/01/2024	3.0	Incorporaciones al Informe Final	Cristian Huanca Melisa Huanca

Tabla de Contenidos

1. Panel 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.....	5
1.4. Entregables	5
2. Organización del Personal	6
2.1. Descripción de los Roles	6
2.2. Personal que Cumplirá los Roles	6
2.3. Métodos de Comunicación	6
3. Planificación del Proyecto	7
3.1. Actividades.....	7
3.2. Carta Gantt.....	9
3.3. Gestión de Riesgos	10
4. Planificación de los Recursos	12
4.1. Hardware.....	12
4.2. Software	12
4.3. Estimación de Costos.....	12
5. Análisis – Diseño	14
5.1. Especificación de requerimientos	14
5.2. Arquitectura.....	14
5.3. Interfaz	15
6. Implementación	15
6.1. Fundamentos de proyectiles.....	15
6.2. Descripción de los programas	17
6.3. Diagramas.....	24
7. Resultados	24
7.1. Estado actual del proyecto	24
7.2. Problemas encontrados y soluciones propuestas	25
8. Pruebas	26
8.1. Descripción de las pruebas realizadas	26
8.2. Resultado de las Pruebas	26
9. Conclusión	27
10. Referencias	28

1. Panel General

1.1. Introducción

El presente proyecto se centra en la estructura organizativa empleada para alcanzar objetivos académicos de manera colaborativa, proporcionando a los estudiantes una valiosa experiencia en el campo de la ingeniería. Para este fin, se hizo uso del set educativo de LEGO Mindstorms Education EV3 para desarrollar un robot con la capacidad de simular un lanzamiento de golf y desplazarse mediante algoritmos interactivos personalizados.

Este informe detalla exhaustivamente la dinámica de equipo, describiendo el enfoque adoptado y las estrategias planificadas para lograr los objetivos establecidos. A medida que fue avanzando el proyecto, se llegó a un modelo del robot que cumplía con los objetivos, dando paso así al desarrollo de la implementación de una interfaz que se conecta mediante un servidor al robot. Este paso permitió realizar un análisis detallado del lanzamiento parabólico, lo que cumplió con los requisitos propuestos.

A lo largo de las siguientes secciones, se analizará en profundidad cada fase del proyecto, desde el diseño y la implementación del robot hasta la realización del análisis del lanzamiento parabólico, subrayando los métodos, herramientas y resultados obtenidos en cada etapa.

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar un robot EV3 capaz de desplazarse y simular un lanzamiento de golf mediante una interfaz gráfica programada en Python.

1.2.2. Objetivos Específicos

- Experimentar con el Set de Lego Mindstorms Ev3 para construir el robot.
- Ensamblar un modelo con estabilidad, movilidad y un mecanismo para realizar el lanzamiento.
- Estudiar el sistema operativo Linux y la librería de Python EV3, investigando e implementando la instalación de ev3dev.
- Estudiar la librería Tkinter para diseñar una interfaz gráfica amigable para el usuario

1.3. Restricciones

- Se debe programar exclusivamente en Python.
- Se debe utilizar el sistema operativo Linux.
- Solo se permite el uso de la plataforma Redmine para documentos y seguimiento del proyecto.
- Obligatorio utilizar el Set de Lego Mindstorms EV3.
- Restricción de tiempo asignado al proyecto.
- El equipo está limitado a un máximo de 5 integrantes.
- Disponibilidad necesaria del robot para codificación y pruebas.
- El robot debe ser capaz de realizar un golpe que genere un ángulo al momento del disparo.

1.4. Entregables

- Informes del proyecto:
 - Formulación del proyecto.
 - Avance del proyecto.
 - Informe final del proyecto.
- Presentaciones del proyecto:
 - Formulación del proyecto.
 - Avance del proyecto.
 - Presentación final del proyecto.
- Bitácoras
- Carta Gantt
- Wiki
- Manual de usuario
- Interfaz
- Robot

2. Organización del Personal

La coordinación dentro de un equipo resulta fundamental para avanzar en cualquier proyecto; por ende, es crucial distribuir las responsabilidades de manera efectiva para alcanzar con éxito los objetivos planteados.

2.1. Descripción de los Roles

Jefe de proyecto: Representante del equipo, supervisa y organiza el progreso del proyecto.

Ensamblador: Encargado del montaje y el armado de las piezas, monitorea el cumplimiento de las funcionalidades del robot, en conjunto con el programador.

Programador: Encargado del área de la codificación y funcionamiento del robot, en colaboración del ensamblador.

Documentador: Encargado de registrar el avance del proyecto, junto con la redacción de los informes.

Diseñador: Encargado de la creación del logotipo, la interfaz gráfica y presentaciones del proyecto.

2.2. Personal que Cumplirá los Roles

Jefe de proyecto	Ensamblador	Diseñador	Programador	Documentador
Melisa Huanca	Liliana Gálvez	Byron Santibáñez	Tihare Cabello	Cristian Huanca

2.3. Métodos de Comunicación

Utilizaremos diversos medios de comunicación para facilitar la interacción y el desarrollo del proyecto. Entre ellos se encuentran: WhatsApp, destinado a la mensajería a través de los grupos de la plataforma; Discord, empleado para realizar reuniones mediante sus canales de voz; y Trello, utilizado como plataforma de gestión y organización de documentos y tareas.

3. Planificación del Proyecto

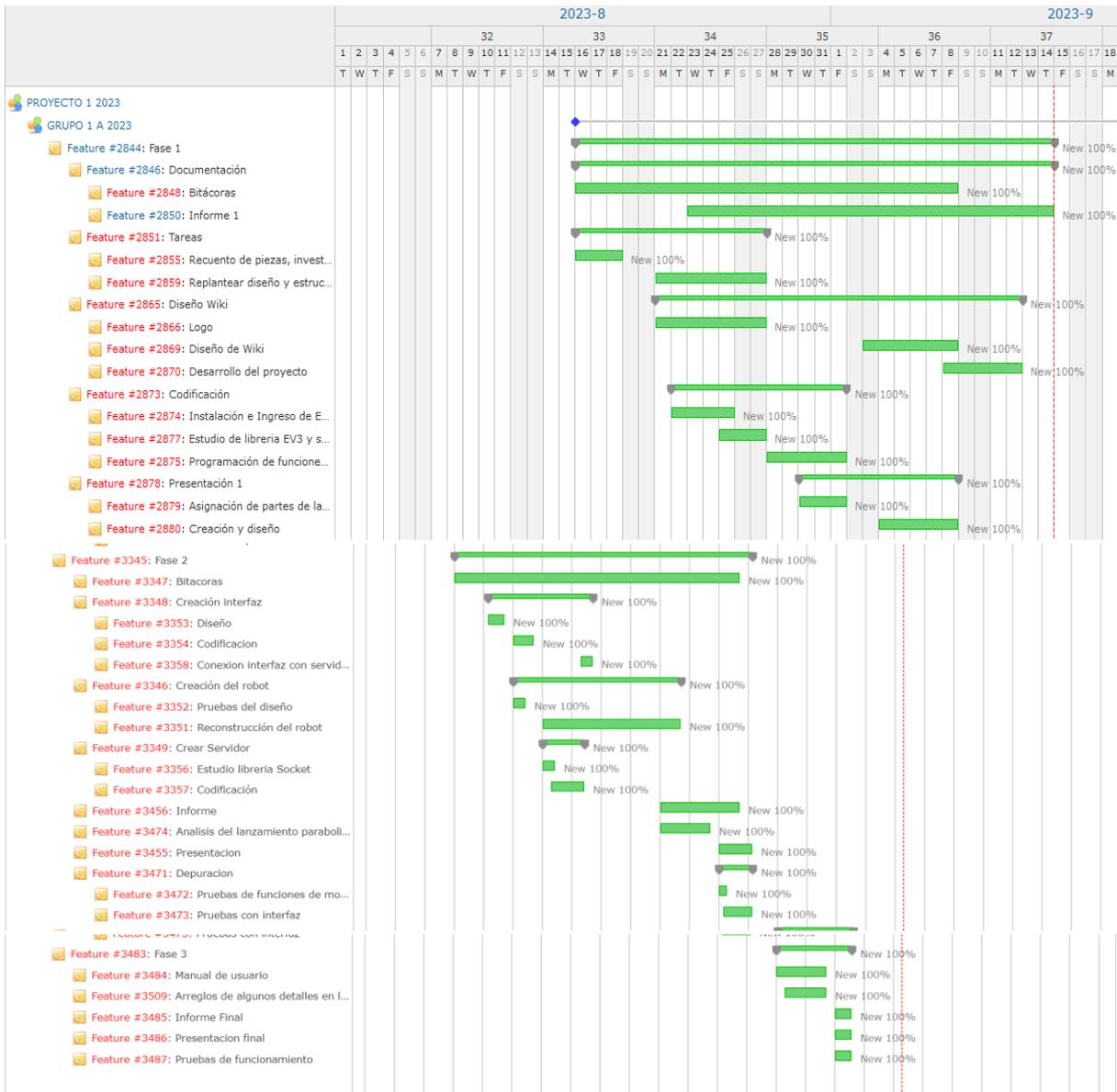
3.1. Actividades

Nombre	Descripción	Responsables	Producto
Formulación del proyecto	Se hace un recuento de las piezas que tiene el Set de Lego Mindstorms Ev3.	Todo el grupo.	Concretado
Avance en el modelo del robot	Se hace el primer modelo del robot que servirá de base de acuerdo a la lluvia de ideas del equipo.	Todo el grupo.	Concretado
Experimentación con el robot	Pruebas de movilidad y programación.	Cristian Huanca Byron Santibáñez	Concretado
Programación de movimientos	Se identifican los movimientos que realizará el robot.	Cristian Huanca Byron Santibáñez	Concretado
Concepción del primer informe	Se empieza el desarrollo del informe.	Melisa Huanca Tihare Cabello	Concretado
Realización de la 1ra presentación	Se realiza la presentación apoyándose del material brindado en el informe.	Liliana Gálvez Melisa Huanca	Concretado
Estimación de costos	Calcular el presupuesto del proyecto.	Liliana Gálvez	Concretado
Comienzo de registros en la wiki	Se capturan y comparten ideas e información del proyecto.	Liliana Gálvez	Concretado

Modificación Carta gantt	Planificación de las actividades a lo largo del semestre.	Cristian Huanca	Concretado
Arquitectura del robot	Se realiza el armado de acuerdo a un modelo propuesto.	Liliana Gálvez	Concretado
Redacción de las bitácoras	Documento donde se registra lo trabajado en la semana.	Cristian Huanca	Concretado
Videos y fotos	Registro visual del avance del proyecto.	Liliana Gálvez	Concretado
Modificación de la arquitectura del robot	Se realizan cambios por problemas con el diseño.	Liliana Gálvez	Concretado
Codificación del servidor	Encargado de la comunicación entre la interfaz y el robot.	Tihare Cabello	Concretado
Actualización de la wiki	Se agregan nuevos aspectos desarrollados en la 2da fase del proyecto.	Cristian Huanca Byron Santibáñez	Concretado
Conexión	Establecer la conexión con la interfaz y el robot.	Tihare Cabello	Concretado
Depuración de las funciones	Prueba de funcionamiento correcto de las funciones.	Tihare Cabello	Concretado
Documentación de funciones	Documentar las funciones indicando su propósito principal.	Tihare Cabello	Concretado
Codificación de la interfaz	Interfaz gráfica con la que interactúa el cliente.	Byron Santibáñez	Concretado
Elaboración del 2do informe	Se agregan aspectos trabajados en la segunda fase.	Cristian Huanca	Concretado
Creación de la 2da presentación	Presentación que muestra los puntos	Byron Santibáñez	Concretado

	más importantes en esta fase.		
Estudio de la librería socket	Estudiar las características necesarias de la librería para el desarrollo del servidor.	Tihare Cabello	Concretado

3.2. Carta Gantt



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.

Riesgo	Probabilidad de Ocurrencia	Nivel de Impacto	Acción Remedial
Desarme del robot a causa de una caída	50%	3	Se comunica al ensamblador y hacemos uso de fotografías que contienen la estructura del robot.
Daño o pérdida de la tarjeta micro SD	40%	1	Se informa al personal a cargo de los implementos, comprar tarjeta micro SD de así ser necesario.
Escasez de Piezas	70%	4	Requerir al personal a cargo las piezas faltantes.
Enfermedad o inconveniente de personal	70%	4	Reorganización del personal para cubrir al integrante faltante.
Reconstrucción total del robot por	50%	2	Diseñar un nuevo robot e investigar para un modelo más efectivo, que cumpla

incompatibilidad de lo requerido en proyecto			con el objetivo.
Horas autónomas escasas(GRUPO)	80%	3	Organización de personal específico para encontrar horarios disponibles y aprovechamiento máximo del horario de clases.
Descarga de batería del EV3	80%	4	Se solicita cargador del EV3, y proceder con su reposición de energía para su funcionamiento.
Error en la codificación	60%	3	Enmendar código e investigar fallos, para así acertar con la variación del código.
Incumplimiento de tareas	40%	2	Reorganización para priorizar tareas atrasadas y volver a un curso normal.
Recibir equipo defectuoso	10%	1	Comunicar inmediatamente al personal a cargo y solicitar cambios del material entregado o una solución a ello.
Personal conflictivo o bajo rendimiento	30%	2	Reunirse con el equipo de trabajo para atender ideas o solicitudes necesarias para llegar a un consenso, manteniendo un buen ambiente laboral. Es posible que haya un cambio de roles.
Uno o más miembros dejan el proyecto	10%	1	Reorganizar roles y tareas.
Interrupción de estudios académicos	20%	2	Seguir con el avance en la medida de lo posible, estableciendo horarios para el desarrollo del proyecto.

4. Planificación de los Recursos

4.1. Hardware

- Set Lego Mindstorm EV3.
- Micro SD, del set de Lego Mindstorm, en el cual se podrán ejecutar las instrucciones del robot. (micro Python)
- Computadores con las herramientas necesarias para el desarrollo del proyecto.

4.2. Software

- Sistema operativo Linux para programar las funciones del robot.
- Trello: plataforma para la organización del proyecto.
- Visual Studio Code: editor de código.
- Canva: herramienta de diseño.

4.3. Estimación de Costos

Costo de Hardware:

Producto	Precio
Set Lego Mindstorm(EV3)	\$ 765.650
Notebook G513IC-HN073W / 15.6" FHD / R7-4800H / Win 11 / 8 GB RAM / 512 GB SSD / NVIDIA® GeForce RTX™ 3050 Laptop GPU / Eclipse Gray.	\$ 979.990
Acer Nitro / Ryzen 7 / 12GB RAM / 512GB SSD / 15.6" FHD / NVIDIA GEFORCE RTX3050.	\$ 849.990
Notebook Ideapad Gaming 3 Intel Core i7-10750H 5.0Gh / 8Gb / 512GB SSD / 15.6"Fhd / Negro Onyx.	\$ 779.990
Notebook HP Spectre x360 Convertible / Intel® Core™ i5 de 11.ª generación / 8 GB	\$ 999.990

/ 512 GB.	
Notebook Toshiba Tecra Z40 C1410LA P/N PT463U-07P01Y	\$ 899.990
Piezas extra	\$ 62.990
Micro SD	\$ 11.990
Total	\$ 5.350.580

Costo de Software:

Producto	Precio
Licencia de Canva / 5 meses	\$ 39.500
Licencia Microsoft Office	\$ 31.500
Total	\$ 71.000

Costo de Trabajador:

Rol	Horas	Horas Extra	Precio / Hora
Jefe de proyecto	72 horas	17 horas	\$ 30.000
Programador	72 horas	15 horas	\$ 25.000
Ensamblador	72 horas	15 horas	\$ 24.000
Diseñador	72 horas	15 horas	\$ 23.000
Documentador	72 horas	12 horas	\$ 23.000
Total	-	-	\$ 10.866.000

Destacado:

- *La contabilización de las horas trabajadas comienza a partir de la formación del grupo de trabajo.*
- *Para la categorización de las horas de trabajo, se tuvo en cuenta el tiempo de trabajo en clases.*
- *Para la categorización de las horas extras, se tuvo en cuenta el tiempo en las que se trabajó fuera del horario de clase, pero dentro del mismo departamento.*

Costo total:

Costo Hardware	\$ 5.350.580
Costo Software	\$ 71.000
Costo Empleados	\$ 10.866.000
Total	\$ 16.287.580

5. Análisis – Diseño

5.1. Especificación de requerimientos

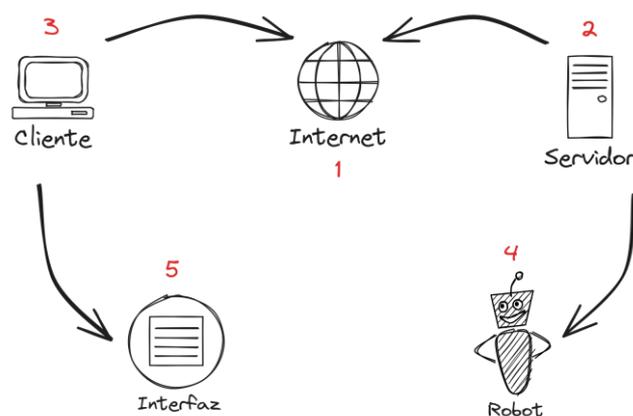
5.1.1 Requerimientos funcionales

- Desarrollar un robot que se comuniqué vía wifi y permita al usuario controlarlo mediante una interfaz gráfica en Python.
- Capacidad para moverse en direcciones hacia adelante, atrás, izquierda y derecha.
- La interfaz gráfica debe ofrecer opciones específicas para acciones como desplazarse, mover el soporte de la bola y realizar el lanzamiento.

5.1.2 Requerimientos no funcionales

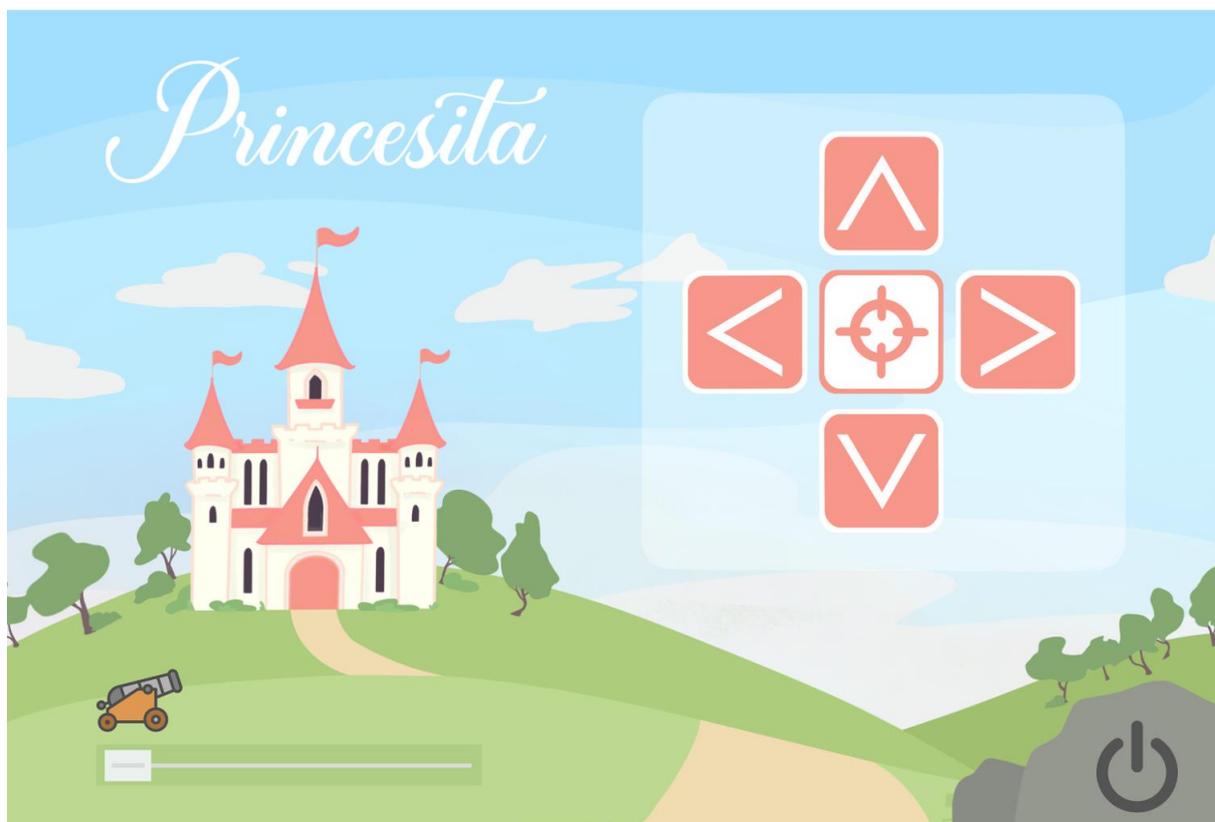
- El proyecto debe incluir un manual detallado con instrucciones completas sobre el funcionamiento integral del robot.
- La interfaz gráfica debe contar con botones específicos para controlar el desplazamiento del robot, una sección para ajustar la rotación del palo de golf y un botón para controlar la posición de la base que sostiene la bola.

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



6. Implementación

6.1. Fundamentos de proyectiles

Tenemos conocimiento de que la pelota realiza un movimiento parabólico. A continuación, se utilizarán las fórmulas correspondientes para calcular el tiempo de vuelo del proyectil. Cabe recalcar, que para este ejercicio, la aceleración de gravedad tiene un valor aproximado de 9.8 m/s

- 1) Calculamos la velocidad con la que el palo golpea la bola usando los conceptos de energía

$$h = 0.25m$$

$$\begin{aligned} E_{ma} &= E_{mb} \\ mgh &= mv^2/2 \\ v &= (2gh)^{1/2} \\ v &= 2.213 \text{ m/s} \end{aligned}$$

- 2) Aplicando el conceptos de colisiones podemos saber la velocidad de la bola al momento de que el palo impacta con esta

$$\begin{aligned} M_{pelota} &= 0.05Kg \\ M_{palo} &= 0.1Kg \end{aligned}$$

$$\begin{aligned} P_i &= P_f \\ M_{palo} * V_{palo_i} + M_{pelota} * V_{pelota_i} &= M_{palo} * V_{palo_f} + M_{pelota} * V_{pelota_f} \\ M_{palo} * V_{palo_i} &= M_{palo} * V_{palo_f} + M_{pelota} * V_{pelota_f} \\ V_{palo_f} &= V_{pelota_f} = V_f \\ V_f &= M_{palo} * V_{palo_i} / (M_{palo} + M_{pelota}) \\ V_f &= 1.25 \text{ m/s} \end{aligned}$$

- 3) Ahora aplicamos los conceptos para analizar un movimiento parabólico con los datos obtenidos

$$t = 1s \text{ (tiempo experimental)}$$

$$X_o = 0m$$

$$Y_o = 0.2m$$

$$X = 1.2m \text{ (distancia experimental)}$$

$$V_o = 1.25m$$

$$\begin{aligned} X &= X_o + V_o * \cos(a) * t \\ \cos(a) &= X / (V_o * t) \end{aligned}$$

$$a = 16.26^\circ$$

$$V_{yf} = 0 \text{ (obtener tiempo que tarda en llegar a la altura máxima)}$$

$$0 = V_o * \text{sen}(a) - g * t$$

$$t = V_o * \text{sen}(a) / g$$

$$t = 0.04s$$

$$Y_f \text{ (altura máxima que alcanza)}$$

$$Y_f = Y_o + V_o * \text{sen}(a) * t - gt^2/2$$

$$Y_f = 0.2 + 1.25 * \text{sen}(16.26^\circ) * 0.04 - 4.9 * (0.04)^2$$

$$Y_f = 0.3m$$

6.2. Descripción de los programas

Servidor

```
1 import socket
2 from robot import Robot
3
4 server = socket.socket()
5
6 PORT = 8800
7
8 server.bind(('',PORT))
9 server.listen(1)
10
11
12 connection, address = server.accept()
13
14
15 robot = Robot()
16
17 actions = {
18     "w" : robot.move_front,
19     "a" : robot.move_left,
20     "d" : robot.move_right,
21     "s" : robot.move_back,
22     "e" : robot.move_punch
23 }
24
25 while True:
26     data = connection.recv(1)
27     keyword = data.decode("utf-8")
28
29     if keyword in actions:
30         actions[keyword]()
31
32     elif keyword == "q":
33         break
34
```

Funciones

```
1 # Función para desactivar completamente el robot
2 def off_robot(self):
3     self.out_a.stop()
4     self.out_b.stop()
5     self.out_c.stop()
6     self.out_d.stop()
7
8 # Función para mover el robot hacia adelante
9 def move_front(self):
10    self.out_b.on_for_rotations(speed = 100, rotations= -2, brake = True, block = True)
11    self.out_b.stop()
12
13 # Función para mover el robot hacia atras
14 def move_back(self):
15    self.out_b.on_for_rotations(speed = 100, rotations= 2, brake = True, block = True)
16    self.out_b.stop()
17
18 # Función para mover las ruedas del robot hacia la derecha
19 def move_right(self):
20    if(self.MIN_VALUE_EJE < self.eje): # Se comprueba el eje de las ruedas para no provocar una falla en el robot / para que no se queden pegadas
21        self.out_c.on_for_degrees(speed = 100, degrees = -1.4, brake = True, block = True)
22        self.eje=-1.4
23        self.contaux1-=1
24        print(self.eje)
25        self.out_c.stop()
26
27 # Función para mover las ruedas del robot hacia la izquierda
28 def move_left(self):
29    if(self.MAX_VALUE_EJE > self.eje): # Se comprueba el eje de las ruedas para no provocar una falla en el robot / para que no se queden pegadas
30        self.out_c.on_for_degrees(speed = 100, degrees = 1.4, brake = True, block = True)
31        self.eje+=1.4
32        self.contaux2+=1
33        print(self.eje)
34        self.out_c.stop()
35
36 # Función para ingresar los grados para hacer el golpe
37 def move_punch(self, degrees):
38    self.degrees = degrees
39    self.out_d.on_for_degrees(speed = 100, degrees=degrees)
40    self.out_d.stop()
41
42 # Función para mover la base de la pelota, hacia adelante
43 def mov_adel(self):
44    if(self.MAX_VALUE_BASE>self.base): # Se comprueba la base de la pelota para que no falle el robot / en este caso para que no se golpee hacia atras
45        self.out_a.on_for_rotations(speed = 100, rotations=-0.0116, brake = True, block = True)
46        self.base += 0.0116
47        self.out_a.stop()
48
49 # Función para mover la base de la pelota, hacia atras
50 def mov_atras(self):
51    if(self.MIN_VALUE_BASE<self.base): # Se comprueba la base de la pelota para que no falle el robot / en este caso es para que no se pase hacia adelante
52        self.out_a.on_for_rotations(speed = 100, rotations=0.0116, brake = True, block = True)
53        self.base -= 0.0116
54        self.out_a.stop()
55
```

Interfaz

```

1  import tkinter as tk
2  from tkinter import ttk
3  import socket
4
5  class App(tk.Tk):
6      STATUS_CONNECTION = False
7      PORT = 8800
8
9      def __init__(self):
10         super().__init__()
11         self.title("Princesita")
12         self.config(padx=10, pady=10, bg="#FFC0CB")
13         self.geometry("800x600")
14         self.canvas = tk.Canvas(self, width=800, height=600)
15         self.canvas.pack()
16         self.resizable(width=False, height=False)
17         self.imagen_fondo = tk.PhotoImage(file="fondo.png")
18         self.imagen_fondo = self.imagen_fondo.subsample(5,5)
19         self.canvas.create_image(0, 0, anchor=tk.NW, image=self.imagen_fondo)
20         self.init_images()
21         self.init_elements()
22
23
24     def init_images(self):
25         imagenes = ["encendido.png", "deslizador.png", "fondo.png"]
26         self.images = [tk.PhotoImage(file=imagen).subsample(12, 12) for imagen in imagenes]
27         self.connect, self.deslizador, self.fondo = self.images
28         imagenes1 = ["flechaFront.png", "flechaBack.png", "flechaLeft.png", "flechaRight.png", "punteria.png"]
29         self.images1 = [tk.PhotoImage(file=imagen).subsample(25, 25) for imagen in imagenes1]
30         self.flechaFront, self.flechaBack, self.flechaleft, self.flechaRight, self.punteria = self.images1
31

```

```

32     def init_elements(self):
33
34         self.btn_Front = tk.Button(
35             self,
36             image=self.flechaFront,
37             bd=0,
38             bg="white",
39             highlightbackground='white',
40             width=80,
41             height=80,
42             command=lambda: self.move_front_car(None)
43         ).place(x=545,y=100)
44
45         self.btn_Back = tk.Button(
46             self,
47             image=self.flechaBack,
48             bd=0,
49             width=80,
50             height=80,
51             command=lambda:self.move_back_car(None)
52         ).place(x=545,y=300)
53
54         self.btn_Left = tk.Button(
55             self,
56             image = self.flechaLeft,
57             bd=0,
58             width=80,
59             height=80,
60             command=lambda:self.move_left_car(None)
61         ).place(x=450,y=200)
62
63

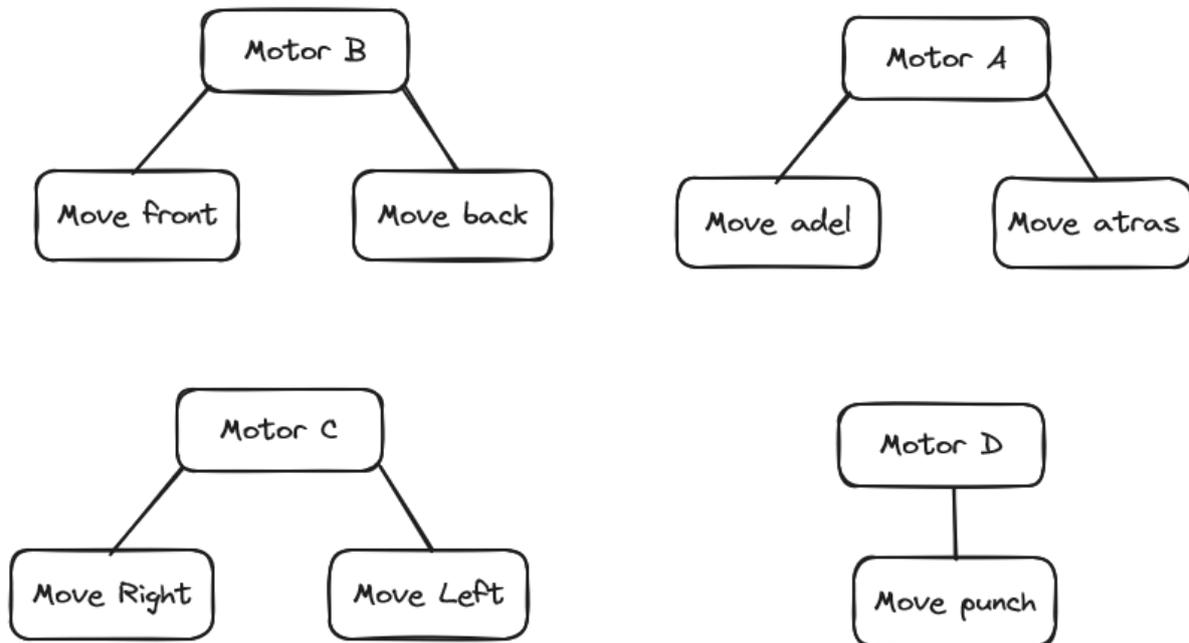
```

```
64         self.btn_Right = tk.Button(  
65             self,  
66             image= self.flechaRight,  
67             bd=0,  
68             width=80,  
69             height=80,  
70             command=lambda: self.move_right_car(None)  
71         ).place(x=640,y=200)  
72  
73         self.btn_Punch = tk.Button(  
74             self,  
75             image=self.punteria,  
76             bd=0,  
77             width=80,  
78             height=80,  
79             command=""  
80         ).place(x=545,y=200)  
81  
82         self.btn_Connect = tk.Button(  
83             self,  
84             image=self.connect,  
85             bd=0,  
86             bg="Red",  
87             width=40,  
88             height=40,  
89             command=self.ip_connect  
90         ).place(x=710,y=500)  
91  
92         self.label_deslizador = tk.Label(  
93             self,  
94             image=self.deslizador,  
95             bd=0,  
96             bg= "#bc8c8c",  
97             highlightthickness=0,  
98             width=60,  
99             height=40,  
100         )
```

```
101     self.label_deslizador.place(x=90, y=500)
102     self.distancia = tk.DoubleVar()
103     style = ttk.Style()
104     style.configure("TScale", troughcolor="#bdc8c",background="#bdc8c", sliderthickness=20)
105
106     def update_image_position(event):
107         value = self.Scala.get()
108         x_position = self.Scala.winfo_x() + value * self.Scala.winfo_width() / self.Scala.cget("to")
109         self.label_deslizador.place(x=x_position - self.label_deslizador.winfo_reqwidth() / 1.5, y=490)
110     self.Scala = ttk.Scale(
111         self,
112         orient="horizontal",
113         from_=1, to=50,
114         style="TScale",
115         length=200
116     )
117     self.Scala.place(x=90, y=540)
118
119     self.Scala.bind("<Motion>", update_image_position)
120     update_image_position(None)
121
122     def ip_connect(self):
123         if self.STATUS_CONNECTION == False:
124             ventana_conexion = tk.Toplevel(self)
125             ventana_conexion.title("Ventana de Conexión")
126             ventana_conexion.geometry("200x150")
127             ventana_conexion.resizable(width=False, height=False)
128             etiqueta = tk.Label(ventana_conexion, text="Ingrese la dirección IP:")
129             etiqueta.pack(pady=10)
130             self.IP = tk.StringVar()
131             entry_ip = tk.Entry(ventana_conexion, textvariable=self.IP)
132             entry_ip.pack(pady=10)
133             btn_conectar = tk.Button(ventana_conexion, text="Conectar", command=lambda: self.robot_connection(self.IP.get()))
134             btn_conectar.pack(pady=10)
135         else:
136             self.robot_connection(self.IP.get())
```

```
138     def robot_connection(self, IP):
139         if self.STATUS_CONNECTION == False:
140             self.client = socket.socket()
141             try:
142                 self.client.connect(("192.168.84.160", self.PORT))
143                 self.STATUS_CONNECTION = True
144                 self.init_images()
145                 self.init_elements()
146             except socket.error:
147                 print("Error. Connection T_T")
148                 self.STATUS_CONNECTION = False
149                 self.init_images()
150                 self.init_elements()
151             finally:
152                 self.init_images()
153                 self.init_elements()
154         elif self.STATUS_CONNECTION == True:
155             self.STATUS_CONNECTION = False
156             self.init_images()
157             self.init_elements()
158
159     def move_front_car(self, event):
160         self.client.send(bytes([ord('w')]))
161
162     def move_left_car(self, event):
163         self.client.send(bytes([ord('a')]))
164
165     def move_right_car(self, event):
166         self.client.send(bytes([ord('d')]))
167
168     def move_back_car(self, event):
169         self.client.send(bytes([ord('s')]))
170
171 if __name__ == "__main__":
172     app = App()
173     app.mainloop()
```

6.3. Diagramas



7. Resultados

7.1. Estado actual del proyecto

En el presente momento, el proyecto ha alcanzado los siguientes hitos:

- Finalización de la versión del robot: La construcción del robot ha sido completada en su totalidad.
- Funciones de movimiento implementadas: El robot puede desplazarse y ejecutar el golpe de la pelota simulando un "hoyo en uno", similar al movimiento en el golf.
- Interfaz gráfica desarrollada con Tkinter: Se ha creado una interfaz funcional en Python con la librería Tkinter para permitir la interacción del usuario con el robot.
- Implementación del servidor: Se ha creado un servidor denominado "Server.py" utilizando la librería de Python "socket" en el entorno EV3Dev para facilitar la comunicación.
- Conexión remota establecida: Se ha logrado la conexión remota con el robot.
- Documentación del proyecto: Se dispone de una wiki detallada que documenta los aspectos clave del proyecto.

- Actualización de la Carta Gantt: La planificación del proyecto se ha revisado y actualizado.
- Registro de actividades, informes y presentaciones: Se han generado bitácoras, informes y presentaciones para documentar y compartir el progreso y los resultados obtenidos.

7.2. Problemas encontrados y soluciones propuestas

Problemas	Soluciones
Falta de materiales	Comunicarse con el responsable de la distribución para obtener un mayor suministro.
Reconstrucción frecuente del robot	Llegar a una idea precisa que cumpla con todas las características y a su vez complete los objetivos establecidos.
Complicaciones con la estabilidad de la estructura en el diseño del robot	Reestructuración del diseño mediante ajustes en la base y fortalecimiento en los puntos débiles del robot.
Modificación frecuente de la interfaz gráfica y funciones debido a la reconstrucción del robot.	Terminar las modificaciones del robot para dar avance al desarrollo de la interfaz y funciones.

8. Pruebas

8.1. Descripción de las pruebas realizadas

- Conexión al Equipo: Se estableció la conexión del PC con el equipo ev3 para la programación con lo que se pudo cargar archivos
- Manipulación del Robot: Se establece la conexión para manipular el funcionamiento del robot y que cumpla las necesidades
- Pruebas de Rendimientos: Se realizó la prueba de la eficiencia del golpe que cumpliera con los requerimiento y poder realizar así el movimiento parabólico

8.2. Resultado de las Pruebas

Pruebas	Resultado
Conexión PC a ev3	Se realizó la conexión mediante los comando de SSH
Movimiento Parabólico	Se trabajó con el brazo del robot con el que realizaron distintas pruebas
Comunicación entre la Interfaz y el Servidor	Se probó la conexión con los cables

9. Conclusión

En esta fase crítica de nuestro proyecto, pruebas de código y al drástico cambio de nuestra interfaz, nos hemos enfrentado a una variedad de desafíos que han tenido un impacto significativo en el progreso general. A pesar de estos contratiempos, hemos logrado con éxito los objetivos que nos planteamos en el comienzo de este proyecto.

En líneas generales, a pesar de los obstáculos encontrados, nuestro equipo ha demostrado un compromiso inquebrantable con el éxito del proyecto. Estos desafíos no solo representaron obstáculos, sino también valiosas oportunidades para aprender y perfeccionar donde dirigir nuestro enfoque de manera individual y grupal. Con la finalización de esta fase, hemos alcanzado un hito importante en el proyecto, consolidando nuestra capacidad para superar adversidades y avanzar hacia la consecución de nuestras proyecciones a futuro.

10. Referencias

Página de Compra de Lego Mindstorm EV3

“Set Básico Lego Mindstorm EV3”. Amazon.com. Disponible: <http://surl.li/ldaib>

Página de Compra de Notebook Asus

“Notebook Asus G513IC-HN073W”. Lider.cl. Disponible: <http://surl.li/ldafv>

Página de Compra de Notebook HP

“Notebook HP spectre x360 convertible 14-ea0510la” HP.com Disponible: <http://surl.li/ldagg>

Página de Compra de Notebook Acer

“Notebook Acer Nitro 5” ripley.com Disponible: <http://surl.li/ldaha>

Página de Compra de Notebook Lenovo

“Ideapad Gaming 3” mercadolibre.cl Disponible: <http://surl.li/ldahk>

Página de Compra de Licencia Canva

“Licencia Canva” canva.com Disponible: <http://surl.li/ldban>