

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA

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



PROYECTO 1

“Robot Golf”

Alumnos:

- **Vaitheare Gomez A.**
- **Kary Tudela H.**
- **Sebastián Cáceres S.**
- **Rubén Salas P.**
- **Mario Villalobos A.**

Asignatura:

Proyecto I.

Profesor:

Humberto Urrutia.

Noviembre 2023

Historial De Cambios

Fecha	Versión	Descripción	Autor(es)
14/11/2023	2.1	-Pasar a 3era persona -Arreglo de redacción -Quitar el mensaje oculto	
21/11/2023	2.2	-Arreglo del formato -tipo de letra y tamaño -Uso de índice y numeración	
24/11/2023	2.3	-Arreglo de objetivos -Quitar restricciones	
26/11/2023	2.4	-Tabla de valores y costos - cambio de tiempos asignados arreglo de los entregables punto 5.1	
28/11/2023	2.5		

Índice

1. Panorama General.....	4
1.1 Introducción.....	4
1.2 Objetivos.....	5
1.2.1 Objetivo General.....	5
1.2.2 Objetivos Específicos.....	5
1.3 Restricciones.....	6
1.4 Entregables.....	7
2. Organización del personal.....	8
2.1 Descripción de roles.....	8
2.2 Personal.....	9
2.3 Comunicación.....	9
3. Planificación del proyecto.....	10
3.1 Actividades.....	10
3.2 Gantt.....	11
3.3 Gestión de Riesgos.....	12
4. Planificación de los recursos.....	13
4.1. Hardware.....	13
4.2. Software.....	13
4.3 Estimación de costos.....	14
5. Análisis y Diseño.....	17
5.1 Especificación de Requerimientos.....	17
5.1.1 Requerimientos Funcionales:.....	17
5.1.2 Requerimientos No Funcionales:.....	17
5.2 Arquitectura.....	18
5.2.1 Conexión.....	18
5.2.2 Componentes.....	19
5.2.3 Sistema de Movimiento.....	20
5.2.4 Sistema de Brazo.....	21

5.3 Interfaz de usuario.....	22
6. Implementación.....	23
6.1 Fundamento de proyectiles.....	23
6.2 Descripción de programas.....	25
6.2.1 Archivos del robot.....	25
6.2.1.1 Servidor (servidor.py).....	25
6.2.1.2 Librería de funciones (libreria.py).....	26
6.2.2 Archivos del notebook.....	27
6.2.2.1 Interfaz de usuario (interfaz.py).....	27
6.3 Diagramas.....	28
Resultados.....	29
7.1 Estado actual del proyecto.....	29
7.2 Problemas encontrados y soluciones.....	29
Conclusión.....	30

1. Panorama General

1.1 Introducción

Este informe abordará los objetivos, la planificación, el diseño del robot y la distribución de roles en el grupo, con el objetivo de finalizar el proyecto dentro del plazo establecido. También se detallarán las actividades de cada integrante, se analizarán posibles riesgos y se proporcionarán costos estimados.

El kit EV3 forma parte de un plan de estudios diseñado para enseñar habilidades básicas de programación y lógica informática, utilizando un enfoque de ingeniería robótica. Incluye 10 proyectos y un desafío final, todos centrados en conceptos clave de robótica y programación.

Este proyecto tecnológico tiene como propósito que los participantes desarrollen, en equipo y a un nivel de ingenieros principiantes, un software. La meta concreta es la creación de un robot utilizando Lego Mindstorms, una línea robótica de LEGO que combina elementos básicos de robótica, como la unión de piezas y la programación interactiva de acciones.

1.2 Objetivos

1.2.1 Objetivo General

- Construir y programar un robot capaz de movilizarse y ejecutar movimientos con un brazo funcional de tipo palo de golf, utilizando EV3 Mindstorm y el lenguaje de programación Python aplicando conocimientos de las ciencias básicas y programación.

1.2.2 Objetivos Específicos

- Comunicarse remotamente con un robot a través de instrucciones dadas por código
- Implementar librería ev3.lang de Python.
- Aplicar conocimientos adquiridos en la asignatura de Mecánica Clásica para que el robot lance un proyectil.
- Ensamblar partes del robot de tal forma que sea resistente y cumpla su objetivo.

1.3 Restricciones

Realizar el proyecto no es tan sencillo como llegar y trabajar. Un análisis previo de las cosas con las que se trabaja es esencial para evitar problemas logísticos y de funcionamiento más adelante.

Restricciones	Descripción
Conexión con el robot mediante wifi	Implica que la misma por la cual se comunica el robot en algún momento puede volverse inestable, ya sea por su distancia con la fuente o lo saturada que la señal esté
Tiempo	El límite de tiempo que tendremos para realizar la construcción y programación del robot hace ejercer más presión y precisión por lo que todo debe ser mucho más eficiente .
Trabajo y Movimiento	Como se debe trabajar con movimiento, dentro del mundo de los legos la única forma de realizarlo es usando engranajes de lego. Debido a su naturaleza, no todos los movimientos mecánicos (de la vida real) son posibles, y dependemos fuertemente de las piezas que se nos faciliten.
Sistema Operativo	Se dejó en claro que el proyecto debería ser realizado en el sistema operativo Linux, teniendo como restricción el no poder trabajar en Windows el cual nos otorga más comodidad.

1.4 Entregables

1. Panorama General
 - Introducción
 - Objetivos
 - General
 - Específicos
 - Restricciones
2. Organización del personal
 - Descripción de roles
 - Personal
 - Comunicación
3. Planificación del proyecto
 - Actividades
 - Gantt
 - Gestión de Riesgos
4. Planificación de Recursos
 - Hardware
 - Software
 - Estimación de costos
5. Análisis y Diseño
 - Especificación de Requerimiento
 - Arquitectura
 - Interfaz
6. Implementación
 - Fundamento de proyectiles
 - Descripción de programas
 - Diagramas
7. Resultados
 - Estado actual del proyecto
 - Problemas encontrados y Soluciones
8. Conclusión
9. Referencias

2. Organización del personal

2.1 Descripción de roles

La organización de este proyecto es crucial ya que tiene límite de tiempos establecido, por esta misma razón para la realización de este proyecto nos distribuiremos el trabajo entregando distintas tareas para cada integrante del grupo, teniendo en cuenta que al ser un equipo debemos velar por el apoyo mutuo en estas tareas para un mejor trabajo colaborativo, a su vez manteniendo una buena organización y productividad de cada uno de los integrantes . Los principales aspectos considerados para los roles fueron:

- Jefe de grupo:

Dentro del proyecto, es fundamental supervisar que los otros roles cumplan con su debida función además de influir en la planificación, gestión y asignación de las tareas. Al mismo tiempo, sirve de ayuda para los demás y es el encargado de poseer la visión general que mantiene alineados los quehaceres del equipo.

- Ensamblador:

Su tarea es trabajar en el código general del proyecto. Tiene que estructurar y modularizar cada una de las funciones del robot por medio de codificación en Python y entender las relaciones que existen entre los sistemas presentes..

- Programador:

Encargado de ingeniar y ensamblar cada uno de los mecanismos que conformarán al robot, siempre teniendo en cuenta las distintas ventajas y desventajas que los sistemas construidos aportan, además del mantenimiento y mejoramiento de los sistemas una vez terminados.

- Documentador:

El documentador tiene la responsabilidad de desarrollar y mantener documentación esencial, que abarca desde manuales de usuario hasta detalles técnicos y comentarios en el código. Colabora estrechamente con el equipo, asegurándose de que la información esté organizada y sea fácilmente accesible. Su labor contribuye significativamente a la continuidad y éxito del proyecto.

2.2 Personal

Trabajador	Cargo
Sebastián Cacéres	Ensamblar
Rubén Salas	Ensamblar
Mario Villalobos	Programar
Kary Tudela	Sub Jefe de proyecto
Vaitheare Gomez	Jefe de proyecto

2.3 Comunicación

La comunicación desempeña un papel crucial en un proyecto al coordinar equipos, garantizar la comprensión de requisitos y expectativas, resolver problemas de manera eficiente, adaptarse a cambios y mantener la motivación del equipo. Facilita la colaboración, reduce malentendidos y permite una respuesta rápida a desafíos, contribuyendo significativamente al éxito y la eficacia global del proyecto, Para esto usamos los siguientes medios:

- Whatsapp
- Discord
- Telegram
- Correo
- Plataforma Redmine

3. Planificación del proyecto

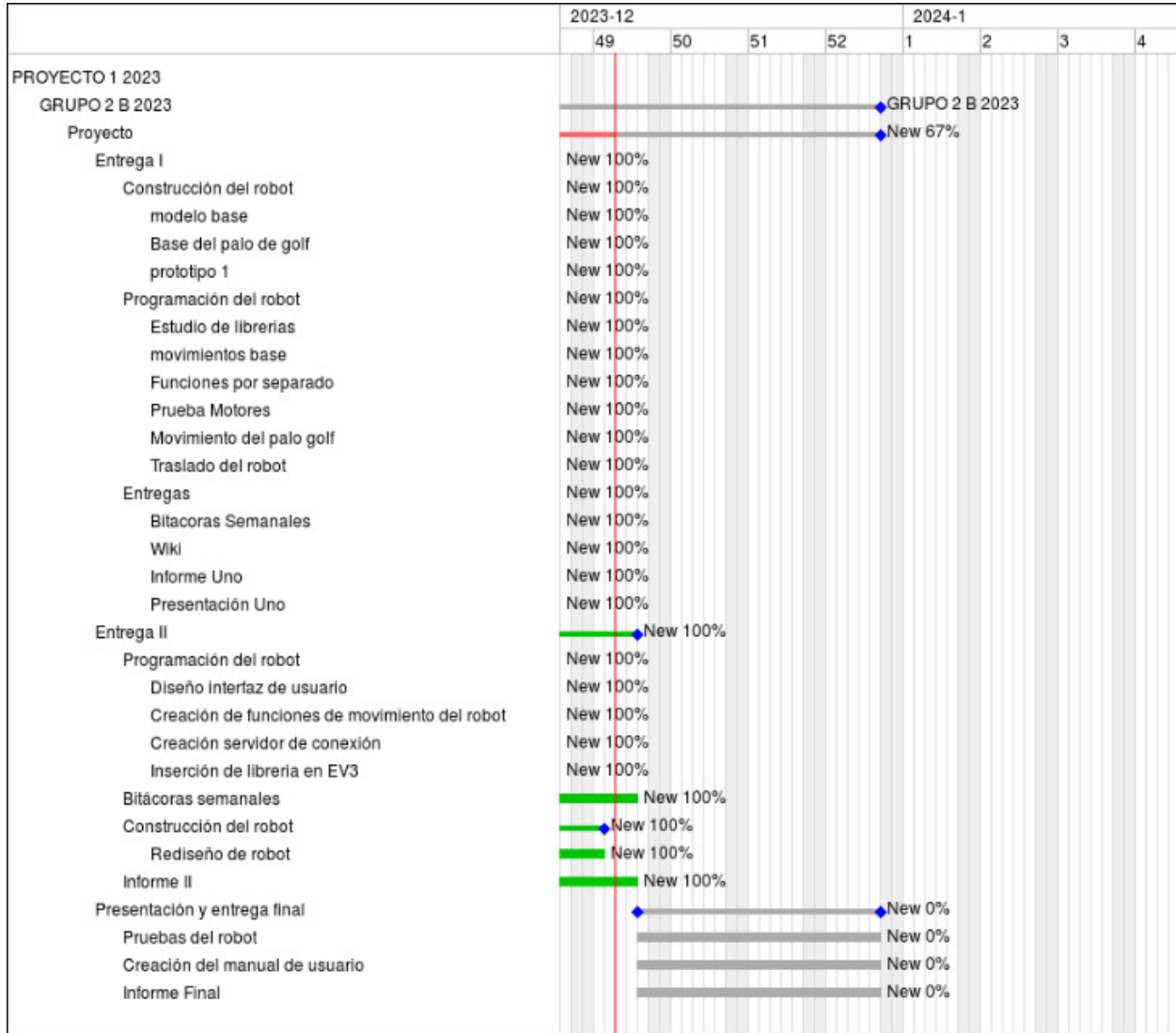
3.1 Actividades

Para llevar a cabo este proyecto se han asignado ciertas actividades para tener una correcta planificación y así cumplir nuestros objetivos a tiempo, a continuación se presentarán las actividades:

- La designación de rol de cada integrante
- El ensamblado del Ev3
- El desarrollo de la documentación del proyecto
- Codificación de un programa Python que ejecute los movimientos del robot
- Actualizaciones a lo largo del proyecto en la plataforma Redmine

Estas actividades y algunas más serán las que tendremos, y cada una de ellas podrá ser asistida por uno o más integrantes del grupo, en caso de que alguien necesite ayuda, siendo así más flexibles a la hora de progresar en cada actividad.

3.2 Gantt



3.3 Gestión de Riesgos

- 1) Catastrófico
- 2) Crítico
- 3) Circunstancial
- 4) Irrelevante

Riesgos	Probabilidad de ocurrencia	Nivel de impacto	Acción remedial
Pérdida de una pieza lego.	50%	4	Reemplazarla.
Desarme del robot a causa de una caída.	10%	2	Volver a construir el robot.
Cable de conexión dañado.	10%	3	Reemplazarlo.
Errores en la programación.	10%	3	Revisar la codificación y realizar una reconstrucción de esta misma.
Estimación del tiempo para el desarrollo.	10%	1	Mejorar la comunicación y motivar a los integrantes del grupo buscando horarios disponibles para todos.
Ausencia de algún integrante	10%	3	Los integrantes del grupo se encargarán temporalmente de las tareas del personal faltante.
Daño o pérdida de la tarjeta SD.	20%	1	Comprar otra tarjeta SD y respaldar la información.
Descarga de batería del EV3.	35%	4	Cargar la batería, mientras el EV3 no sea utilizado.

4. Planificación de los recursos

Además de la planificación de personal y de las actividades, dentro de un proyecto también se tiene que tener en cuenta los recursos de los que se dispone para trabajar. A continuación se detalla cada recurso requerido para la realización del proyecto:

4.1. Hardware

En recursos de hardware, se detalla lo siguiente:

- Kit LEGO MINDSTORMS ev3
- HP Pavilion Ryzen 5, 8GB X64
- Toshiba Core i5, 8GB
- Tarjeta Memoria Micro SD XC 8GB Kingston
- Usb Inalámbrico Rojo TP-Link
- Piezas de repuesto kit

4.2. Software

En recursos de Software, se detalla lo siguiente:

- Visual Studio Code
- Nano
- S.O. ev3_Dev
- Python

4.3 Estimación de costos

Costos de Software

Producto	Cantidad	Valor	Valor Total
Kit LEGO MINDSTORMS ev3	1	CLP \$1.200.000	CLP \$1.200.000
HP Pavilion Ryzen 5, 8GB X64	1	CLP \$799.000	CLP \$799.000
Toshiba Core i5, 8GB	Renta (31 días por 4 meses)	CLP \$4.650	CLP \$18.600
Tarjeta Memoria Micro SD XC 8GB Kingston	1	CLP \$15.900	CLP \$15.900
Usb Inalámbrico Rojo TP-Link	1	CLP \$17.600	CLP \$17.600
Piezas de repuesto kit	1	CLP \$30.000	CLP \$30.000
		Total	CLP \$2.081.100

Costos por personal

Trabajador	Cargo	Horas Mensuales	Valores Por Hora
Sebastián Cacéres	Ensamblar	15	\$32.000
	Documentar	8.5	\$28.000
	Programar	15	\$35.000
Rubén Salas	Ensamblar	18	\$32.000
	Documentar	15	\$28.000
	Programar	5.5	\$35.000
Mario Villalobos	Ensamblar	15	\$32.000
	Documentar	4.5	\$28.000
	Programar	19	\$35.000
Kary Tudela	Sub Jefe de proyecto	15	\$37.000
	Ensamblar	3.5	\$32.000
	Documentar	5	\$28.000
	Programar	15	\$35.000
Vaitheare Gomez	Jefe de proyecto	20	\$37.000
	Ensamblar	2.5	\$32.000
	Documentar	5	\$28.000
	Programar	11	\$35.000

Sueldo del Personal

Personal	Sueldo mensual	Sueldo por 4 Meses
Sebastián Cacéres	1.243.000	4.972.000
Ruben Salas	1.188.500	4.754.000
Mario Villalobos	1.271.000	5.084.000
Kary Tudela	1.332.000	5.328.000
Vaitheare Gomez	1.345.000	5.380.000
	6.379.500	25.518.000

Costo Total durante los 4 meses.

Costos Hardware	CLP \$2.081.100
Sueldo del personal	CLP \$25.518.000
Total Del Proyecto	CLP \$27.599.000

Considerando las tablas anteriores se puede obtener un estimado de **\$27.599.000 (Veintisiete millones quinientos noventa y nueve mil pesos)** destinados para la completación del proyecto

5. Análisis y Diseño

5.1 Especificación de Requerimientos

El robot deberá ser capaz de moverse y enviar una pelota de pinpon (Simulando una pelota de golf) a un objetivo determinado. Este movimiento se efectuará con el robot que simulará el palo de golf, su trayectoria tendrá que ser calculada siguiendo el modelo de lanzamiento parabólico (Ideal) visto en la asignatura de mecánica clásica.

A continuación, se mostraran los requerimientos funcionales (Tareas que debe cumplir el robot) y no funcionales (Condiciones que deben haber para que el robot pueda realizar sus tareas) que se deben cumplir para lograr el objetivo previsto:

5.1.1 Requerimientos Funcionales:

- Crear un servidor para poder comunicarse con el cliente
- Mediante una interfaz gráfica moverse por los ejes (x,y) y también golpear la pelota con cierta intensidad para la simulación del palo de golf.

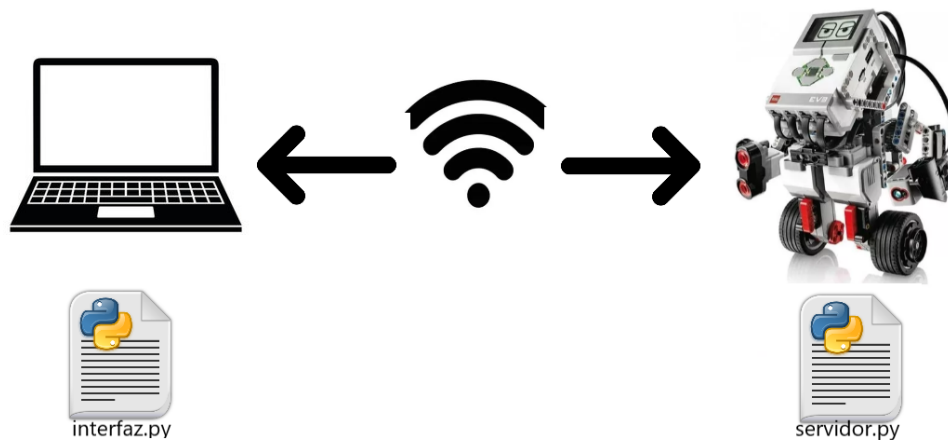
5.1.2 Requerimientos No Funcionales:

- Mecánicos: Mantener la estabilidad en todo momento al hacer que el robot cumpla sus funciones.
- Software: El robot debe estar construido únicamente por las piezas del LEGO MINDSTORMS ev3 y por las piezas adicionales prestadas y sobre la interfaz gráfica deberá ser escrita con el lenguaje de programación Python y utilizando Tkinter.

5.2 Arquitectura

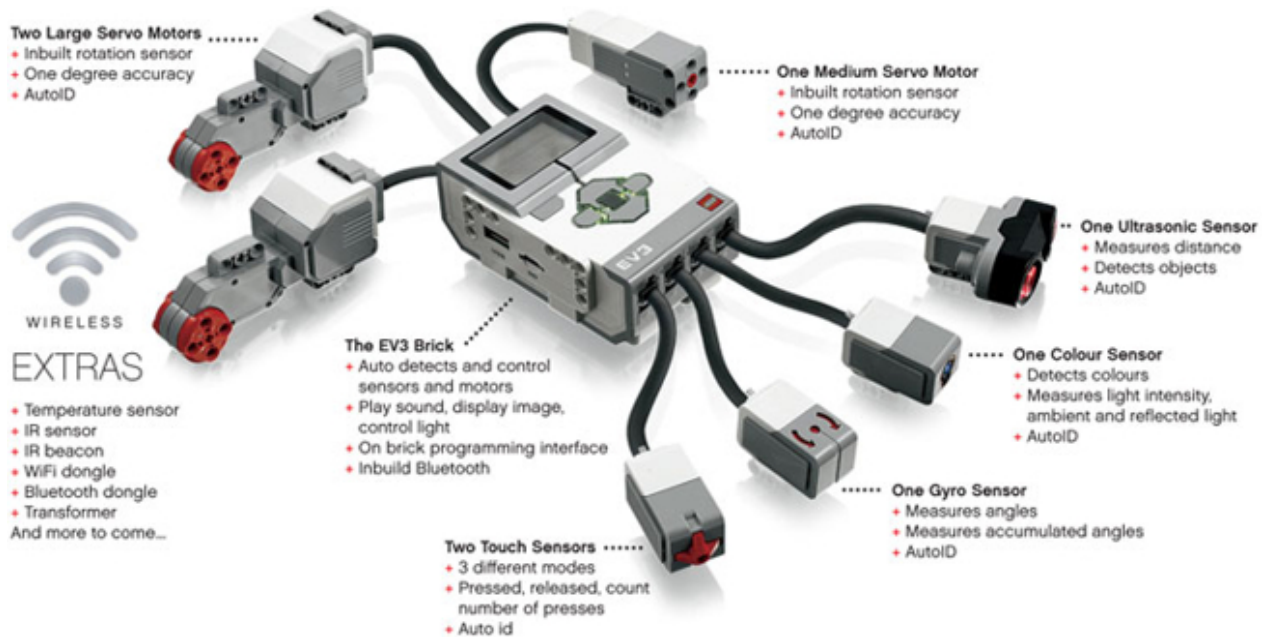
5.2.1 Conexión

Ambos, el robot y el notebook deben estar conectados a la misma red wifi, se debe ejecutar el archivo “servidor.py”, posteriormente se ejecuta la interfaz en el notebook, de esta forma se realizará la conexión remota. Una vez abierta la interfaz esta se conectará al servidor del robot y gracias a esto el usuario podrá controlar el robot desde la interfaz.



5.2.2 Componentes

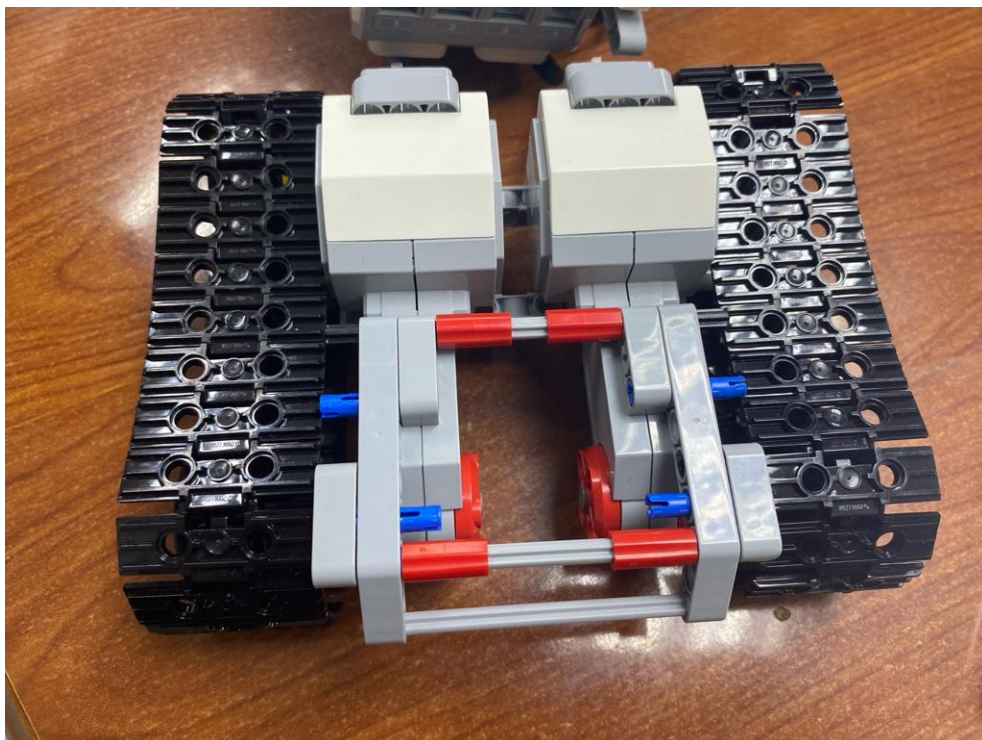
El bloque EV3 internamente se compone de un procesador ARM9, una memoria SD que soporta su sistema operativo EV3-Dev, basado en Linux; un dongle de conexión Wifi y 8 puertos RJ12 modificados para sensores y motores.



El robot usará tres motores, dos de movilidad en forma de oruga y el tercer motor es destinado al movimiento del robot que simulará el palo de golf.

5.2.3 Sistema de Movimiento

El movimiento del robot se llevará a cabo gracias a una base sólida, la cual consta de dos orugas las cuales facilitan su movimiento sobre las superficies irregulares y lisas.



5.2.4 Sistema de Brazo

La pelota de ping-pong será golpeada por el brazo del robot (que simula el palo de golf) el cual funciona con el puerto D. El robot consta de dos fases, una de carga del brazo (para la fuerza con la que golpeará la pelota y el ángulo que tendrá el palo de golf) y la fase de tiro (cuando ya golpea la pelota).

La fase de carga se considera el estado neutro del robot, inicialmente el palo de golf tomará un ángulo de inclinación de 0° , luego al salir de su estado neutro pasa a la fase de tiro donde toma un ángulo de inclinación de 75° y toma la fuerza necesaria para el golpe de la pelota.

Es importante los cambios de energía que hace el robot a lo largo de sus dos fases tanto en el robot como en la pelota como en el robot. La energía cinética y mecánica en el golf se puede entender a través de los principios básicos de la física, especialmente en el contexto de un sistema ideal sin fuerza de roce y condiciones climáticas ideales:

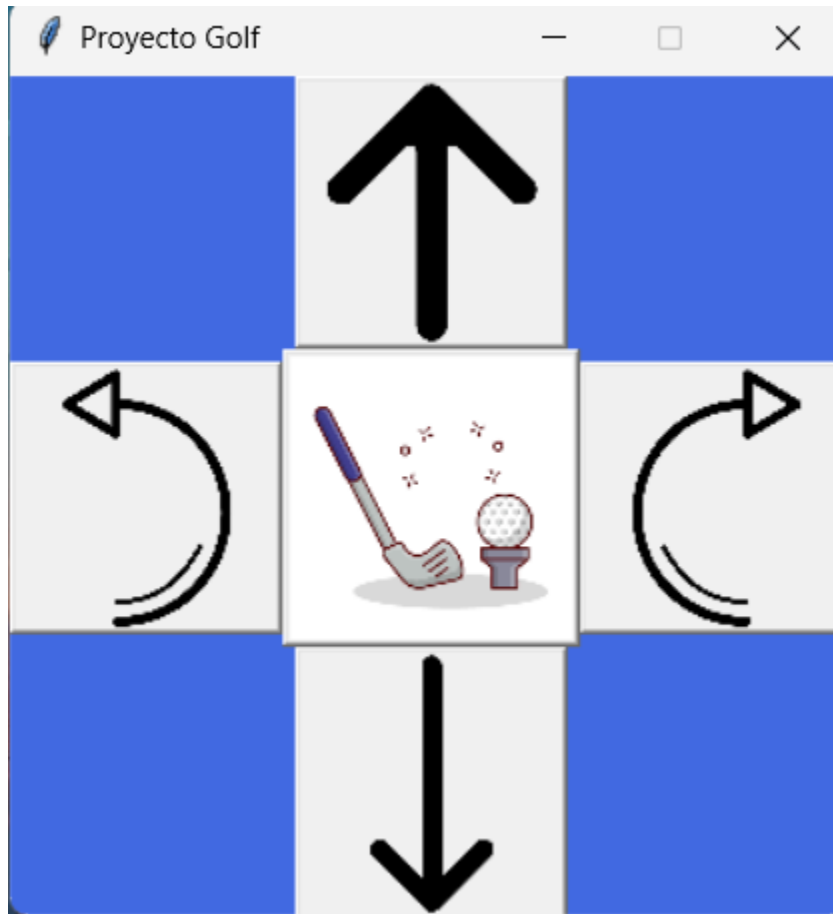
Energía Cinética en el Palo:

- Antes de realizar el golpe, el golfista potencialmente eleva el palo a una altura determinada. En este punto, el palo tiene energía potencial gravitatoria debido a su posición elevada.
- Al iniciar el swing, parte de esta energía potencial se convierte en energía cinética del palo. A medida que el palo se mueve hacia abajo, su velocidad aumenta, y por lo tanto, su energía cinética también.

Transferencia de Energía al Golpear la Pelota:

- Cuando el robot realiza el golpe, la cabeza del palo se mueve hacia la pelota. En el momento del impacto, parte de la energía cinética del palo se transfiere a la pelota. Este proceso se debe a la colisión entre la cabeza del palo y la pelota. La energía cinética del palo se transfiere a la pelota de manera eficiente si el impacto es sólido y centrado.

5.3 Interfaz de usuario



Esta es la interfaz de usuario que permitirá al “Robot Golf” moverse, diseñamos una interfaz intuitiva y amigable para el usuario, una interfaz que con solo verla una vez, se pueda deducir su funcionamiento.

La interfaz consta de 5 botones, los cuáles 4 de ellos controlan el movimiento del robot, la flechas indicarán la dirección del movimiento, esto para el caso de los botones “adelante” y “retroceder”, en cambio los botones “izquierda” y “derecha” permiten que el robot gire, y posteriormente el usuario deberá presionar el botón “adelante” o “retroceder” según sus necesidades.

Finalmente el botón del centro, que permite al robot golpear con su sistema de brazo, en este caso una pelota de ping-pong.

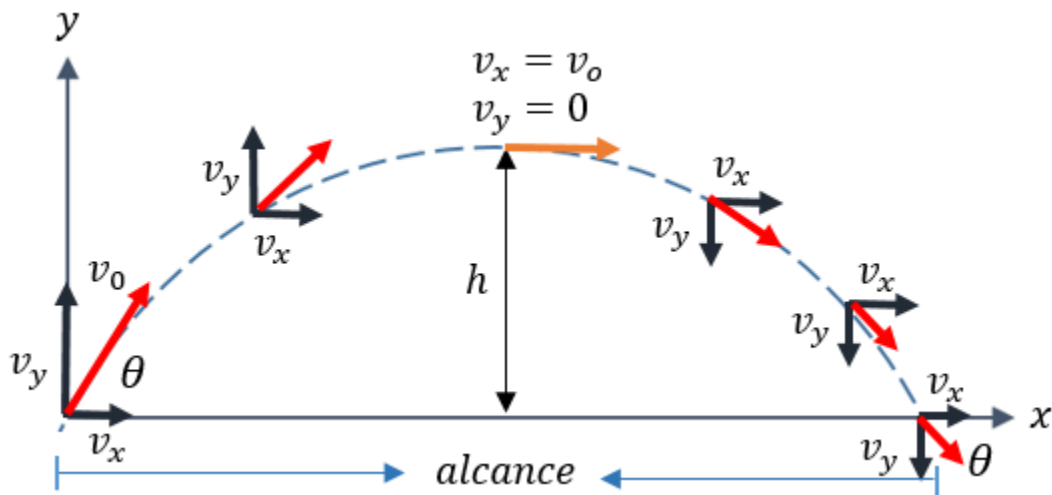
6. Implementación

6.1 Fundamento de proyectiles

Actualmente el “Robot Golf” consta de 2 golpes distintos, un golpe suave cuyo movimiento de la pelota forma un movimiento rectilíneo uniforme (MRU).

Y otro golpe más fuerte cuyo movimiento de la pelota corresponderá a una parábola, generando un movimiento rectilíneo uniforme acelerado en el eje ‘y’.

Las ecuaciones de los movimiento son las siguientes:



Movimiento rectilíneo uniforme (MRU):

a.- $x = x_0 + v_{0x} \times t$

b.- $v_x = v_0 \times \cos(\theta)$

Movimiento rectilíneo uniforme acelerado (MRUA):

c.- $y = y_0 + v_{0y} \times t + \frac{1}{2} \times a \times t^2$

d.- $v_y = v_0 \times \text{sen}(\theta) - g \times t$

Los datos que se conocen son:

$$\theta = 30^\circ$$

$$g = 9.8 \text{ m/s}^2$$

$$x = 0.3 \text{ m} \quad || \quad x_0 = 0 \text{ m}$$

$$y = 0.15 \text{ m} \quad || \quad y_0 = 0 \text{ m}$$

$$v_0 = \text{¿?} \quad || \quad t = \text{¿?}$$

De v_0 sacamos que:

$$v_{0x} = v_0 \cdot \cos(45^\circ) = v_0 \cdot \sqrt{2}/2$$

$$v_{0y} = v_0 \cdot \sin(45^\circ) = v_0 \cdot \sqrt{2}/2$$

Por lo tanto se concluye que:

$$v_{0x} = v_{0y}$$

De la ecuación 'a' se despeja v_0 :

$$x = x_0 + v_{0x} \times t \Rightarrow v_0 = 2x/t\sqrt{2}$$

Se reemplaza v_{0y} en la ecuación 'c':

$$y = y_0 + v_{0y} \times t + \frac{1}{2} \times a \times t^2 \Rightarrow 0 = v_0 \cdot \sqrt{2}/2 \cdot t - 4.9t^2$$

Reemplazamos v_0 :

$$0 = (2x/t\sqrt{2}) \cdot (\sqrt{2}/2) \cdot t - 4.9t^2$$

Al resolver tenemos que **$t \approx 0.2474$ segundos**

Y se concluye que **$v_0 \approx 1.714$ m/s**

6.2 Descripción de programas

6.2.1 Archivos del robot

6.2.1.1 Servidor (servidor.py)

Es el programa encargado de realizar la conexión mediante un servidor creado por la librería socket de Python. Este ejecuta un bucle infinito durante el cual recibe instrucciones desde la interfaz (notebook), trabaja en conjunto con la librería de funciones del robot.

```
1  from libreria import *
2  import socket
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         avanzar()
18     if (char == 's'):
19         retroceder()
20     if (char == 'd'):
21         girarDerecha()
22     if (char == 'a'):
23         girarIzquierda()
24     if (char == 'q'):
25         print("Terminando la sesion....")
26         break
27     if (char == ' '):
28         detener()
```

6.2.1.2 Librería de funciones (libreria.py)

En este archivo de python se definen las funciones responsables de interactuar con el robot. Aquí se encuentran las distintas acciones que el robot puede llevar a cabo.

```
1  #!/usr/bin/env python3
2  from ev3dev2.motor import LargeMotor, OUTPUT_A, OUTPUT_B, OUTPUT_C, OUTPUT_D, SpeedPercent, MoveTank
3
4  brazo_robot = LargeMotor(OUTPUT_D)
5  base_robot = MoveTank(OUTPUT_B,OUTPUT_C)
6
7  def avanzar():
8      print("Avanzando...")
9      base_robot.on(SpeedPercent(100),SpeedPercent(100))
10 def retroceder():
11     print("Retrocediendo...")
12     base_robot.on(SpeedPercent(-100),SpeedPercent(-100))
13 def girarDerecha():
14     print("Girando a la derecha...")
15     base_robot.on(SpeedPercent(100),SpeedPercent(-100))
16 def girarIzquierda():
17     print("Girando a la izquierda...")
18     base_robot.on(SpeedPercent(-100),SpeedPercent(100))
19 def golpear():
20     brazo_robot.on(SpeedPercent(100))
21 def detener():
22     base_robot.stop()
23     brazo_robot.stop()
```

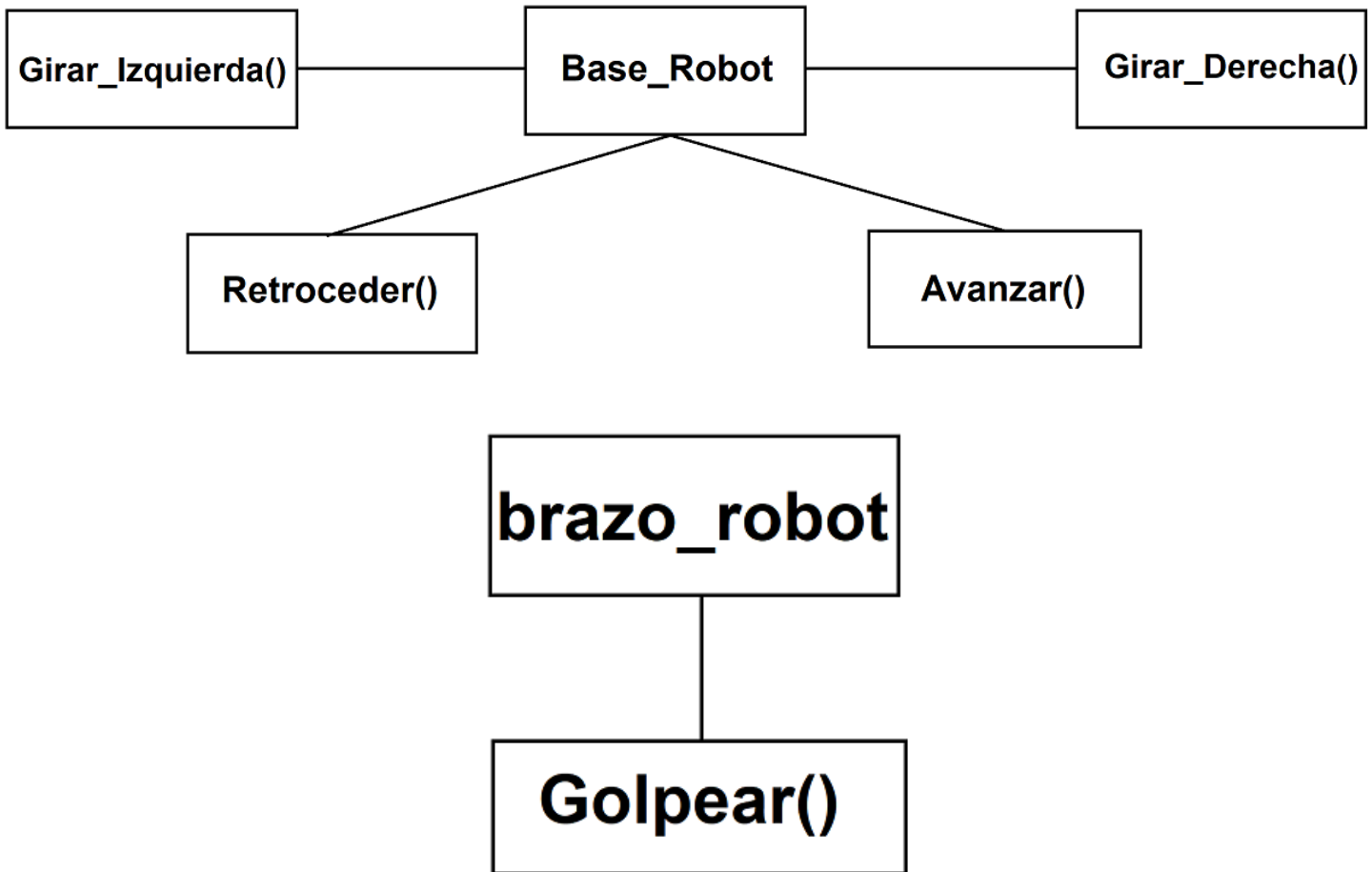
6.2.2 Archivos del notebook

6.2.2.1 Interfaz de usuario (interfaz.py)

Presenta una interfaz gráfica desarrollada mediante la librería Tkinter de Python. En esta interfaz se incluyen los botones correspondientes a los movimientos básicos del robot junto con el botón disparo.

```
1 import tkinter as tk
2 from tkinter import PhotoImage
3
4 # Función que se ejecuta cuando se presiona un botón del joystick
5 def boton_presionado(direccion):
6     print(f"Botón <direccion> presionado")
7
8 # Crear la ventana
9 ventana = tk.Tk()
10 ventana.title("Proyecto Golf")
11 ventana.geometry("335x335") # Tamaño fijo de la ventana
12 ventana.resizable(False, False)
13 ventana.configure(bg='royalblue') # Fondo de color azul real (arcoíris)
14
15 # Cargar imágenes para los botones del joystick y escalarlas
16 imagen_izquierda = PhotoImage(file="imagen_izquierda.png").subsample(5, 5)
17 imagen_arriba = PhotoImage(file="imagen_arriba.png").subsample(5, 5)
18 imagen_abajo = PhotoImage(file="imagen_abajo.png").subsample(5, 5)
19 imagen_derecha = PhotoImage(file="imagen_derecha.png").subsample(5, 5)
20 imagen_golpear = PhotoImage(file="imagen_golpear.png").subsample(2, 2)
21
22 # Configurar los botones del joystick en una cuadrícula para centrarlos
23 boton_izquierda = tk.Button(ventana, image=imagen_izquierda, command=lambda: boton_presionado("Izquierda"))
24 boton_arriba = tk.Button(ventana, image=imagen_arriba, command=lambda: boton_presionado("Arriba"))
25 boton_abajo = tk.Button(ventana, image=imagen_abajo, command=lambda: boton_presionado("Abajo"))
26 boton_derecha = tk.Button(ventana, image=imagen_derecha, command=lambda: boton_presionado("Derecha"))
27 boton_golpear = tk.Button(ventana, image=imagen_golpear, command=lambda: boton_presionado("Golpear"))
28
29 # Organizar los botones en una cuadrícula
30 boton_izquierda.grid(row=1, column=2)
31 boton_arriba.grid(row=0, column=3)
32 boton_abajo.grid(row=2, column=3)
33 boton_derecha.grid(row=1, column=4)
34 boton_golpear.grid(row=1, column=3)
35
36 # Iniciar el bucle principal de tkinter
37 ventana.mainloop()
```

6.3 Diagramas



Resultados

7.1 Estado actual del proyecto

- El robot ev3 “robot golf” en el apartado de construcción se encuentra finalizado en su totalidad.
- En cuestión de función de movimiento el robot ya cuenta con la capacidad de avanzar, retroceder , girar de izquierda a derecha y el palo de golf ya es completamente funcional.
- Carta Gantt actualizada.
- Bitácoras e informe al día.
- Wiki del proyecto aún en desarrollo, faltando por implementar evolución del proyecto entre otros detalles.
- Funcionamiento de la interfaz y diseño de la misma aún en desarrollo.
- Conexión entre el servidor y la interfaz aún en desarrollo.

7.2 Problemas encontrados y soluciones

Problemas encontrados	Soluciones
Inestabilidad del brazo del robot, el movimiento que produce el motor del brazo provoca que el brazo entero del robot tienda a moverse en exceso.	Reconstruir y añadir soportes al motor del brazo para evitar inestabilidad y un posible desacoplamiento del brazo.
Problemas de conexión wifi debido a la cantidad de personas conectados a la misma red.	Usar una red wifi local privada desde nuestros dispositivos móviles, así evitamos problemas de conexión ya sea por tráfico de datos o por interferencias de red.
Fallos en la inicialización del robot, posiblemente debido a archivos corruptos.	Formateo de la tarjeta SD y reinstalación del sistema operativo ev3dev.
Fallos a la hora de golpear la pelota de ping-pong debido a la mala estructura del brazo del robot.	Rediseño del brazo del robot, aumentando su alcance y grosor. De esta forma puede alcanzar la pelota de ping-pong sin problemas.

Conclusión

A medida que el proyecto avanzaba, surgieron diversas dificultades que, aunque representan desafíos, no obstaculizaron el progreso completo de cada etapa. Es relevante que nos encontramos ligeramente retrasados con respecto a lo esperado, los resultados obtenidos hasta ahora difieren levemente con la planificación establecida por el equipo en la fase anterior.

En la siguiente etapa del proyecto, nos enfocaremos en acercar el proyecto al usuario común. Esto implica la creación de un manual de usuario detallado y la mejora significativa de la interfaz gráfica para garantizar una experiencia más amigable y accesible, además del mejoramiento de la estructura del robot.