

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA

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



PROYECTO 1

Informe Final

“GolfBot”

Alumnos:

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

Asignatura:

Proyecto I.

Enero 2024

Profesor:

Humberto Urrutia.

Historial De Cambios

Fecha	Versión	Descripción	Autor(es)
14/11/2023	2.1	-Pasar a 3era persona. -Arreglo de redacción. -Quitar el mensaje oculto.	Rubén Salas Vaitheare Gómez
21/11/2023	2.2	-Arreglo del formato -tipo de letra y tamaño. -Uso de índice y numeración.	Mario Villalobos Sebastián Cáceres
24/11/2023	2.3	-Arreglo de objetivos. -Quitar restricciones.	Kary Tudela
26/11/2023	2.4	-Tabla de valores y costos. - Cambio de tiempos asignados arreglo de los entregables punto 5.1	Vaitheare Gómez Mario Villalobos Rubén Salas
25/12/2023	3.0	- Desarrollo informe final	Kary Tudela Sebastián Cáceres

Índice

1. Panorama General.....	5
1.1 Introducción.....	5
1.2 Objetivos.....	6
1.2.1 Objetivo General.....	6
1.2.2 Objetivos Específicos.....	6
1.3 Restricciones.....	7
1.4 Entregables.....	8
2. Organización del personal.....	8
2.1 Descripción de roles.....	8
2.2 Personal.....	9
2.3 Comunicación.....	10
3. Planificación del proyecto.....	11
3.1 Actividades.....	11
3.2 Carta 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
5.3 Sistema de Brazo.....	20
5.4 Interfaz de usuario.....	21
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.....	27
Resultados.....	28
7.1 Estado actual del proyecto.....	28
Conclusión.....	29

1. Panorama General

1.1 Introducción

EV3 es un módulo del plan de estudios diseñado para enseñar habilidades básicas de razonamiento y lógica de programación informática utilizando un contexto de ingeniería robótica. Contiene una secuencia de 10 proyectos (más un desafío final) organizados en torno a 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.

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 desarrollarán todas las bases del proyecto, pasando por todos sus respectivos procesos, los cuales son: planteamiento del problema, planificación, distribución de tareas, creación del diseño básico, armado y programación del robot, fase de pruebas y su despliegue.

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.
- 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.
- Diseñar una interfaz de usuario usando la librería Tkinter. Debe permitir al usuario controlar el robot desde el computador.

1.3 Restricciones

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.
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.

1.4 Entregables

En este proyecto, se piden como entregables los siguientes documentos:

Informes y Presentaciones: Los cuáles se dividen en las 3 etapas del proyecto. Formulación del proyecto, Ejecución del proyecto / Fase de Avance del proyecto y Fase Final del proyecto.

Bitácoras Semanales: Documentos semanales en los que se registran las actividades realizadas por el grupo durante todo el proyecto. En estas bitácoras se documentan las actividades realizadas durante la semana, como también se documentan los futuros puntos a resolver.

Wiki del Proyecto: Página alojada en 'Redmine' en donde se registra todo lo relacionado al proyecto y su desarrollo / avance.

Manual de Usuario: Documento orientado al usuario que da a conocer el funcionamiento y el uso del robot junto a la interfaz de usuario.

Robot en su estado final: Robot construido a partir del set LEGO MINDSTORMS EV3 completamente operativo y/o funcional.

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.

- **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 de proporcionar un mantenimiento adecuado al robot en caso de que sea necesario.

- **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

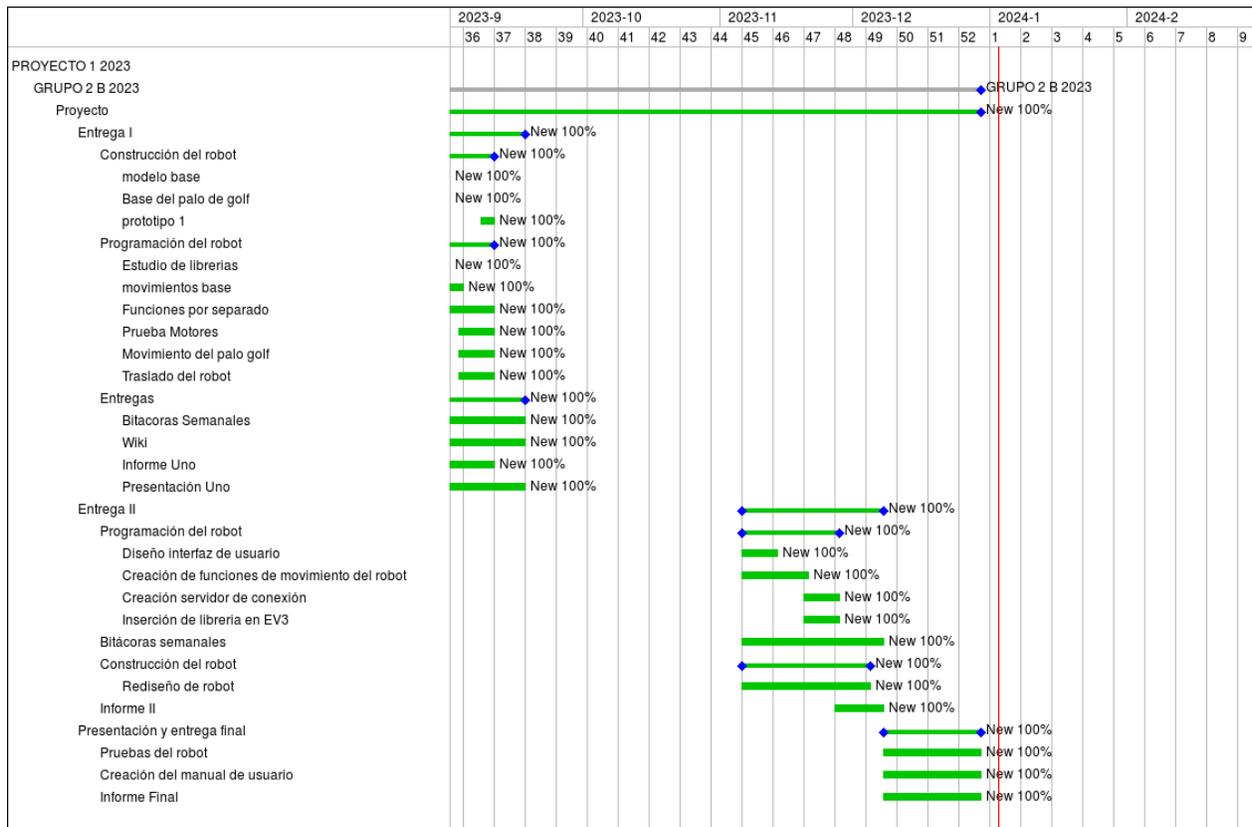
3. Planificación del proyecto

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

3.2 Carta 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

Dentro de un proyecto se debe 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 Hardware

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 golpear una pelota de tenis de mesa, simulando una pelota de golf. Este movimiento lo realizará 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.

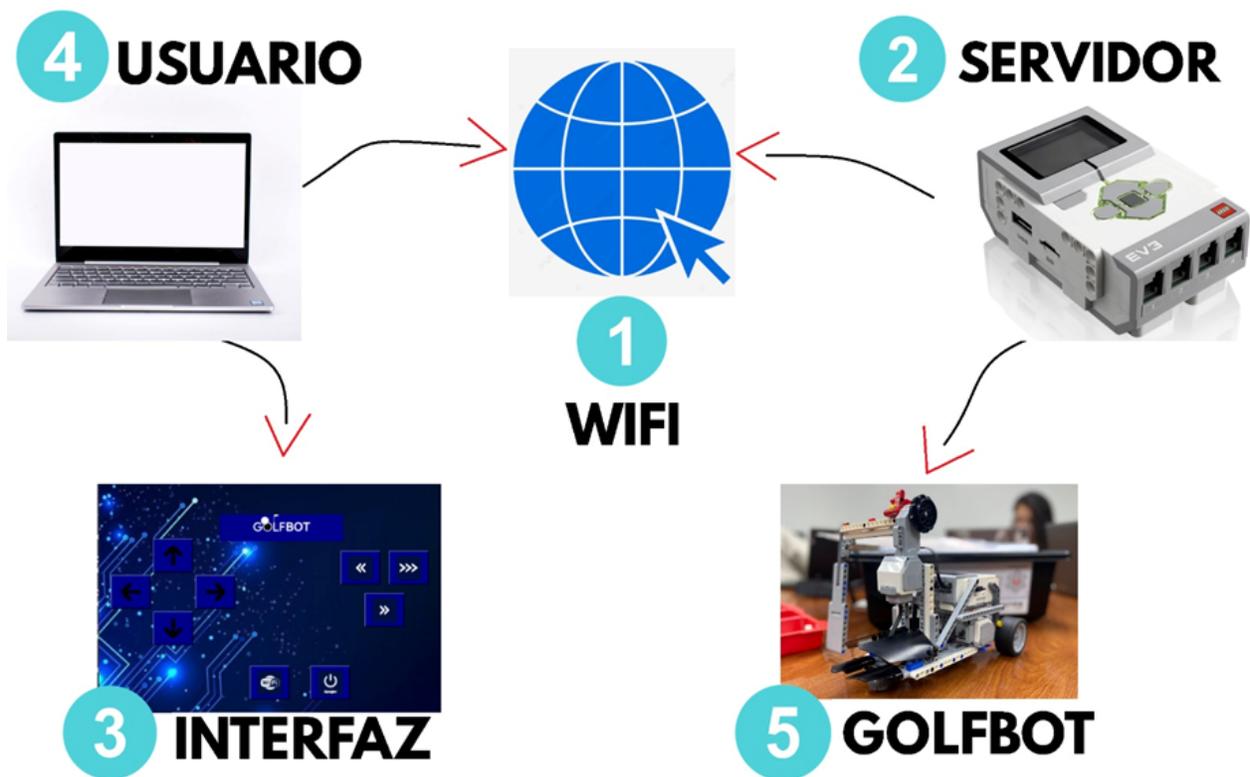
5.1.2 Requerimientos No Funcionales:

- 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 la interfaz gráfica deberá ser escrita con el lenguaje de programación Python, utilizando la librería Tkinter.

5.2 Arquitectura

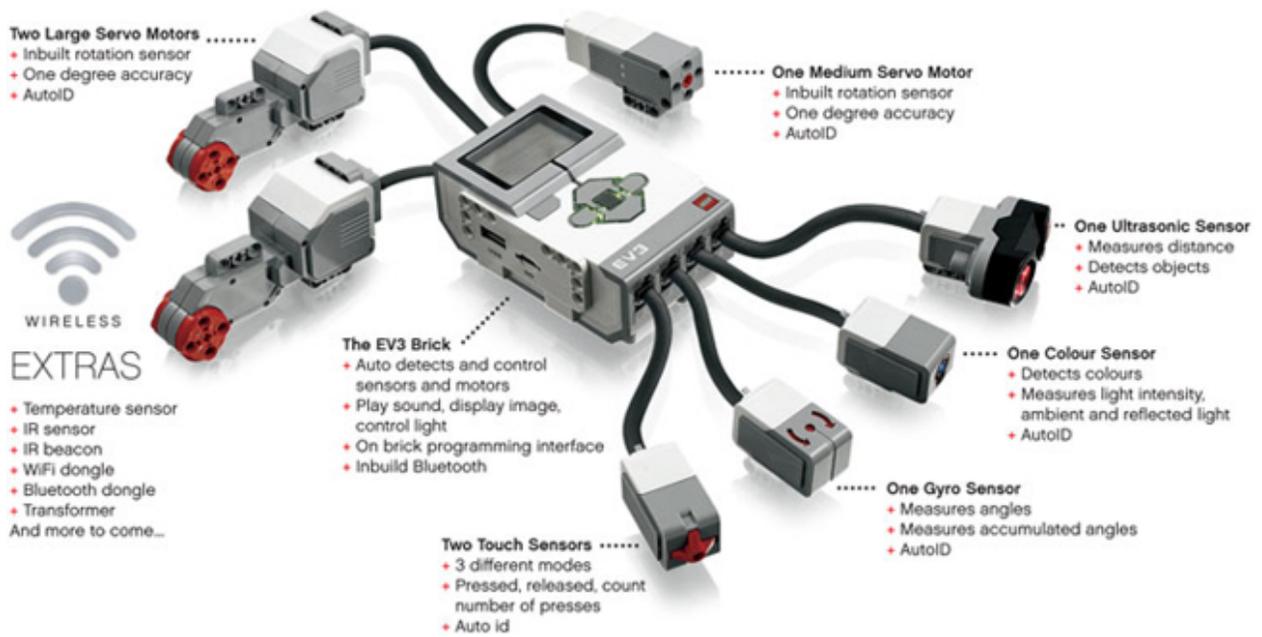
5.2.1 Conexión

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.



5.3 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).

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:

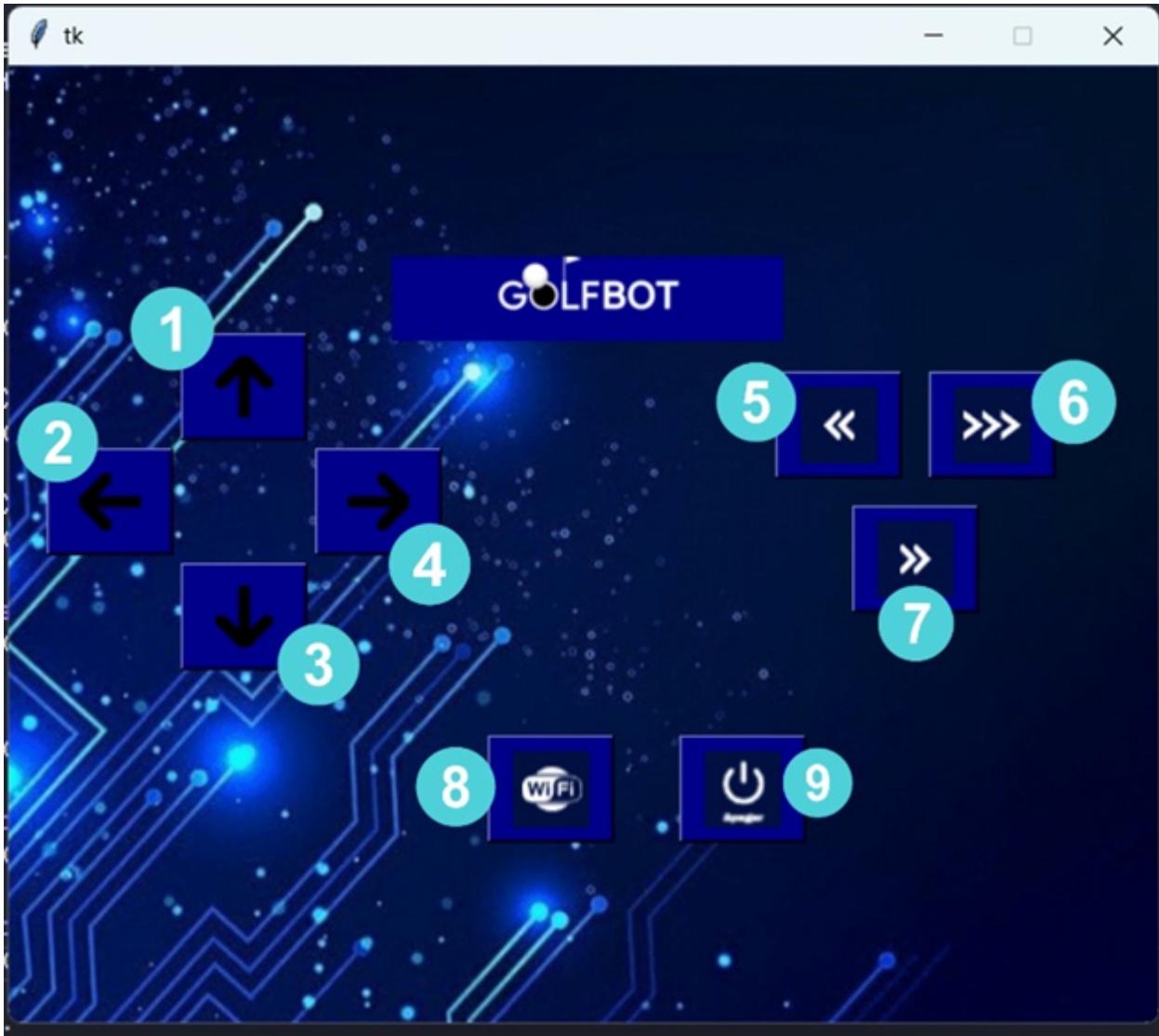
Energía Cinética en el Palo:

- 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.

5.4 Interfaz de usuario



Esta es la interfaz de usuario que permitirá al “GolfBot” 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 9 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.

A la derecha de la interfaz se encuentran 3 botones, los cuales controlan el movimiento del brazo del robot. Siendo uno de ellos para que el brazo se ‘devuelva’, este movimiento se utiliza principalmente para que el brazo agarre altura y transfiera una mayor energía a la hora de golpear la pelota. Los otros 2 botones tienen una funcionalidad similar, permiten que el brazo golpee la pelota, siendo la única diferencia entre ellos la velocidad en que se realiza la acción.

Finalmente en la parte inferior de la interfaz se encuentran 2 botones, uno que permite la conexión con el robot, y el otro para cerrar la interfaz.

En resumen:

Los botones (1,2,3,4) su función es desplazar al robot hacia “adelante”, “atrás” a la vez que rotar de “izquierda” como hacia la “derecha”

- El botón 5 mueve el palo hacia atrás
- El botón 7 mueve el palo hacia adelante
- El botón 6 mueve el palo hacia adelante con mucha más fuerza
- El botón 8 hace la conexión con el servidor
- El botón 9 cierra la interfaz Tkinter

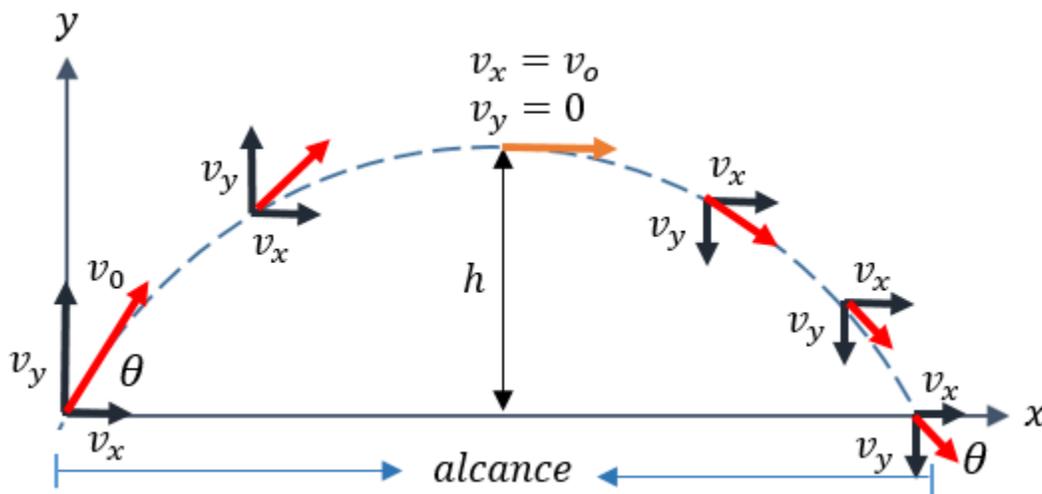
6. Implementación

6.1 Fundamento de proyectiles

Actualmente “GolfBot” 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:

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

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

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.

```
import socket
from library import *
s = socket.socket()
print("Socket creado")
port = 19999
s.bind( "", port )
print("El socket se creo con puerto:{}".format(port))
s.listen(5)
print("EL socket is listening...")
connect, addr = s.accept()
print("Se conecto a {}".format(addr))
while True:
    rawByte = connect.recv(1)
    char = rawByte.decode('utf-8')
    if (char == 'w'):
        avanzar()
    if (char == 's'):
        retroceder()
    if (char == 'd'):
        rotarDerecha()
    if (char == 'a'):
        rotarIzquierda()
    if (char == 'p'):
        golpeSuave()
    if (char == 'f'):
        golpeFuerte()
    if (char == 'i'):
        devolverBrazo()
    if (char == ' '):
        detener()
        detenerBrazo()
```

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.

```
from pybricks.hubs import EV3Brick
from pybricks.ev3devices import Motor
from pybricks.parameters import Port
from pybricks.robotics import DriveBase

RobotGolf = EV3Brick
motor_izquierda = Motor(Port.A)
motor_derecha = Motor(Port.B)
motor_brazo = Motor(Port.D)
base_robot = DriveBase(motor_izquierda, motor_derecha, wheel_diameter = 55.5, axle_track = 100)

# Funciones
def avanzar():
    print("Avanzando...")
    base_robot.straight(-10)

def retroceder():
    print("Retrocediendo...")
    base_robot.straight(10)

def rotarDerecha():
    print("Rotando derecha...")
    base_robot.turn(-10)

def rotarIzquierda():
    print("Rotando izquierda...")
    base_robot.turn(10)

def golpeSuave():
    print("Golpe suave...")
    motor_brazo.run(100)

def golpeFuerte():
    print("Golpe fuerte...")
    motor_brazo.run(100000)

def devolverBrazo():
    print("Retrocediendo brazo...")
    motor_brazo.run(-100)

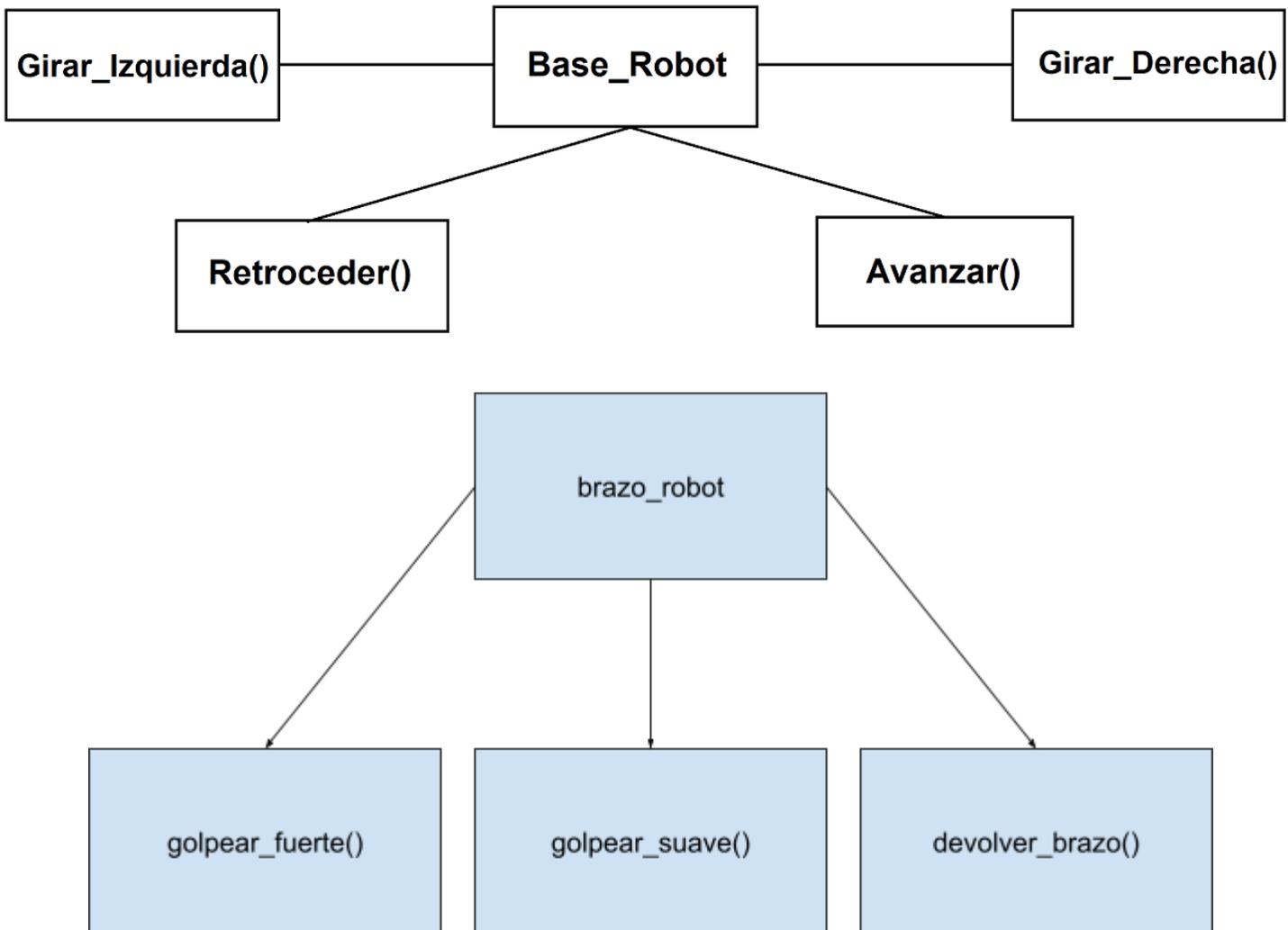
def detener():
    base_robot.stop()
```

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.

6.3 Diagramas



Resultados

7.1 Estado actual del proyecto

- El robot ev3 “GolfBot“ 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 terminada.
- Funcionamiento de la interfaz y diseño de la misma finalizado.
- Conexión entre el servidor y la interfaz completada.

Conclusión

Podemos afirmar que este proyecto nos ha resultado de mucho apoyo para desarrollar tanto habilidades individuales como grupales, el cambio de roles jugó un papel importante al forzarnos a aprender a ser funcionales en diversos roles. Hemos tenido un desarrollo interpersonal al aprender a organizarnos y su vez la manera en que dimos solución a los obstáculos que se nos presentaban constantemente.

A pesar de los desafíos que enfrentamos, tanto a nivel personal como en grupo, hemos logrado concluir este proyecto en su totalidad.