

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA CIVIL EN COMPUTACIÓN E  
INFORMÁTICA



**Informe de Planificación II**  
**“Golf-Craft”**

**Integrantes:** Nelson Ramírez

Cristofer Pinto

Alvaro Guarachi

Jose Escalante

Matias Suazo

**Asignatura:** Proyecto I

**Profesor:** Humberto Urrutia

|                                                  |    |
|--------------------------------------------------|----|
| 1. Panorama general.....                         | 3  |
| 1.1. Introducción.....                           | 3  |
| 1.2. Objetivos.....                              | 3  |
| 1.2.1. Objetivo general.....                     | 3  |
| 1.2.2. Objetivos específicos.....                | 3  |
| 1.3. Restricciones generales.....                | 4  |
| 1.4. Entregables.....                            | 4  |
| 2. Organización del personal.....                | 4  |
| 2.1. Descripción de los roles.....               | 4  |
| 2.2. Distribución de los roles.....              | 5  |
| 2.3. Mecanismos de comunicación.....             | 5  |
| 3. Planificación del proyecto.....               | 6  |
| 3.1. Actividades.....                            | 6  |
| 3.2. Asignación del tiempo.....                  | 8  |
| 3.4. Gestión de riesgos.....                     | 9  |
| 4. Planificación de recursos.....                | 10 |
| 4.1. Hardware.....                               | 10 |
| 4.2. Software.....                               | 11 |
| 4.3. Estimación de costos.....                   | 11 |
| 5. Análisis y diseño.....                        | 12 |
| 5.1. Requerimientos.....                         | 12 |
| 5.2. Arquitectura.....                           | 13 |
| 5.2.1. Movilización.....                         | 13 |
| 5.2.2. Sensor ultrasónico.....                   | 14 |
| 5.2.3. Sistema de golpeo.....                    | 15 |
| 5.3. Diseño de la interfaz.....                  | 16 |
| 6. Implementación.....                           | 16 |
| 6.1. Fundamentos de Lanzamiento Parabólico. .... | 16 |
| 6.2. Descripción del programa.....               | 19 |
| 6.2.1. Interfaz gráfica.....                     | 19 |
| 6.2.2. Código del servidor.....                  | 20 |
| 6.2.2. Código de funciones.....                  | 20 |
| 7. Resultados.....                               | 21 |
| 8. Problemas y propuesta de solución .....       | 21 |
| 9. Conclusión.....                               | 22 |
| 10. Referencias.....                             | 22 |

# 1. Panorama general

## 1.1. Introducción

En la búsqueda constante de aplicar la tecnología de forma innovadora y creativa, el presente proyecto se centra en el diseño y desarrollo de un robot utilizando el kit de Lego Mindstorms EV3. Este robot tiene como objetivo principal la ejecución de una tarea específica: golpear una pelota de golf de manera precisa. Para lograrlo, se han establecido objetivos generales y específicos que abarcan desde la conceptualización y diseño del robot hasta la programación de sus movimientos y la implementación de un control remoto a través de una interfaz gráfica.

El Lego Mindstorms EV3 es una herramienta versátil que permite combinar la construcción de estructuras físicas con la programación, lo que brinda oportunidades para el aprendizaje de conceptos de ingeniería y la experimentación en el campo de la robótica. En este contexto, este proyecto busca demostrar la capacidad del EV3 para ser utilizado en aplicaciones prácticas y desafiantes, como el desarrollo de un robot capaz de interactuar con su entorno de manera autónoma.

En el transcurso de este informe, se describirán los procesos de diseño y construcción del robot, así como la programación necesaria para permitir su movilidad y precisión en el golpeo de la pelota. Además, se presentará la implementación de una interfaz gráfica que permitirá controlar el robot de manera remota.

## 1.2. Objetivos

### 1.2.1. Objetivo general

Desarrollar un robot utilizando un kit de Lego Mindstorms EV3 el cual sea capaz de golpear una pelota de golf, controlado a través de una interfaz gráfica.

### 1.2.2. Objetivos específicos

- Diseñar un robot que cumpla con las acciones requeridas.
- Construir el robot usando las piezas del Lego Mindstorms EV3
- Programar el robot para que se pueda movilizar y también realizar golpes precisos a una pelota de golf.
- Implementar un sistema de control remoto mediante una interfaz gráfica, la cual permita dirigir el robot.

### **1.3. Restricciones generales**

- Limitación de piezas
- Limitación del tiempo durante el semestre
- El robot debe ser diseñado solamente para golpear una pelota
- Obligatoriedad de control remoto del robot
- Las entradas del robot está limitado a 4 puntos
- Documentación a plataforma Redmine

### **1.4. Entregables**

- Informes y presentaciones:
  - ❖ Planificación y avance del proyecto I.
  - ❖ Código del robot ev3.
  - ❖ Avance del proyecto II
  - ❖ Informe y presentación final.
- Bitácora semanal.
- Wiki y manual de usuario.
- Producto final.

## **2. Organización del personal**

### **2.1. Descripción de los roles**

- Jefe de grupo: Encargado de la toma de decisiones, definición de objetivos, asignación de las tareas y la planificación de las actividades.
- Programador: Encargado de desarrollar el código para que el robot pueda realizar los movimientos además de golpes.
- Ensamblador: Encargado de armar el robot con características acorde a lo requerido.
- Diseñador: Encargado de la creación del logo y el aspecto visual de la interfaz gráfica.
- Documentador: Encargado de realizar los informes, presentaciones y bitácoras.

## 2.2. Distribución de los roles

Se designaron responsables para cumplir labores en un tiempo previamente establecido. Además, se establecieron involucrados, los cuales serán de apoyo para el responsable del rol.

Entre el informe de planificación I y II se realizó un cambio de roles con el fin de que los integrantes se involucren en diferentes áreas del proyecto.

Roles parte I:

| <b>Rol</b>    | <b>Responsable</b> | <b>Involucrados</b>           |
|---------------|--------------------|-------------------------------|
| Jefe de Grupo | Nelson Ramirez     | -                             |
| Programador   | Cristofer Pinto    | Jose Escalante                |
| Ensamblador   | Matias Suazo       | José Escalante                |
| Diseñador     | José Escalante     | Alvaro Guarachi, Matias Suazo |
| Documentador  | Alvaro Guarachi    | Cristofer Pinto               |

Roles parte II:

| <b>Rol</b>    | <b>Responsable</b> | <b>Involucrados</b>           |
|---------------|--------------------|-------------------------------|
| Jefe de grupo | Alvaro Guarachi    | -                             |
| Programador   | Jose Escalante     | Cristofer Pinto               |
| Ensamblador   | Cristofer Pinto    | José Escalante                |
| Diseñador     | Matias Suazo       | Nelson Ramirez                |
| Documentador  | Nelson Ramirez     | Alvaro Guarachi, Matias Suazo |

## 2.3. Mecanismos de comunicación

Una vez concretados los integrantes del grupo se procedió a crear grupos de trabajos tanto en whatsapp el cual servirá para la comunicación con los demás integrantes, coordinación de horarios, etc; Discord con el fin de realizar reuniones de trabajo además de subir documentos en relación al proyecto y Gmail, medio por el cual se realizarán contacto con el profesor.

- Discord
- WhatsApp
- Gmail

### 3. Planificación del proyecto

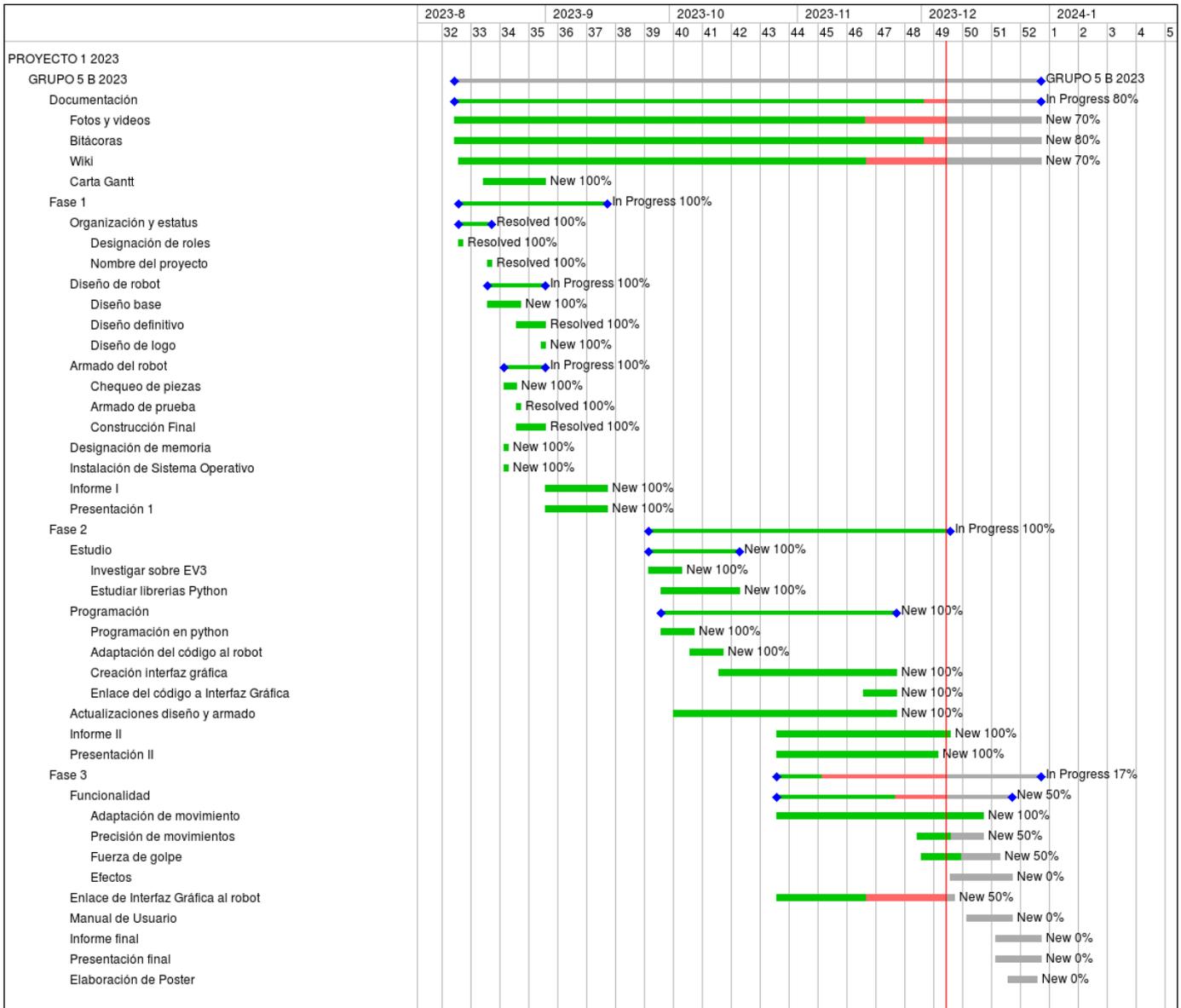
#### 3.1. Actividades

| Actividad                                   | Descripción                                                       | Responsable                                       |
|---------------------------------------------|-------------------------------------------------------------------|---------------------------------------------------|
| Investigar sobre LEGO-EV3                   | Recopilar información sobre el lego EV3.                          | Jose Escalante, Cristofer Pinto                   |
| Chequeo de piezas                           | Buscar las piezas indicadas para el armado.                       | Jose Escalante                                    |
| Redacción de Bitácoras                      | Redactar bitácora.                                                | Matias Suazo                                      |
| Fotos y Videos                              | captura de fotos y videos durante el desarrollo del proyecto.     | Alvaro Guarachi y Cristofer Pinto                 |
| Diseño de Robot                             | Se define la forma del robot.                                     | Jose Escalante, Cristofer Pinto y Álvaro Guarachi |
| Diseño de logo                              | En base al proyecto define el diseño.                             | Matias Suazo                                      |
| Armado de Robot                             | Construcción del robot con las piezas de legos.                   | Jose Escalante y Cristofer Pinto                  |
| Investigación librerías de python           | Recopilar información. mediante videos, documentos, etc.          | Cristofer Pinto y Jose Escalante                  |
| Diseño de la interfaz.                      | Se define un diseño para la interfaz gráfica.                     | Matias Suazo                                      |
| Investigar librerías de Tkinter y servidor. | Recopilar información sobre el funcionamiento de estas librerías. | Jose Escalante y Cristofer Pinto.                 |

|                                           |                                                      |                                  |
|-------------------------------------------|------------------------------------------------------|----------------------------------|
| Actualizar el diseño del robot.           | se actualizó el diseño del robot                     | Cristofer Pinto.                 |
| Estimación de costos                      | Planificar los presupuestos del proyecto.            | Alvaro Guarachi                  |
| Programación de los movimientos del robot | programar las acciones del robot mediante el código. | Jose escalante y Cristofer Pinto |
| Wiki                                      | Creación de la wiki en base a los datos del informe  | Nelson Ramirez                   |
| Carta Gantt                               | Planificar las actividades de la carta gantt         | Alvaro Guarachi                  |
| Cambio de piezas                          | Solicitar piezas adicionales para el robot           | Cristofer Pinto                  |
| Diseño de la interfaz gráfica.            | Definir el aspecto visual de la interfaz.            | Matias Suazo                     |

### 3.2. Asignación del tiempo

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



### 3.3. Gestión de riesgos

| Riesgos                                     | Probabilidad de concurrencia | Nivel de impacto | Acción remedial                                                                     |
|---------------------------------------------|------------------------------|------------------|-------------------------------------------------------------------------------------|
| Corrupción de la tarjeta micro sd.          | 20%                          | 3                | Formateo o cambio tarjeta micro sd.                                                 |
| Pérdida de la tarjeta micro sd.             | 40%                          | 3                | Obtener otra tarjeta sd.                                                            |
| Desarme del robot a causa de un accidente.  | 50%                          | 3                | Reconstruir el robot.                                                               |
| Escasez de piezas.                          | 20%                          | 2                | Solicitar la pieza faltante o buscar una similar.                                   |
| Inconveniente personal de algún integrante. | 60%                          | 2                | Un integrante del grupo debe reemplazar temporalmente el rol del personal faltante. |
| Errores de los software utilizados.         | 15%                          | 1                | Reinstalar o reparar el software dañado.                                            |
| Hardware dañado.                            | 15%                          | 1                | Reemplazar el hardware dañado.                                                      |
| Descarga de batería del Lego EV3.           | 25%                          | 1                | Cargar batería o reemplazar por otra.                                               |

#### Niveles de impacto:

- 1** <- Indiferente
- 2** <- Medio
- 3** <- Perjudicial
- 4** <- Catastrófico

## 4. Planificación de recursos

### 4.1. Hardware

- Notebook
- Lego Mindstorms EV3
- MicroSD
- wi-fi Dongle

### 4.2. Software

- Microsoft Office
- Discord
- Terminal de Ubuntu(Linux)
- Aseprite
- Whatsapp

### 4.3. Estimación de costos

Se definen los costos de hardware en base a los recursos que los integrantes del proyecto utilizarán para el inicio, desarrollo y la finalización del proyecto.

#### Hardware

| Producto            | Cantidad | Costo por unidad(\$) | Costo Total(\$)   |
|---------------------|----------|----------------------|-------------------|
| Notebook            | 2        | 600000               | 1200000           |
| Celular             | 1        | Gratuito             | Gratuito          |
| Lego Mindstorms EV3 | 1        | 1000000              | 1000000           |
| Wifi-Dongle         | 1        | Gratuito             | Gratuito          |
| MicroSD             | 1        | 8000                 | 8000              |
| Pelota de Golf      | 1        | 5000                 | 5000              |
| <b>Total</b>        |          |                      | <b>\$ 2213000</b> |

## Software

Se definen, los costos por software, los que serían programas o servicios que se utilizarán durante la realización del proyecto.

| Producto           | Cantidad | Costo por unidad(\$) | Costo total(\$) |
|--------------------|----------|----------------------|-----------------|
| Microsoft Office   | 1        | 15000                | 15000           |
| Discord            | 1        | 21440                | 21440           |
| Terminal de Ubuntu | 1        | Gratuito             | Gratuito        |
| Aseprite           | 1        | Gratuito             | Gratuito        |
| Whatsapp           | 5        | Gratuito             | Gratuito        |
| <b>Total</b>       |          |                      | 36440           |

## Costo de Mano de Obra

Se definen los costos de mano de obra, estos se definirán en base al sueldo por hora de cada integrante, estos serán distintos debido a la ponderación del cargo, cada integrante cobra su tarifa única. El total de horas correspondiente al proyecto son de 96 horas( 4 meses). Cada integrante trabajará el mismo total de horas que su compañero por lo que si su labor principal ha culminado este apoyará labores de sus colegas.

| Cargo         | Valor Hora Trabajada(\$) | Horas Mensuales Trabajadas | Valor(\$) total Proyecto(96 horas) |
|---------------|--------------------------|----------------------------|------------------------------------|
| Jefe de Grupo | 35000                    | 24                         | 3360000                            |
| Ensamblador   | 24000                    | 24                         | 2304000                            |
| Programador   | 30000                    | 24                         | 2880000                            |
| Diseñador     | 29000                    | 24                         | 2784000                            |
| Documentador  | 24000                    | 24                         | 2304000                            |
| <b>Total</b>  |                          |                            | 13632000                           |

## Costo Proyecto

Suma total de los presupuestos.

| Categoría    | Valor(\$) |
|--------------|-----------|
| Hardware     | 2213000   |
| Software     | 36440     |
| Mano de Obra | 13632000  |
| Total        | 15881440  |

**El presupuesto del proyecto es de \$15.881.440 pesos.**

## 5. Análisis y diseño

### 5.1. Requerimientos:

El robot será controlado por medio de una interfaz gráfica, en el cual se podrá acceder a botones con distintas funcionalidades. Las principales funcionalidades son realizar movimientos en distintas direcciones y golpear la pelota con distintos niveles de fuerzas.

A continuación se indicarán los requisitos funcionales y no funcionales:

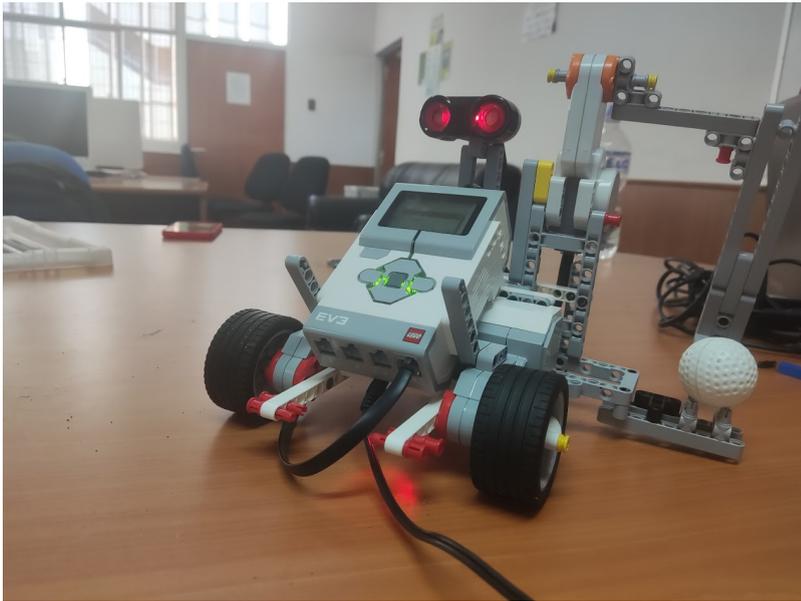
#### Requisitos funcionales:

- El robot debe ser controlado por medio de una interfaz gráfica.
- El robot debe ser diseñado de manera que sea capaz de golpear una pelota de golf de manera precisa.
- El robot debe ser capaz de moverse en distintas direcciones.
- Debe poder realizar golpes simulando un palo de golf.
- El robot debe ser programado para realizar movimientos específicos que le permitan golpear la pelota con precisión.

#### Requisitos no funcionales:

- La comunicación del robot con el servidor debe ser en tiempo real para proveer una experiencia fluida al usuario.
- La batería del robot debe tener buena autonomía para garantizar un buen funcionamiento.
- La interfaz gráfica debe ser amigable para todo tipo de usuario.

## 5.2. Arquitectura.



### 5.2.1. Movilización:



#### Chasis:

El robot está construido sobre un chasis que incorpora dos ruedas delanteras y una bola de acero en la parte trasera, que sirve de apoyo. Las ruedas delanteras son responsables de la dirección y la propulsión, mientras que la trasera proporciona estabilidad.

#### Sistema de Propulsión:

Las dos ruedas delanteras son los motores principales encargados de la propulsión del robot. La velocidad y dirección del robot se controla ajustando la velocidad relativa de las ruedas izquierda y derecha.

#### Dirección:

La dirección se logra variando la velocidad relativa de las ruedas izquierda y derecha. Por ejemplo, si ambas ruedas giran a la misma velocidad, el robot se moverá en línea recta, si una rueda gira más rápido que la otra, el robot girará.

### **Controlador:**

El motor de las ruedas recibe señales de control y las traduce en acciones específicas de movimiento. Esto se realiza por medio de una interfaz gráfica que simula un control remoto.

### **5.2.2. Sensor ultrasónico:**



### **Funcionamiento del sensor:**

El robot incluye un sensor ultrasónico para medir la distancia a objetos cercanos.

Utiliza datos del sensor ultrasónico para realizar ajustes en el movimiento del robot y evitar colisiones.

Implementa un sistema de retroalimentación que utiliza los datos del sensor ultrasónico para obtener la distancia desde el robot hacia el objetivo, lo cual sirve para poder realizar un golpe preciso con el palo de golf.

### **5.2.3. Sistema de golpeo:**

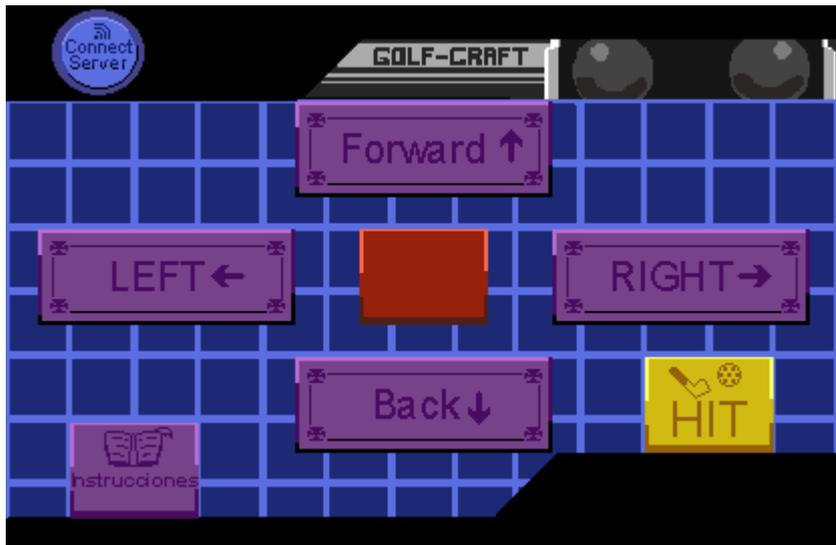


#### **Implementación del Palo de Golf:**

El robot cuenta con un palo de golf montado en el lateral izquierdo, permitiendo realizar acciones de golpeo a una pelota de minigolf, la cual se encuentra sobre una pequeña base, además de tener un chasis que se ocupará para devolver el brazo a su posición original .

Integra un motor que controla el movimiento, el cual recibe señales por un botón que se encuentra en la interfaz gráfica, la cual permite seleccionar la velocidad con la cual girará el palo de golf.

### 5.3 Diseño de la interfaz.



La interfaz del robot tendrá 6 botones los cuales son para cada movimiento tendrá una función que hará el robot:

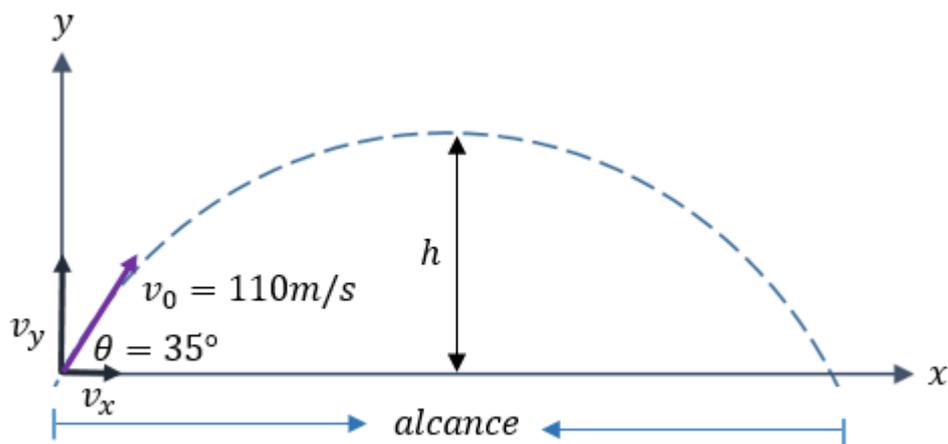
- 1- **"Forward"**: este botón tendrá la función de avanzar solo hacia adelante.
- 2- **"Right"**: este botón tendrá la función de mover las robot al lado de la derecha.
- 3- **"Left"**: este botón tendrá la función de mover al robot al lado de la izquierda.
- 4- **"Back"**: este botón tendrá la función de mover el robot hacia atrás.
- 5- **"Connect server"**: este botón tendrá la función de conectar el robot al servidor.
- 6- **"Hit"**: este botón tendrá la función de golpear con el brazo del robot.

## 6. Implementación

### 6.1 Fundamentos de Lanzamiento Parabólico.

Aplicando conocimientos de Mecánica Clásica Cuántica Universal el Robot realizará un golpe de un palo o brazo hacia una pelota de golf el cual realizará un movimiento parabólico.

*Referencia del lanzamiento parabólico.*



Para el cálculo se ocuparán las fórmulas de cinemática (MRU y MRUV) aprendidas en la asignatura de Mecánica Clásica.

### Formulas de MRU

$$X = X_0 + V_0 \cdot t$$

### Formulas de MRUV

$$Y = Y_0 + V_0 \cdot t + \frac{1}{2} \cdot a \cdot (t)^2$$

### Formulas de Velocidad

$$V_x = V_0 \cdot \cos \theta$$

$$V_y = V_0 \cdot \sin \theta$$

1. Fórmula (MRUV) la ocuparemos para calcular la distancia o el tiempo de vuelo que se demora la pelota hasta llegar a su punto.
2. Fórmula (MRU) la ocuparemos para calcular la distancia.

II) Datos obtenidos de la prueba del robot.

$$V(\text{inicial}) = 10 \text{ (m/s)}$$

$$\text{ángulo} = 45 \text{ (grados)}$$

$$\text{Masa}(\text{pelota}) = 0.5 \text{ (kg)}$$

$$X(\text{total}) = 0.3 \text{ (m)}$$

$$Y(\text{inicial}) = 0.02 \text{ (m)}$$

$$g = 9.81 \text{ (m/s}^2\text{)}$$

- Velocidades iniciales del tiro:

**Eje x** ->  $V_x = V(\text{inicial}) \cdot \cos 45$

$$V_x = 10 \cdot \cos 45$$

$$V_x = 7 \text{ (m/s)}$$

**Eje y** ->  $V_y = V(\text{inicial}) \cdot \sin 45$

$$V_y = 10 \cdot \sin 45$$

$$V_y = 7 \text{ (m/s)}$$

Cálculo del tiempo que se demora en recorrer la pelota hasta llegar al suelo:

$$Y = y(\text{inicial}) + V_y(\text{inicial}) \cdot t + \frac{1}{2} g \cdot t^2$$

$$0 = 0.02 + 7 \cdot t + \frac{1}{2} \cdot (-9.81) \cdot t^2$$

$$0 = 0.02 + 7 \cdot t - 4.9 \cdot t^2$$

Por método de bescara:

$$t = 1.43 \text{ (s)}$$

- Cálculo de la altura máxima que llegará a tener la pelota.

**$v = 0$ , ya que al llegar a su altura máxima se detiene su ascenso y comienza su descenso.**

$$Y(\text{máxima}) = Y(\text{inicial}) + V_y \cdot t + \frac{1}{2} \cdot g \cdot (t)^2$$

$$y(\text{máxima}) = 0.02 + 0 \cdot 1.43 + 4.9 \cdot (1.43)^2$$

$$y(\text{máxima}) = 0.02 + 0 + 4.9 \cdot 2.04$$

$$y(\text{máxima}) = 0.02 + 10$$

$$y(\text{máxima}) = 10.02 \text{ (m)}$$

## 6.2 Descripción del programa

### 6.2.1 Interfaz gráfica:

```
1 import tkinter as tk
2 from tkinter import messagebox
3 import os
4 import socket
5
6 class ControlRemotoGUI:
7     def __init__(self, master):
8         self.master = master
9         master.title("Control Remoto LEGO EV3")
10
11         server= socket.socket()
12         ip= "192.168.1.84"
13         port= 8080
14
15         def avanzar():
16             server.send(bytes([ord['w']]))
17
18         def retroceder():
19             server.send(bytes([ord['s']]))
20
21         def izquierda():
22             server.send(bytes([ord['a']]))
23
24         def derecha():
25             server.send(bytes([ord['d']]))
26
27         def golpear():
28             server.send(bytes([ord['e']]))
29
30         def detener():
31             server.send(bytes([ord[' ']]))
32
33         def conectar(addrres):
34             try:
35                 server.connect((addrres, port))
36                 messagebox.showInfo("Conectado, se establecio la conexion correctamente")
37             except socket.error:
38                 messagebox.showwarning("Conexion erronea, verifique la direccion ip")
39
```

```
40
41 # Obtener la ruta actual
42 script_dir = os.path.dirname(os.path.abspath(__file__))
43
44 # Cargar imagenes desde la carpeta
45 self.img_avanzar = self.load_image(os.path.join(script_dir, "Arriba.gif"))
46 self.img_retroceder = self.load_image(os.path.join(script_dir, "Abajo.gif"))
47 self.img_izquierda = self.load_image(os.path.join(script_dir, "Izquierda.gif"))
48 self.img_derecha = self.load_image(os.path.join(script_dir, "Derecha.gif"))
49 self.img_golpear = self.load_image(os.path.join(script_dir, "Stop.gif"))
50 self.img_conectar = self.load_image(os.path.join(script_dir, "Conectar.gif"))
51
52 # Configuracion de la ventana
53 master.geometry("770x500")
54 master.resizable(width=False, height=False)
55
56 # Cargar la imagen de fondo y almacenarla como un atributo de instancia
57 self.bg_image = tk.PhotoImage(file=os.path.join(script_dir, "Background.gif"))
58
59 # Mostrar la imagen de fondo en un Label
60 label = tk.Label(master, image=self.bg_image)
61 label.place(x=0, y=0, relwidth=1, relheight=1)
62
63 self.btn_avanzar = tk.Button(master, image=self.img_avanzar, command=self.avanzar)
64 self.btn_avanzar.place(relx=0.5, rely=0.25, anchor=tk.CENTER)
65
66 self.btn_retroceder = tk.Button(master, image=self.img_retroceder, command=self.retroceder)
67 self.btn_retroceder.place(relx=0.5, rely=0.75, anchor=tk.CENTER)
68
69 self.btn_izquierda = tk.Button(master, image=self.img_izquierda, command=self.izquierda)
70 self.btn_izquierda.place(relx=0.25, rely=0.5, anchor=tk.CENTER)
71
72 self.btn_derecha = tk.Button(master, image=self.img_derecha, command=self.derecha)
73 self.btn_derecha.place(relx=0.75, rely=0.5, anchor=tk.CENTER)
74
75 # Boton de golpear
76 self.btn_golpear = tk.Button(master, image=self.img_golpear, command=self.golpear)
77 self.btn_golpear.place(relx=0.5, rely=0.5, anchor=tk.CENTER)
78
79 # Boton de conectar con el servidor
80 self.btn_conectar = tk.Button(master, image=self.img_conectar, command=self.conectar(ip))
81 self.btn_conectar.place(relx=0.15, rely=0.1, anchor=tk.CENTER)
82
```

### 6.2.2 Código del servidor.

```
1 import socket
2
3 s= socket.socket()
4 print("Socket creado")
5 port = 8080
6 s.bind(('', port))
7 print("El socket se creo con puerto: {}".format(port))
8 s.listen(5)
9 print("EL socket is listening...")
10 connect, addr = s.accept()
11 print("Se conecto a {}".format(addr))
12 while True:
13     rawByte = connect.recv(1)
14     char = rawByte.decode('utf-8')
15     if (char == 'w'):
16         Mov_Avanzar()
17     if (char == 's'):
18         Mov_Retroceder()
19     if (char == 'd'):
20         Mov_Izquierda()
21     if (char == 'a'):
22         Mov_Derecha()
23     if (char == 'e'):
24         Mov_Golpear()
25     if (char == ' '):
26         Mov_Detener()
27
```

### 6.3.3 Código de la funciones del robot.

```
1 from ev3dev2.motor import LargeMotor, MoveTank,OUTPUT_A, OUTPUT_B, OUTPUT_C
2 from time import sleep
3
4 m= MoveTank(OUTPUT_A,OUTPUT_B)
5 n= LargeMotor(OUTPUT_C)
6 l= LargeMotor(OUTPUT_A)
7 r= LargeMotor(OUTPUT_B)
8
9 def Mov_Avanzar():
10     m.run_forever(speed_sp=-400)
11
12 def Mov_Retrocer():
13     m.run_forever(speed_sp=400)
14
15 def Mov_Izquierda():
16     l.run_forever(speed_sp=-60)
17     r.run_forever(speed_sp=60)
18
19 def Mov_Derecha():
20     l.run_forever(speed_sp=60)
21     r.run_forever(speed_sp=-60)
22
23 def Mov_Golpear():
24     n.on_for_deggres(speed=10, deggres=45)
25     n.on_for_deggres(speed=100, deggres=-105)
26     n.off()
27
28 def Detener():
29     m.stop()
30     n.stop()
```

## 7. Resultados

- El cuerpo del robot ya se encuentra construido casi en su totalidad.
- La movilidad del robot está lista, se pueden controlar todas sus acciones de movimiento.
- El palo de golf aún está en proceso de modificación.
- La interfaz gráfica cuenta con varios botones, asignando distintas acciones para cada uno, como avanzar, retroceder, golpear con el palo de golf y conectar con el servidor.
- EL código de la interfaz gráfica se encuentra terminado.
- El código de las funciones del robot no está terminado por completo, aún está en constante cambio agregando nuevas funciones.
- La conexión será vía remota
- Wiki, Carta Gantt actualizada
- Bitácoras, informes y presentaciones al día

## 8. Problemas y propuesta de solución:

| <b>Problemas</b>                                                                                        | <b>Propuesta de solución</b>                                                                   |
|---------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| Tiempos pocos prácticos de la carta Gantt y modificación obligada debido a sucesos externos al proyecto | Reordenamiento de carta, además de revisión y modificación de fechas.                          |
| Inestabilidad en red Wifi del departamento, dificultando la conexión del robot ev3 y laptop.            | Conexión a red 5G de equipo móvil de los integrantes.                                          |
| Dificultad para reforzar el cuerpo del robot y darle estabilidad para el lanzamiento parabólico         | Adición de piezas a los costados del robot para darle estabilidad y también reforzar su brazo. |

## 9. Conclusión

El proyecto va por buen rumbo a cumplir el objetivo principal, desde la concepción hasta la implementación de la interfaz gráfica. Todo este proceso ha sido complementado con los conocimientos obtenidos en asignaturas de plan común, como asignaturas de programación.

A lo largo de este proyecto, se ha detallado todo el proceso de diseño, construcción e implementación, esto con el fin de llevar un registro de todo el desarrollo. Por otro lado, ha ocurrido problemas en distintos apartados, uno de los principales es el tema del cálculo para lograr que la pelota llegue a el objetivo deseado, el cual aún se encuentra en proceso para lograr la solución.

Además, durante este proceso,, se ha destacado la versatilidad del Lego mindstorms EV3, el cual ha sido adaptado y programado para poder simular un palo de golf.

## 10. Referencias

- <https://www.ev3dev.org>
- <https://robotsquare.com/2015/10/06/explor3r-building-instructions/>
- <https://ev3dev-lang.readthedocs.io/projects/python-ev3dev/en/stable/>
- <https://www.aposteriori.com.sg/wp-content/uploads/2018/02/Python-Commands-on-EV3DEV.pdf>