

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA

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

Ingenierí@
Computación e Informática

**Parte III
“Alligator 3000”**

Alumno(os): Brian Lopez
Karen Mamani
Angie Martinez
Polette Montt
Bastian Sucso

Asignatura: Proyecto I

Profesor: Humberto Urrutia López

Diciembre – 2023

Historial de Cambios

Fecha	Versión	Descripción	Autor(es)
13/11/2023	1.0	Formulación del Proyecto	Brian Lopez
02/11/2023	1.1	Desarrollo del plan de avance	Polette Montt Angie Martinez
10/11/2023	1.2	Corrección de la segunda parte.	Karen Mamani Angie Martinez
05/12/2023	1.3	Corrección de la segunda parte.	Angie Martinez Polette Montt

Tabla 1: Historial de Cambios

Tabla de Contenidos

1.	Índice de tablas	5
2.	Índice de figuras	5
3.	Panorama General	6
3.1.	Introducción	6
3.2.	Objetivos	6
3.2.1.	Objetivo General	6
3.2.2.	Objetivos Específicos	6
3.3.	Restricciones	7
3.4.	Entregables	7
4.	Organización del Personal	8
4.1.	Descripción de los Roles	8
4.2.	Personal que cumplirá los Roles	8
4.3.	Mecanismos de Comunicación	8
5.	Planificación del Proyecto	9
5.1.	Actividades	9
5.2.	Asignación de Tiempo	11
5.3.	Gestión de Riesgos	11
6.	Planificación de los Recursos	12
6.1.	Hardware	12
6.2.	Software	12
6.3.	Estimación de Costos	12
6.3.1.	Costos de Hardware y software	12
6.3.2.	Costos de Gestión	13
6.3.3.	Costos totales del proyecto	13
7.	Análisis y Diseño	14
7.1.	Especificación de Requerimiento	14
7.2.	Arquitectura	15
7.3.	Interfaz	16
8.	Implementación	18
8.1.	Fundamentos del movimiento parabólico y Velocidad Tangencial.	18
8.2.	Descripción de los programas	22

Proyecto I Plan de Proyecto

8.3. Diagramas	29
9. Resultados	30
9.1. Estado Actual del Proyecto	30
9.2. Problemas Encontrados y Solución Propuesta	30
10. Conclusión	31
11. Referencias	32

1. Índice de tablas

Tabla 1: Historial de Cambios	2
Tabla 2: Roles.	8
Tabla 3: Actividades.	9
Tabla 4: Gestión de riesgos.	11
Tabla 5: Costos de hardware y software.	13-14
Tabla 6: Costos de Gestión.	14
Tabla 7: Costos totales.	14
Tabla 8: Problemas y soluciones	30

2. Índice de figuras

Ilustración 1: Carta Gantt.	11
Ilustración 2: Arquitectura.	16
Ilustración 3: Interfaz.	16
Ilustración 4: Interfaz de conexión	17
Ilustración 5: Movimiento parabólico.	18
Ilustración 6: Velocidad Tangencial.	18
Ilustración 6: Diagrama I.	29
Ilustración 7: Diagrama II.	29

3. Panorama General

3.1. Introducción

El kit LEGO Mindstorms EV3 representa una fusión tecnológica y creatividad sin igual. Concebido como una pieza maestra en el mundo de la robótica educativa, este conjunto armoniza la fusión entre la venerable destreza de la marca LEGO y las capacidades vanguardistas de la ingeniería robótica. Dotado de una amalgama de elementos mecánicos, electrónicos y de programación, el EV3 se erige como un compendio lúdico y didáctico que despierta la curiosidad, el ingenio y el aprendizaje práctico en jóvenes y adultos por igual. Con su diversidad de sensores, actuadores y una interfaz intuitiva, este encomiable kit invita a explorar los reinos de la automatización, la resolución de problemas y el desarrollo de habilidades STEM (Ciencia, Tecnología, Ingeniería y Matemáticas), desencadenando así el potencial para crear, innovar y transformar ideas en realidades palpables.

En el contexto académico, ha tomado forma el ambicioso proyecto 'Alligator 3000', concebido como una parte del programa de estudios en la asignatura de Proyectos I. Este trabajo ha cobrado vida gracias a la utilización de componentes extraídos del kit educativo anteriormente mencionado. Dentro de las páginas de este compendio, se desplegará detalladamente el trabajo realizado durante las sesiones formativas. Resulta imperativo resaltar que este singular trabajo amalgama los conocimientos adquiridos por los estudiantes en el transcurso de su formación académica, combinando los preceptos inculcados en los cursos de Mecánica clásica, Taller de Programación I y II.

3.2. Objetivos

3.2.1. Objetivo General

- La misión primordial de esta actividad consiste en la concepción y manufactura de un robot EV3, dotado con la capacidad de golpear una pelota mediante una interfaz gráfica meticulosamente confeccionada a través del lenguaje de programación Python.

3.2.2. Objetivos Específicos

- Aprender las distintas herramientas y funcionalidades que trae la serie LEGO EV3 para armar un robot utilizando sus piezas.
- Incursionar en el entendimiento de las librerías de python para el desarrollo del software para el proyecto.
- Desarrollar e implementar un software que nos permita maniobrar los movimientos del robot.

3.3. Restricciones

- Para realizar la conexión del robot con el ordenador es necesaria una red wifi estable.

Proyecto I Plan de Proyecto

- El producto del robot que entreguemos no será más que un prototipo, por lo cual, no estará más que destinado a cumplir los requerimientos del proyecto.
- El robot debe moverse y golpear una pelota. Debemos involucrar y describir fórmulas físicas en su accionar.

3.4. Entregables

- Manual de usuario: Instructivo en el cual se da a conocer el funcionamiento del robot.
- Documentación subida a Redmine:
 - Wiki
 - Bitácoras
 - Carta Gantt
 - Informes
 - Presentaciones
- Robot "Alligator 3000": El robot armado y funcional.

4. Organización del Personal

"Para optimizar la eficiencia en nuestro equipo, cada miembro del grupo tiene una tarea asignada y se espera que la complete a tiempo. Aunque hay responsables designados para cada área, es fundamental la colaboración de todos para garantizar el éxito tanto a nivel individual como grupal

4.1.Descripción de los Roles

- **Jefe de grupo:** responsable de representar al grupo, la organización y liderar el proceso de toma de decisiones.
- **Programador:** encargado de crear e implementar el código en Python para habilitar las funciones requeridas del robot.
- **Ensamblador:** responsable de concebir y ensamblar el robot de manera que pueda movilizarse en todas las direcciones y golpear una pelota..
- **Diseñador:** a cargo de la estética y diseño visual de la interfaz gráfica.
- **Documentador:** responsable de elaborar informes, presentaciones, bitácoras, manuales de usuario y documentación del proyecto.

4.2.Personal que cumplirá los Roles

Rol	Responsable	Involucrados
Jefe de grupo	Brian Lopez	-
Programador	Polette Montt	Brian Lopez, Bastian Sucso
Ensamblador	Karen Mamani	Angie Martinez
Diseñador	Bastian Sucso	Polette Montt, Brian Lopez
Documentador	Angie Martinez	Bastian Sucso, Polette Montt,

Tabla 2: Roles.

4.3.Mecanismos de Comunicación.

Comunicación entre los integrantes del proyecto: En cuanto a la comunicación interna del equipo, se utilizarán plataformas como Whatsapp y Discord. Además, se planificaron reuniones presenciales para abordar los ajustes relacionados con el robot y el proyecto en su conjunto.

5. Planificación del Proyecto

5.1. Actividades

Actividad	Descripción	Responsable
Redacción de bitácoras	Registro de las actividades que se desarrollan semanalmente.	Polette Montt Bastian Sucso
Redacción de Carta Gantt	Planificación de las actividades y tiempo estimado en la realización de las mismas.	Bastian Sucso Angie Martinez
Redmine	Manejo y uso de las distintas funcionalidades que posee el Redmine	Angie Martinez Bastian Sucso
Construcción del Robot	Análisis, diseño y construcción del robot.	Karen Mamani
Wiki	Se comparte información del proyecto.	Angie Martinez
Estimación de costos	Calcular el presupuesto del proyecto.	Angie Martinez
Informe I	Redactar Informe de Formulación de proyecto	Polette Montt, Bastian Sucso, Angie Martinez, Brian Lopez, Karen Mamani
Presentación I	Diseñar la presentación del informe I	Angie Martinez
Fotografía	Registro de imágenes del avance/modificaciones del robot.	Angie Martinez
Instalar SO en la tarjeta SD	Instalación del SO en la tarjeta SD que fue insertada en el robot	Brian Lopez Bastian Sucso

Proyecto I Plan de Proyecto

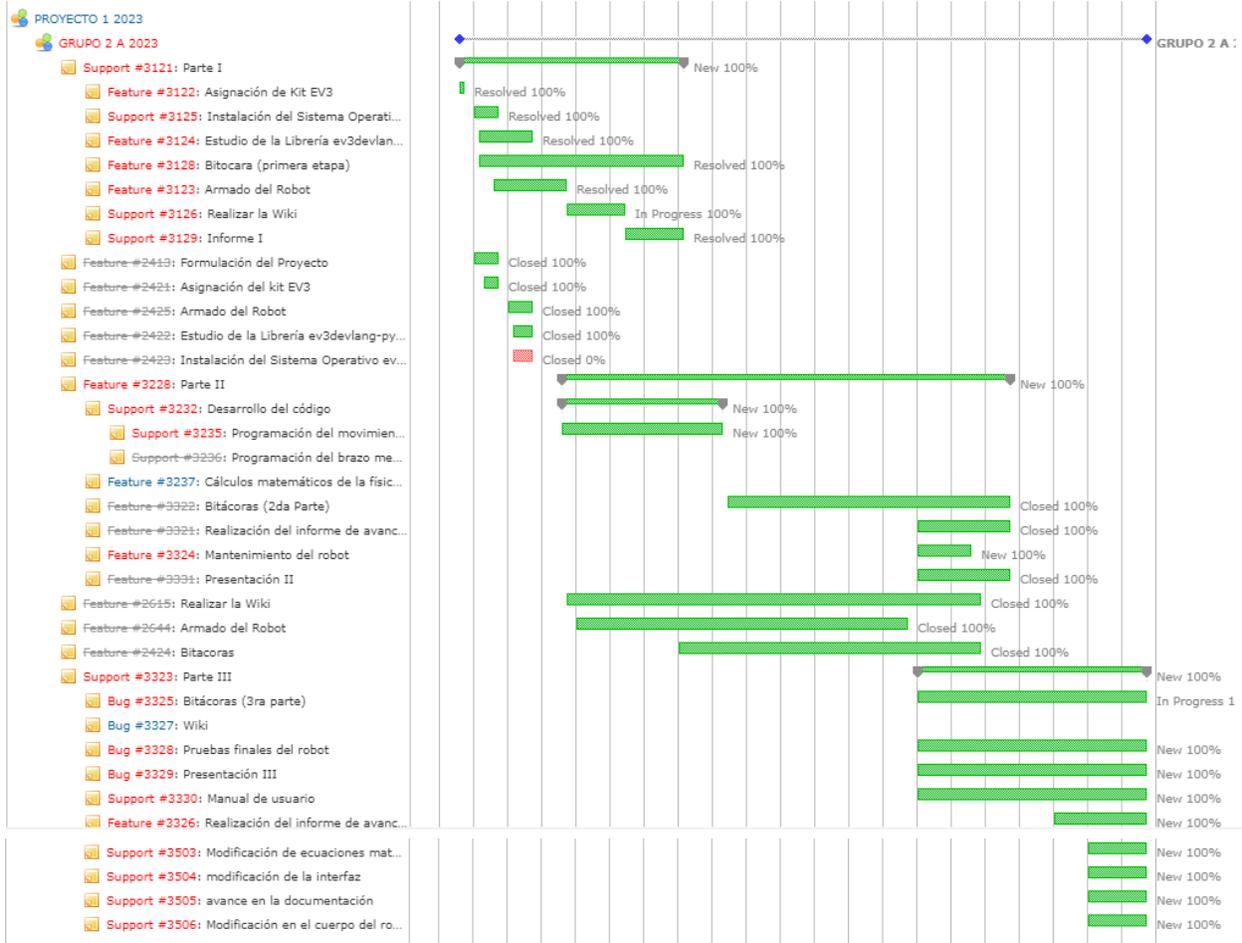
Estudio y Programación ev3dev	Estudio del lenguaje de programación Python y de la librería ev3dev-lang-python.	Brian Lopez Bastian Sucso Polette Montt
Programación de los movimientos	Crear las funciones en el código para el movimiento del robot	Brian Lopez, Bastian Sucso, Polette Montt
Creación de Interfaz Gráfica	Programación de la interfaz gráfica usando Tkinter	Brian Lopez
Implementación interfaz gráfica en el servidor	Manejo remoto del robot desde la interfaz	Brian Lopez, Bastian Sucso, Polette Montt
Informe II	Redacción del informe del avance del proyecto.	Karen Mamani, Bastian Sucso, Polette Montt, Angie Martinez, Brian Lopez
Presentación II	Creación y diseño de la presentación del informe II	Angie Martinez
Modificación del cuerpo del robot.	Cambiar las piezas para mejorar el diseño.	Brian Lopez
Modificación de la interfaz gráfica.	Mejorar el diseño tomando en cuenta las críticas constructivas recibidas.	Polette Montt
Informe III	Redacción del informe final del proyecto.	Bastian Sucso Angie Martinez
Presentación III	Creación y diseño de la presentación del informe III	Angie Martinez
Revisión de fórmulas físicas.	Verificar si las fórmulas son correctas y van de la mano con el mecanismo del robot.	Brian Lopez Karen Mamani
Pruebas del robot	Verificar el movimiento y funciones del robot.	Brian Lopez Polette Montt

Proyecto I Plan de Proyecto

Tabla 3: Actividades.

5.2. Asignación de Tiempo

La Carta Gantt se ha concebido con el objetivo de optimizar la planificación y gestión de nuestras actividades a lo largo del semestre. Además, brinda la posibilidad de registrar el tiempo dedicado a cada tarea y compararlo con el tiempo estimado para evaluar nuestra eficiencia en la ejecución de las actividades planificadas.



5.3. Gestión de Riesgos

Niveles de impacto:

1. Catastrófico
2. Crítico
3. Circunstancial
4. Irrelevante

Riesgos	Probabilidad de concurrencia	Nivel de impacto	Acción remedial
Escasez de piezas.	20%	3	Reemplazar las piezas faltantes.
Extravío de la tarjeta de memoria.	15%	1	Respaldar la información y agregarla a una nueva.
Reconstrucción total del robot	40%	2	Contar con fotos del robot para volver a construirlo.
Falta de cumplimiento de tareas.	20%	2	Reorganizar los roles asignados, e incentivar al grupo a entregar a tiempo sus partes.
Enfermedad, falta de un integrante.	40%	3	Repartir las tareas del integrante entre el grupo.
Descarga de batería del EV3.	45%	4	Cargar la batería, mientras el EV3 no sea utilizado.

Tabla 4: Gestión de riesgos.

6. Planificación de los Recursos

6.1. Hardware

- Notebook Msi modern b5
- Kit Lego MINDSTORMS (EV3)
- Lego Mindstorms Ev3 L-servo Motor
- Micro SD (8 GB)
- Dongle USB Wifi (Tp-link Ac60)
- Router
- Toshiba tecra z40-c
- Asus VivoBook x409FB
- Tablet Samsung Galaxy Tab S4

6.2. Software

- Visual Studio para el desarrollo del software
- Balena Edge es una plataforma que transfiere el archivo a la SD
- EV3DEV Python Simulator para ir realizando implementaciones
- Redmine donde se subira toda la documentación de los avances
- Canva

6.3. Estimación de Costos

6.3.1. Costos de Hardware y Software

Nombre del producto	Precio	Cantidad	Total
EV3 LEGO MINDSTORMS	\$1,299,000	1	\$1,299,000
Lego Mindstorms Ev3 L-servo Motor	\$112,500	3	\$337,500
Toshiba tecra z40-c	\$349,000	2	\$698,000
Tarjeta MicroSD 8GB	\$3,990	1	\$3,990
Tp-link Ac600	\$13,495	1	\$13,495

Proyecto I Plan de Proyecto

Asus VivoBook X409FB	\$491,720	1	\$491,720
Tablet Samsung Galaxy Tab S4	\$499,990	1	\$499,990
Canva (Versión Gratuita)	Gratuito	1	\$0
Visual code Studio	Gratuito	1	\$0
Librería EV3	Gratuito	1	\$0
Total			\$3,343,695

Tabla 5: Costos de Hardware.

6.3.2 Costos de gestión

Integrante	Valor por hora	Horas Trabajadas	Valor total
Angie Martinez	\$10,000	53	\$530,000
Bastian Sucso	\$11,000	52,75	\$580,250
Brian Lopez	\$12,000	60,3	\$723,600
Polette Montt	\$10,000	50,75	\$507,500
Karel Mamani	\$10,000	50,75	\$507,500

Tabla 6: Costos de gestión.

6.3.3 Costos totales del proyecto

Nombre	Costo total
Costos de Software y Hardware	\$3.343.695
Costos de Gestión	\$2,848,850
Costo total proyecto	\$6,192,545

Tabla 7: Costos totales.

En resumen, la estimación del proyecto durante 4 meses tendrá un presupuesto de \$6,192,545.

7. Análisis y Diseño

7.1. Especificación de Requerimiento

Requerimientos funcionales:

- El robot debe tener la capacidad de moverse.
- Los usuarios controlan el robot a través de una interfaz gráfica en un dispositivo técnico (por ejemplo, una computadora portátil).
- El robot golpea una pelota.
- Se espera que el robot exhiba la habilidad de impactar una pelota, haciendo uso de las fórmulas físicas asociadas al movimiento parabólico.

Requerimientos no funcionales:

- Los movimientos del robot se logran mediante la interfaz gráfica, que alberga los comandos necesarios para propiciar el avance, retroceso, giro a la izquierda y derecha, así como para ejecutar con precisión el impacto a una pelota.
- Los movimientos y la interfaz gráfica es programada en Python.
- La simulación del Swing de golf se realiza mediante el uso de las piezas del LEGO EV3

7.2. Arquitectura

En la concepción del robot destinado al golf, se ha llevado a cabo una adecuada investigación para idear la manera más óptima de instalar un brazo mecánico encargado de ejecutar el movimiento de impacto. Este proceso ha implicado la meticulosa selección de las piezas idóneas con el fin de construir un brazo articulado capaz de emular el gesto de un swing de golf. Asimismo, se ha procedido al cálculo y uso de fórmulas físicas asociadas al movimiento parabólico.

En lo concerniente a su programación, se ha empleado el software provisto por LEGO para el EV3. Dentro de este entorno, se ha elaborado un programa que ejerce control sobre los motores y sensores del autómatas. Este programa incorpora algoritmos diseñados con precisión, los cuales facultarán al robot para desplazarse hacia la ubicación idónea, ajustar su posición para un golpe óptimo y, finalmente, ejecutar el movimiento de impacto de manera plenamente controlada.

7.2.1 Conexión del robot

Proyecto I Plan de Proyecto

1. El Servidor y el notebook deben estar conectados a la misma red Wifi.
2. Para iniciar la conexión remota con el notebook, se debe ejecutar el archivo Server.py.
3. Una vez iniciada, la interfaz gráfica puede ser activada por el usuario para controlar el robot a distancia.
4. La interfaz se conecta al servidor del robot, permitiendo al usuario enviar comandos para controlar los movimientos del robot.
5. Los movimientos del robot son ejecutados en respuesta a las instrucciones enviadas por el usuario a través del servidor establecido entre el robot y el notebook.

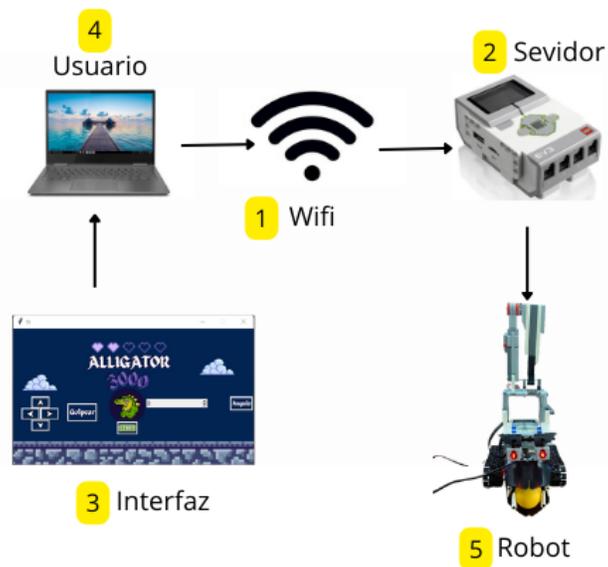


Ilustración 2: Arquitectura

7.3. Interfaz

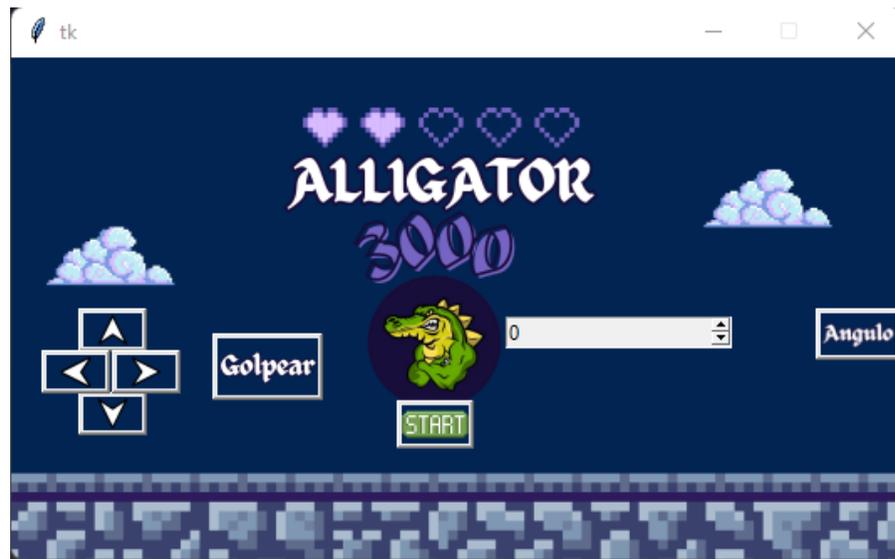


Ilustración 3: Interfaz.

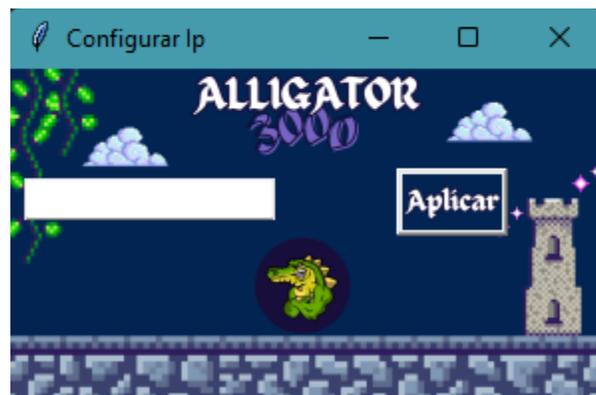


ilustración 4: interfaz de la conexión ip

El usuario contará con una interfaz, similar a la representada en la imagen (Ilustración 3), que servirá como pantalla principal. En esta interfaz, se encontrarán los elementos clave para la interacción con el robot, brindando una experiencia centralizada y accesible. A continuación se dará una breve descripción de los elementos que componen la interfaz:

- La interfaz proporcionará flechas direccionales para indicar las distintas orientaciones del robot: avanzar, retroceder, girar a la izquierda y girar a la derecha.
- Se incluirá un botón denominado "Golpear", el cual activará la función de golpear la pelota.

Proyecto I Plan de Proyecto

- Para establecer la conexión entre el cliente y el servidor en el EV3, se integrará un botón llamado "Start", el cual abre una pestaña que contiene opciones para poder permitir al usuario vincularse mediante la dirección IP del servidor.
- En la barra desplegable, se modifican los valores del ángulo que se utilizaran para posicionar el brazo al golpear.
- El botón "Ángulo" se ajusta con alguno de los 2 valores para seleccionar (45° o 90°) en la casilla de su lado izquierdo. Al presionar el botón el robot ajusta su brazo de acuerdo al ángulo seleccionado.

8. Implementación

8.1. Fundamentos del movimiento parabólico y de Velocidad Tangencial.

El Swing realiza un movimiento parabólico, en el cual habrá 3 posibles situaciones a suceder, de acuerdo a 3 tipos de ángulos de trayectoria del brazo mecánico (0° , 45° y 90°). A continuación, se utilizarán las fórmulas correspondientes para calcular las velocidades iniciales del proyectil de acuerdo a cada ángulo. Cabe recalcar, que no se considera el roce del viento y la aceleración de gravedad tiene un valor aproximado de 9,8 m/s.

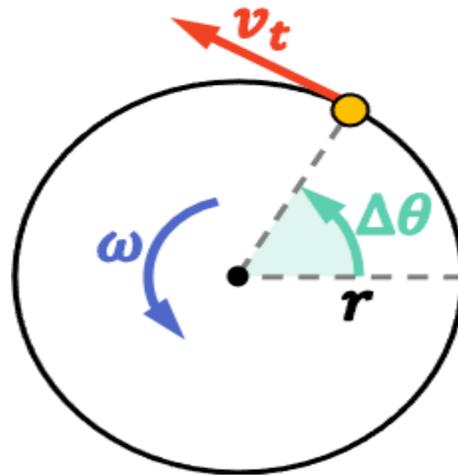


Ilustración 5: Velocidad Tangencial.

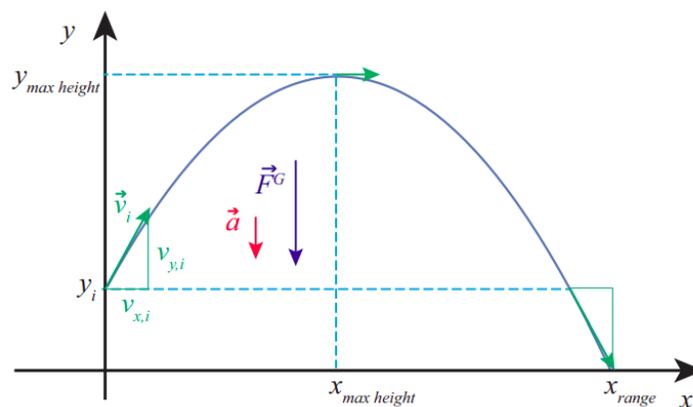


Ilustración 6: Movimiento parabólico.

Información sacada de la [Guía de Uso](#) :

- El motor del brazo mecánico gira con una velocidad angular de 160-170 RPM.

Para los cálculos realizados se utilizó el valor inferior del intervalo, es decir 160 RPM, este se consideró solamente para la evaluación de la velocidad tangencial en el lanzamiento con ángulo de inclinación en 0° . Esta decisión tiene su fundamento en que se carece de los conocimientos para desarrollar el cálculo de la velocidad tangencial cuando está implicada una velocidad influenciada por una aceleración (para los casos de lanzamientos con trayectoria de ángulo de 45° y 90°), ya que en cursos anteriores solo se alcanzó a desarrollar con velocidad constante. Aún así con ayuda de las fórmulas para calcular Componentes horizontales y verticales de posición se pudo obtener las velocidades iniciales para los ángulos de 45° y 90° .

Información recopilada de datos experimentales:

- ❖ Para el lanzamiento con ángulo de inclinación del brazo en 0° se recopiló:

- $X_0 = 0 \text{ m}$
- $Y_0 = 0,22 \text{ m}$
- $X_f = 0,56 \text{ m}$
- $Y_f = 0 \text{ m}$
- $t = 0,5 \text{ s}$

- ❖ Para el lanzamiento con ángulo de inclinación del brazo en 45° se recopiló:

- $X_0 = 0 \text{ m}$
- $Y_0 = 0,22 \text{ m}$
- $X_f = 0,71 \text{ m}$
- $Y_f = 0 \text{ m}$
- $t = 0,6 \text{ s}$

- ❖ Para el lanzamiento con ángulo de inclinación del brazo en 90° se recopiló:

- $X_0 = 0 \text{ m}$
- $Y_0 = 0,22 \text{ m}$
- $X_f = 0,76 \text{ m}$
- $Y_f = 0 \text{ m}$

$$\rightarrow t = 0,65 \text{ s}$$

1. Velocidad Tangencial para el ángulo de inclinación del brazo en 0° :

Fórmula:

$$\rightarrow v_t = r \cdot \omega$$

Donde:

- v_t (m/s) es la Velocidad Tangencial del extremo del brazo (que coincidirá con la velocidad inicial de la pelota al ser lanzada).
- r (m) es la longitud del brazo(radio desde el eje de rotación hasta el extremo del brazo).
- ω (rad/s) es la velocidad angular del motor.

Datos proporcionados :

- $r = 0,134 \text{ m}$
- $\omega = 160 \text{ RPM} \rightarrow 16,8 \text{ rad/s}$

Procedimiento de la fórmula:

$$v_t = 0,134 \cdot 16,8$$

$$v_t = 2,25 \text{ (m/s)}$$

2. Calcular el ángulo de inclinación con la que sale eyectada la pelota utilizando los datos obtenidos del brazo con inclinación en 0° :

Fórmulas a utilizar (Componentes horizontal):

$$\rightarrow X_f = X_0 + V_0 \cos \theta t$$

Datos proporcionados:

- $X_f = 0,56 \text{ m}$
- $X_0 = 0 \text{ m}$

Proyecto I Plan de Proyecto

- $V_0 = Vt = 2,25 \text{ m/s}$
- $t = 0,5 \text{ s}$

Procedimiento:

$$X_f = X_0 + V_0 \cos \theta t$$
$$0,56 = 0 + 2,25 \cdot \cos \theta \cdot 0,5$$
$$\cos \theta = \frac{0,56}{2,25 \cdot 0,5}$$

$$\cos \theta = \frac{0,56}{1,13}$$

$$\cos \theta = \frac{0,56}{1,13}$$

$$\theta = \arccos\left(\frac{0,56}{1,13}\right)$$
$$\theta = 60,26^\circ$$

3. Sacar la velocidad inicial del brazo con inclinación en 45° usando sus datos obtenidos en conjunto del ángulo de inclinación sacado anteriormente:

Fórmulas a utilizar (Componentes horizontal):

$$\rightarrow X_f = X_0 + V_0 \cos \theta t$$

Datos proporcionados:

- $X_f = 0,71 \text{ m}$
- $X_0 = 0 \text{ m}$
- $\theta = 60,26^\circ$
- $t = 0,6 \text{ s}$

Procedimiento:

$$X_f = X_0 + V_0 \cos \theta t$$
$$0,71 = 0 + V_0 \cdot \cos (60,26^\circ) \cdot 0,6$$
$$V_0 = \frac{0,71}{\cos (60,26^\circ) \cdot 0,6}$$

$$V_0 = 2,37 \text{ m/s}$$

4. Sacar la velocidad inicial del brazo con inclinación en 90° usando sus datos obtenidos en conjunto del ángulo de inclinación sacado anteriormente:

Fórmulas a utilizar (Componentes horizontal):

$$\rightarrow X_f = X_0 + V_0 \cos \theta t$$

Datos proporcionados:

- $X_f = 0,76$ m
- $X_0 = 0$ m
- $\theta = 60,26^\circ$
- $t = 0,65$ s

Procedimiento:

$$\begin{aligned} X_f &= X_0 + V_0 \cos \theta t \\ 0,76 &= 0 + V_0 \cdot \cos (60,26^\circ) \cdot 0,65 \\ V_0 &= \frac{0,76}{\cos (60,26^\circ) \cdot 0,65} \end{aligned}$$

$$V_0 = 2,39 \text{ m/s}$$

5. Resultados:

Finalmente obtuvimos que los valores para las velocidades iniciales son:

- Brazo con inclinación en $\sphericalangle 0^\circ = 2,25$ m/s.
- Brazo con inclinación en $\sphericalangle 45^\circ = 2,37$ m/s.
- Brazo con inclinación en $\sphericalangle 90^\circ = 2,39$ m/s.

8.2.Descripción de los programas

- **Funciones:**

```
#!/usr/bin/env python3
from ev3dev2.motor import LargeMotor, MediumMotor, OUTPUT_A, OUTPUT_B,
OUTPUT_C, OUTPUT_D, SpeedPercent, MoveTank
import time
from ev3dev2.sound import Sound

sound = Sound()

ruedas = MoveTank(OUTPUT_B,OUTPUT_C)
brazo= LargeMotor(OUTPUT_D)

def Adelante():
    ruedas.on(SpeedPercent(100),SpeedPercent(100))

def Atras():
    ruedas.on(SpeedPercent(-70),SpeedPercent(-70))

def Izquierda():
    ruedas.on(SpeedPercent(70),SpeedPercent(-70))

def Derecha():
    ruedas.on(SpeedPercent(-70),SpeedPercent(70))

def Brazo():
    brazo.on_for_rotations(100,-1)
def Inclinacion0():
    brazo.on_for_degrees(10, -0,brake=True)
def Inclinacion45():
    brazo.on_for_degrees(10, 45, brake=True)
def Inclinacion90():
    brazo.on_for_degrees(10, 90, brake=True)
def Parar():
    ruedas.stop()
```

- **Server:**

```
#!/usr/bin/env python3
import socket
from Function import *
s = socket.socket()
print("Socket creado")
port = 8080
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'):
        Adelante()
    if (char == 's'):
        Atras()
    if (char == 'd'):
        Derecha()
    if (char == 'a'):
        Izquierda()
    if (char == 'p'):
        Brazo()
    if (char == 'i'):
        Inclination0()
    if (char == 'o'):
        Inclination45()
    if (char == 'u'):
        Inclination90()
    if (char == ' '):
        Parar()
    if (char == 'q'):
        print("Terminada la sesion...")
        break
```

- **Cliente:**

```
#!/usr/bin/env python3
from tkinter import *
from tkinter import messagebox
from tkinter import ttk
import tkinter as tk
import socket
import time
import sys

#Funciones de los botones

def getAddress():
    w_ip = Toplevel()
    w_ip.geometry("300x169")
    imagen1 = PhotoImage(file = "imagen2.png")
    labelimagen1 = Label(w_ip, image = imagen1)
    labelimagen1.pack()

    dato = StringVar(w_ip)
    w_ip.title("Configurar Ip")
    ip = ttk.Entry(w_ip, textvariable=dato).place(x=8,y=55)
    boton_aplicar = tk.PhotoImage(file="aplicar.png")
    boton_aplicar = Button(w_ip, image = boton_aplicar, command=lambda: [conectar(dato.get(), port), w_ip.destroy()]).place(x=195,y=50)
    w_ip.mainloop()

def conectar(address, port):
    try:
        clientSocket.connect((address, port))
        messagebox.showinfo("Mensaje Servido", "Cliente conectado al robot: {0} : {1}".format(ipAddress, port))
    except (socket.error, socket.timeout) as e:
        messagebox.showerror("Error", f"No se pudo conectar al robot en la dirección {address} : {port}")

def forward():
    clientSocket.send(bytes([ord('w')]))
```

```
def backward():
    clientSocket.send(bytes([ord('s')]))

def right():
    clientSocket.send(bytes([ord('d')]))

def left():
    clientSocket.send(bytes([ord('a')]))

def hit():
    clientSocket.send(bytes([ord('p')]))

def end_session():
    clientSocket.send(bytes([ord('q')]))

def soltar(event):
    clientSocket.send(bytes([ord(' ')]))

def Angulo():
    if(datos.get()=="0"):
        clientSocket.send(bytes([ord('i')]))
    elif(datos.get() == "45"):
        clientSocket.send(bytes([ord('o')]))
    elif(datos.get() == "90"):
        clientSocket.send(bytes([ord('u')]))

window = Tk()
window.geometry("533x300")
imagen = PhotoImage(file = "imagen1.png")
labelimagen = Label(window, image = imagen)
labelimagen.pack()
window.resizable(False, False)

#Botones en la ventana

boton_arri = tk.PhotoImage(file="arriba.png")
button_1 = Button(image= boton_arri,repeatdelay= 50, repeatinterval=
50,command=forward)
button_1.place(x=40,y=150)
```

```
boton_abajo = tk.PhotoImage(file="abajo.png")
button_2 = Button(image= boton_abajo,repeatdelay= 50, repeatinterval= 50,
command=backward)
button_2.place(x=40,y=200)

boton_derech = tk.PhotoImage(file="derecha.png")
button_3 = Button(image= boton_derech,repeatdelay= 50, repeatinterval= 50,
command=right)
button_3.place(x=60,y=175)

boton_izq = tk.PhotoImage(file="izquierda.png")
button_4 = Button(image= boton_izq,repeatdelay= 50, repeatinterval= 50,
command=left)
button_4.place(x=18,y=175)

boton_golpear = tk.PhotoImage(file="golpear.png")
button_5 = Button(window, image = boton_golpear,
command=hit).place(x=120,y=165)

datos = StringVar(window)
angulo = Spinbox(window, state ="readonly", values = ("0", "45", "90"),
textvariable=datos).place(x=295,y=155)
boton_angulo = tk.PhotoImage(file="angulo.png")
button_6 = Button(window, text = "ingresar", command=Angulo, image =
boton_angulo).place(x=480,y=150)

button_1.bind('<ButtonRelease-1>',soltar)
button_2.bind('<ButtonRelease-1>',soltar)
button_3.bind('<ButtonRelease-1>',soltar)
button_4.bind('<ButtonRelease-1>',soltar)

boton_conec = tk.PhotoImage(file="conectar.png")
button_connect = Button(window, image = boton_conec, command=lambda:
getAddress()).place(x=230,y=205)
ipAddress = "192.168.84.197"

# Read the command line argument for the IP address of the server
if len(sys.argv) > 2:
```

```
    print("Usage: client-laptop.py [IP-addr-of-robot]")
    sys.exit(1)
elif len(sys.argv) == 2:
    ipAddress = sys.argv[1]
    print("Using specified IP address: {}".format(ipAddress))
else:
    print("Using default IP address: {}".format(ipAddress))

clientSocket = socket.socket()
port = 8080
window.mainloop()
```

8.3. Diagramas:

Mediante los siguientes diagramas se muestran las funciones de movimiento del robot.

B, C y D representan los motores OUTPUT B, OUTPUT C, OUTPUT D respectivamente.

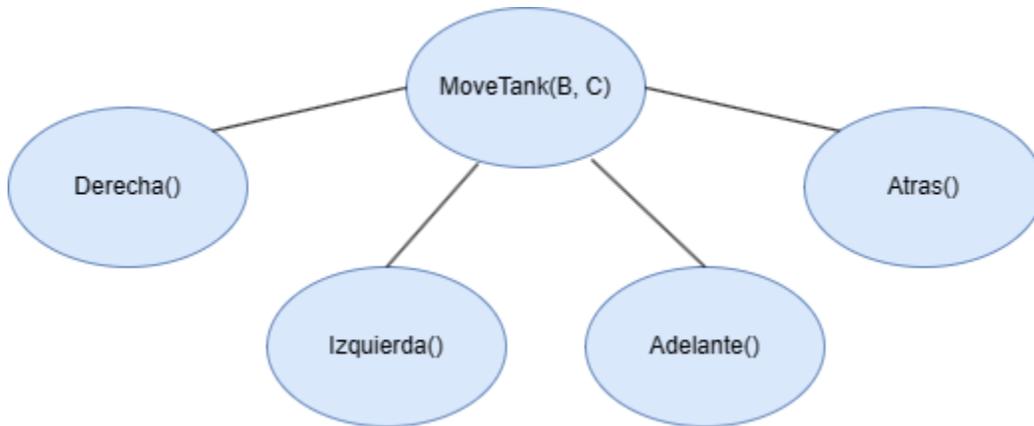


Ilustración 7: Diagrama I

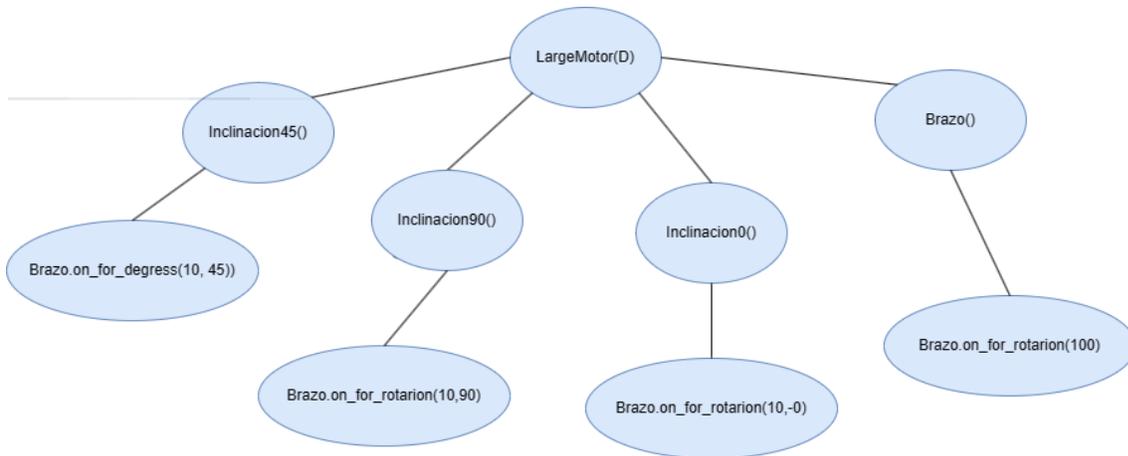


Ilustración 8: Diagrama II

9. Resultados

9.1. Estado Actual del Proyecto

- La versión finalizada del robot “Alligator 3000”
- Las funciones de movimiento. Esto significa que el robot ya se desplaza en distintas direcciones y puede movilizar el brazo, diseñado para golpear la pelota.
- Interfaz gráfica, se realizó mediante el uso de la librería “Tkinter”. La interfaz sirve para facilitar la experiencia del usuario.
- La instauración del servidor involucró el empleo de la biblioteca de Python llamada "Socket", la cual facilita la conexión entre el host y el cliente. Este servidor está implementado en el entorno EV3DEV bajo el nombre de Server.py. Actualmente, se ha concluido su desarrollo y se encuentra plenamente operativo.
- La conexión vía remota.
- La wiki del proyecto.
- Carta Gantt actualizada.
- Bitácoras, informes y presentaciones concretadas.

9.2. Problemas Encontrados y Solución Propuesta

Problemas encontrados	Soluciones
El robot, se encontraba descargado muchas veces.	Cargarlo al principio de la sesión de clases
La inestabilidad del robot.	Se modificaron las fuerzas del motor y se agregaron piezas para reforzar.
Incongruencias en los tiempos de la carta gantt, con la realidad del proyecto.	Se reajustó el tiempo teniendo en cuenta las horas trabajadas en clases, autónomas y los días feriados.
El uso de una librería errónea (Pybrics) en la implementación de los movimientos del robot “Pybrics”	Se cambió de librería, a la librería: Ev3dev2
Falta de conocimientos en el manejo de Visual Studio Code y conocimiento en redes y conexiones.	Preguntar a personas que manejan más el tema y aprender de forma autónoma en base a prueba y error.

Tabla 8: Problemas encontrados y solución propuesta.

10. Conclusión

El proyecto desarrollado a lo largo del semestre en el marco del curso de Proyecto I, ha representado un desafío significativo para el equipo de estudio. La tarea de armar un Robot EV3 para ejecutar el movimiento del swing de golf implicó una serie de obstáculos y desafíos a lo largo del proceso.

A pesar de la planificación establecida en la carta Gantt, el equipo enfrentó contratiempos en relación con el cumplimiento de los plazos establecidos. Asimismo, se encontraron problemas relacionados con la estructura misma del robot, lo que demandó ajustes y refinamientos constantes en la configuración para lograr los resultados esperados.

Sin embargo, entre las múltiples adversidades encontradas, la más ardua fue la dificultad experimentada al intentar establecer la conexión del robot con el servidor y ejecutar la interfaz gráfica para controlar el movimiento del mismo desde un ordenador. Esta fase se convirtió en el punto de inflexión, presentando un serio desafío. Los esfuerzos del equipo se vieron obstaculizados por fallos recurrentes en la programación y falta de conocimiento en la forma de conexión, lo que generó una situación de incertidumbre considerable.

A pesar de estos contratiempos, el equipo no cejó en su empeño y se mantuvo persistente en su búsqueda de soluciones viables. Finalmente, tras investigaciones y pruebas continuas, se logró superar las barreras técnicas y se consiguió establecer la conexión exitosa del robot, entre el servidor y el cliente, permitiendo el control remoto desde el ordenador.

Esta experiencia no solo representó un reto técnico, sino también un valioso aprendizaje para el equipo. La resolución de problemas complejos, la adaptabilidad frente a imprevistos y la perseverancia fueron habilidades fundamentales cultivadas durante este proyecto. A pesar de las dificultades, el logro de establecer la conexión deseada enfatiza la importancia de la colaboración, la dedicación y el ingenio para superar los obstáculos en la consecución de metas ambiciosas.

En resumen, el proyecto ha sido una experiencia enriquecedora que ha permitido al equipo no solo expandir sus conocimientos técnicos, sino también fortalecer habilidades clave para afrontar desafíos futuros en el ámbito de la ingeniería y la tecnología.

11. Referencias

[1] Salario para Ingeniero Informática en Chile - Salario Medio. (s/f). Talent.com. Recuperado el 14 de septiembre de 2023, de <https://cl.talent.com/salary?job=ingeniero+informatica>

[2] Getting started with LEGO® MINDSTORMS Education EV3 MicroPython — ev3-micro python 2.0.0 documentation. (s/f). Pybricks.com. Recuperado el 14 de septiembre de 2023, de <https://pybricks.com/ev3-micropython/index.html>

[3] ev3dev Home. (s/f). Ev3dev.org. Recuperado el 14 de septiembre de 2023, de <https://www.ev3dev.org/>

[4] Informe 2 - Grupo 3 A 2022 - redmine. (s/f). Uta.Cl. Recuperado el 6 de diciembre de 2023, de <http://pomerape.uta.cl/redmine/documents/1165>

[5] Informe Avance Del Proyecto - grupo 4 A 2022 - redmine. (s/f). Uta.Cl. Recuperado el 6 de diciembre de 2023, de <http://pomerape.uta.cl/redmine/documents/1169>

[6] Informes II (v3) - grupo 6 A 2022 - redmine. (s/f). Uta.Cl. Recuperado el 6 de diciembre de 2023, de <http://pomerape.uta.cl/redmine/documents/975>

[7] Informe Avance - grupo 1 A 2022 - redmine. (s/f). Uta.Cl. Recuperado el 6 de diciembre de 2023, de <http://pomerape.uta.cl/redmine/documents/1158>

[8]. Guía de Uso LEGO MINDSTORM. Recuperado el 4 de enero de 2024, de:

http://chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3_user_guide_esmx-6ac740d3cdd578cc6a52d10d7d173da9.pdf