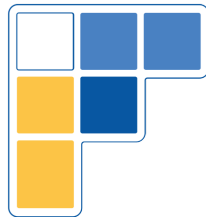


UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA



DEPARTAMENTO DE INGENIERÍA EN COMPUTACIÓN E INFORMÁTICA



Avance de Proyecto “GOLFFENHEIMER”

Alumno(s): Bruno Améstica
Jorge Cáceres
Ignacio Garrido
Katalina Oviedo
Fernando Pizarro

Asignatura: Proyecto I

Profesor: Humberto Urrutia

Diciembre – 2023



Historial de Cambios

Fecha	Versión	Descripción	Autor(es)
29/08/2023	1.0	Formulación del Proyecto	Katalina Oviedo
14/11/2023	2.0	Desarrollo del Proyecto	Bruno Améstica Ignacio Garrido Katalina Oviedo Fernando Pizarro



Tabla de Contenido

1. Panorama General	4
1.1. Introducción	4
1.2. Objetivos	4
1.2.1. Objetivo General	4
1.2.2. Objetivos Específicos	4
1.3. Restricciones	4
1.4. Entregables	5
2. Organización del Personal	6
2.1. Descripción de los Roles	6
2.2. Personal que cumplirá los Roles	6
2.3. Mecanismos de Comunicación	6
3. Planificación del Proyecto	7
3.1. Actividades	7
3.2. Asignación de Tiempo	8
3.3. Gestión de riesgos	9
4. Planificación de los Recursos	10
4.1. Hardware	10
4.2. Software	10
4.3. Estimación de Costos	10
4.3.1. Costos de Hardware	10
4.3.2. Costos de software	10
4.3.3. Costo Gestión	11
4.3.4. Costo Total	11
5. Análisis y Diseño	12
5.1. Especificación de Requerimiento	12
5.2. Arquitectura	12
5.3. Interfaz Gráfica	13
6. Implementación	15
6.1. Fundamentos de Projectiles.	15
6.2. Descripción de los programas	17
6.3. Diagramas.	18
7. Resultados	19
7.1. Estado actual del proyecto	19
7.2. Problemas encontrados y solución propuesta	19
8. Pruebas	20
8.1. Descripción de las pruebas realizadas	20
8.2. Resultado de las pruebas	20
9. Conclusión	21
10. Referencias	22



1. Panorama General

1.1. Introducción

El proyecto “Golffenheimer” consiste en diseñar y programar un robot capaz de golpear una pelota de golf. Para realizar este proyecto, se utilizó el kit de Lego Mindstorms EV3, el cual es una herramienta educativa que permite a los estudiantes aprender sobre programación, mediante la construcción y control de robots con diferentes funciones y capacidades. El kit incluye una unidad central inteligente que puede programarse con diferentes lenguajes y que puede interactuar con motores y sensores.

En el presente informe se definirán principalmente los objetivos que se pretenden alcanzar con el proyecto, tomando en cuenta posibles contratiempos que pudieran dificultar su desarrollo; los roles de los integrantes del grupo y la distribución de tareas entre estos; el tiempo necesario para realizar el proyecto y el costo monetario de su producción. Además, se proporciona una especificación de requisitos, arquitectura e interfaz gráfica. Se abordan los fundamentos teóricos y técnicos relacionados con la implementación del proyecto y por último se señalan los resultados obtenidos así como los problemas y soluciones que ha tenido el proyecto.

1.2. Objetivos

1.2.1. Objetivo General

Configurar un robot utilizando el set LEGO MINDSTORMS EV3, que sea capaz de golpear una pelota de golf a través de una interfaz gráfica programada en Python que permita controlar todas y cada una de las acciones del robot.

1.2.2. Objetivos Específicos

- Diseñar software efectivo para el control del robot.
- Construir un diseño de robot útil, que cumpla con lo requerido.
- Realizar pruebas para evaluar el funcionamiento y el rendimiento del robot, para identificar y corregir posibles errores o mejoras.

1.3. Restricciones

Para llevar a cabo el proyecto de forma exitosa, es necesario cumplir con ciertas restricciones.

- Piezas: trabajar de acuerdo a las piezas de lego disponibles.
- Puertos: el robot cuenta con cuatro puertos para motores y cuatro para sensores, por lo que no es posible utilizar más de 4 motores o sensores.
- Lenguaje de programación: la programación debe ser realizada en el lenguaje Python.
- Control: el control del robot debe ser vía remota.



- Conexión: la conexión del robot con el computador debe ser mediante la misma red wifi.
- Sistema Operativo: el proyecto se debe realizar con el sistema operativo Linux

1.4. Entregables

Los entregables a lo largo del proyecto son los siguientes:

- Informes del proyecto:
 - Formulación del proyecto.
 - Avance del proyecto.
 - Informe final del proyecto.
- Presentaciones del proyecto:
 - Formulación del proyecto.
 - Avance del proyecto.
 - Presentación final del proyecto.
- Bitácoras semanales.
- Carta gantt.
- Wiki.
- Producto final.



2. Organización del Personal

2.1. Descripción de los Roles

- **Jefe de grupo:** encargado de organizar y supervisar al equipo de trabajo.
- **Ensamblador:** encargado de diseñar y armar el robot de forma que cumpla con lo requerido.
- **Programador:** encargado de desarrollar el código del robot para que ejecute sus movimientos correctamente.
- **Documentador:** encargado de realizar los informes, presentaciones, bitácoras, fotos, manual de usuario.
- **Diseñador:** encargado de diseñar el logo, diseñar la wiki y la interfaz gráfica.

2.2. Personal que cumplirá los Roles

Rol	Responsable
Jefe de grupo	Fernando Pizarro
Programador	Jorge Cáceres Katalina Oviedo Fernando Pizarro
Ensamblador	Jorge Cáceres Bruno Améstica
Documentador	Ignacio Garrido Bruno Améstica Katalina Oviedo Fernando Pizarro
Diseñador	Bruno Améstica

Tabla 1: Roles.

2.3. Mecanismos de Comunicación

Para una comunicación efectiva entre los miembros del equipo se estableció como medio de comunicación la aplicación Discord. Por este medio se establecen los horarios de las reuniones, compartimos la documentación del proyecto, como videos, fotos, informes, bitácoras, etc. También se proponen nuevas ideas para la mejora del proyecto.



3. Planificación del Proyecto

3.1. Actividades

La planificación de las actividades es esencial para el éxito del proyecto, pues son las acciones que nos permitirán lograr los objetivos específicos. En esta etapa, determinaremos el alcance de la planificación de todas las actividades del proyecto, teniendo en cuenta las prioridades, los recursos, los tiempos y las funcionalidades de cada tarea.

Actividad	Descripción	Responsable
Asignación de roles	Asignación de roles dentro del equipo.	Fernando Pizarro
Contabilizar piezas	Recuento de piezas del kit de lego.	Katalina Oviedo
Búsqueda de ideas	Búsqueda de ideas sobre la estructura del robot.	Jorge Cáceres
Construcción del robot	Armado de la estructura del robot.	Jorge Cáceres
Instalación SO	Se instala el SO en la tarjeta micro SD.	Fernando Pizarro
Redacción de bitácoras	Registro de actividades que se realizan semanalmente.	Fernando Pizarro
Redacción de Carta Gantt	Planificación de actividades a realizar durante el semestre.	Ignacio Garrido
Wiki	Se comparte información del proyecto.	Bruno Améstica
Informe I	Redacción de informe final del proyecto	Katalina Oviedo
Presentación I	Elaborar presentación final del proyecto.	Ignacio Garrido
Estudiar librería	Estudio de la librería ev3dev-lang-python.	Katalina Oviedo
Programación de movimientos	Codificar los movimientos del robot.	Katalina Oviedo



Programar interfaz gráfica	Diseño de interfaz gráfica.	Jorge Cáceres
Creación del servidor	Elaboración del servidor.	Katalina Oviedo
Informe II	Redacción de informe final del proyecto	Ignacio Garrido
Presentación II	Elaborar presentación de avance del proyecto.	Bruno Améstica
Manual de usuario	Escritura del manual de usuario.	Fernando Pizarro
Informe final	Redacción de informe final del proyecto	Katalina Oviedo
Presentación final	Elaborar presentación final del proyecto.	Ignacio Garrido

Tabla 2: Actividades.

3.2. Asignación de Tiempo

Se ha elaborado una Carta Gantt para estimar el tiempo que se empleará en cada actividad del proyecto y organizar el tiempo entre actividades de forma más eficiente.

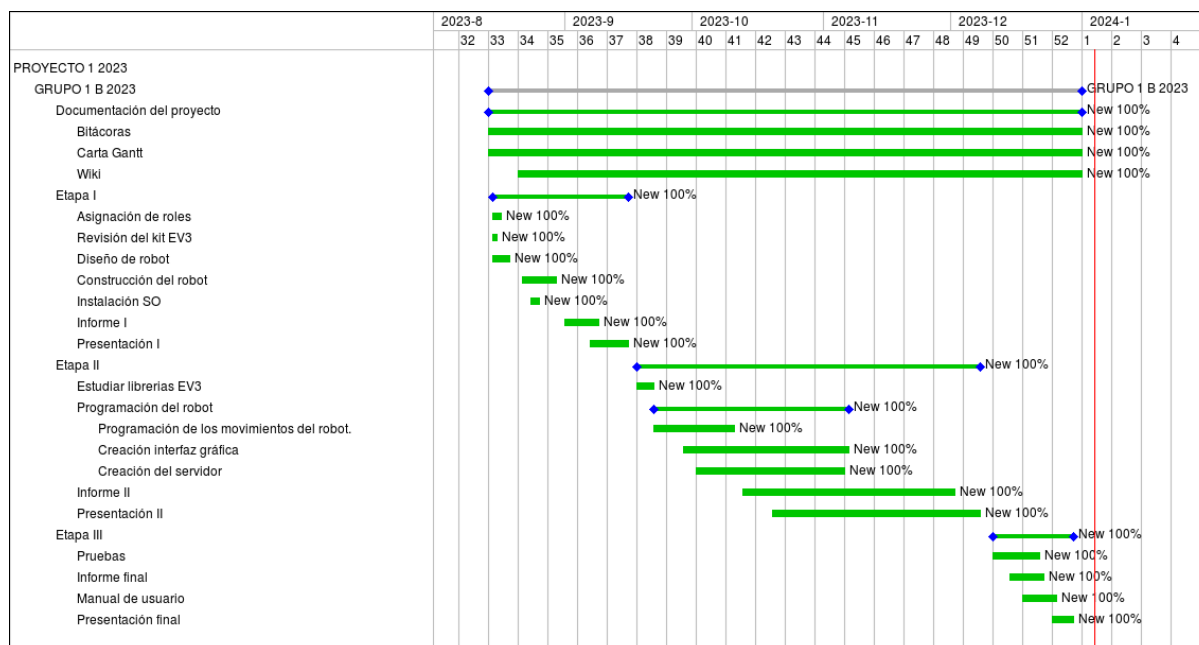


Ilustración 1: Carta Gantt.



3.3. Gestión de riesgos

Se elabora la siguiente tabla para establecer las acciones a tomar ante los posibles riesgos, según su nivel de impacto y probabilidad de ocurrencia. Los niveles de impacto son:

1. Catastrófico.
2. Crítico.
3. Marginal.
4. Despreciable.

Las acciones a tomar se determinarán según la categoría de cada riesgo.

Riesgos	Probabilidad de ocurrencia	Nivel de impacto	Acción remedial
Falta de piezas.	80%	3	Buscar la pieza faltante o reemplazarla por una similar.
Ausencia del personal.	50%	4	Reorganizar al equipo para que se avance en la tarea del personal ausente.
Incumplimiento de tareas.	40%	2	Reasignación de roles para encontrar el más óptimo.
Reconstrucción total del robot por no cumplir lo requerido.	20%	1	Buscar nuevas ideas que cumplan con lo pedido y llevarlas a cabo.
Problemas de Hardware.	20%	1	Solicitar reemplazo del hardware.
Desarme del robot a causa de una caída.	15%	2	Volver a armar el robot.
Desgaste y/o mal funcionamiento de motores.	15%	3	Reemplazar la pieza en mal estado.

Tabla 3: Gestión de riesgos.



4. Planificación de los Recursos

4.1. Hardware

- Kit Lego MINDSTORMS (EV3)
- Wifi dongle USB
- Micro SD (8GB)
- Notebook (Arrendados)
- Router

4.2. Software

- Visual Studio Code

4.3. Estimación de Costos

Los costos relacionados al hardware, software y gestión del personal se muestran en las siguientes tablas:

4.3.1. Costos de Hardware

Productos	Cantidad	Valor unitario	Valor total
Kit Lego MINDSTORMS (EV3)	1	\$1.229.990	\$1.229.990
Dongle USB Wifi	1	\$7.990	\$7.990
Micro SD (8 GB)	1	\$5.000	\$5.000
Notebook (Arrendado)	5	\$ 50.000	\$1.000.000
Router	1	\$15.000	\$15.000
Total			\$2.257.980

Tabla 4: Costos de Hardware.

4.3.2. Costos de software

El único software que se utilizó para poder realizar la codificación del robot fue Visual Code Studio, el cual se gratuito.



4.3.3. Costo Gestión

Encargado	Personas	Valor hora	Horas mensuales	Valor total por 4 meses
Jefe de Grupo	1	\$9.000	38,5	\$1.386.000
Programador	1	\$7.000	38,5	\$1.078.000
Ensamblador	1	\$5.000	38,5	\$770.000
Documentador	1	\$6.500	38,5	\$1.001.000
Diseñador	1	\$5.500	38,5	\$847.000
Total				\$5.082.000

Tabla 5: Costos de Gestión.

4.3.4. Costo Total

Costos	Costo total
Costos de Hardware	\$2.257.980
Costos de Software	\$0
Costos de Gestión	\$5.082.000
Costo total del proyecto	\$7.339.980

Tabla 6: Costos totales.

5. Análisis y Diseño

5.1. Especificación de Requerimiento

- Requerimientos funcionales:
 - El robot debe ser controlado mediante la implementación de una interfaz gráfica.
 - El robot debe tener la capacidad de moverse.
 - El robot debe tener la capacidad de golpear una pelota de golf.
- Requerimientos no funcionales:
 - La elaboración del código debe ser con el lenguaje de programación Python.
 - La interfaz gráfica debe contener los movimientos del robot tales como moverse hacia delante, hacia atrás, hacia el lado izquierdo y hacia el lado derecho, además de mover el brazo del robot para lograr realizar el golpe de la pelota.
 - El robot debe ser construido con piezas del kit LEGO MINDSTORMS EV3.
 - La estructura del robot debe ser estable y capaz de poder realizar todo tipo de movilidad.

5.2. Arquitectura

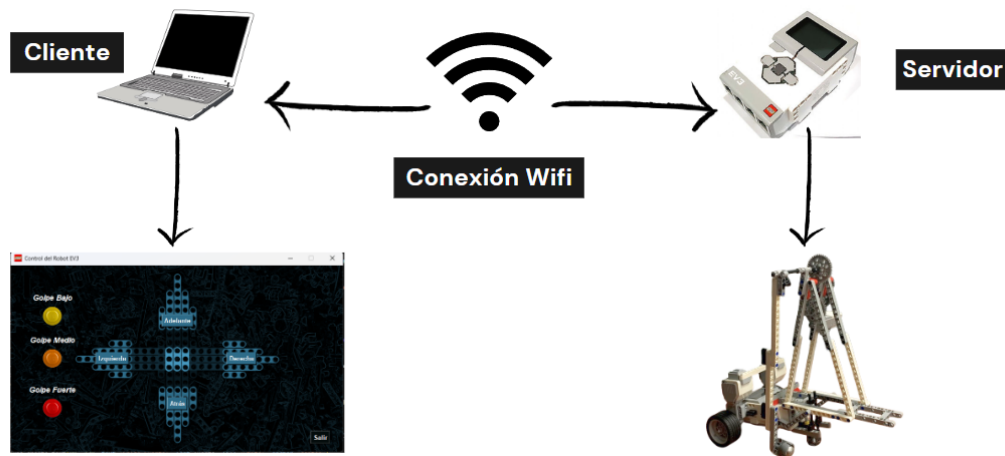


Ilustración 2: Arquitectura.

1. El notebook y el robot deben estar conectados a la misma red WiFi para un correcto funcionamiento.
2. Se establece una conexión SSH entre el notebook y el robot.
3. Se inicia el archivo servidor.py para obtener la conexión remota con el notebook.
4. El usuario debe iniciar la interfaz gráfica para controlar el robot a distancia.
5. La interfaz se conectará al servidor y el usuario podrá controlarlo.

Con estos procesos realizados, el robot estará en operación.

5.3. Interfaz Gráfica

En la ventana inicial de la interfaz, el usuario deberá ingresar la ip correspondiente, si ingresa una ip inválida aparecerá un mensaje de error en el que notificará que hubo un error al conectar, en el caso de que se ingrese una ip válida notificará que la conexión fue exitosa. En la esquina inferior derecha se encuentra el botón “SALIR” el cual permite salir del programa en cualquier momento.

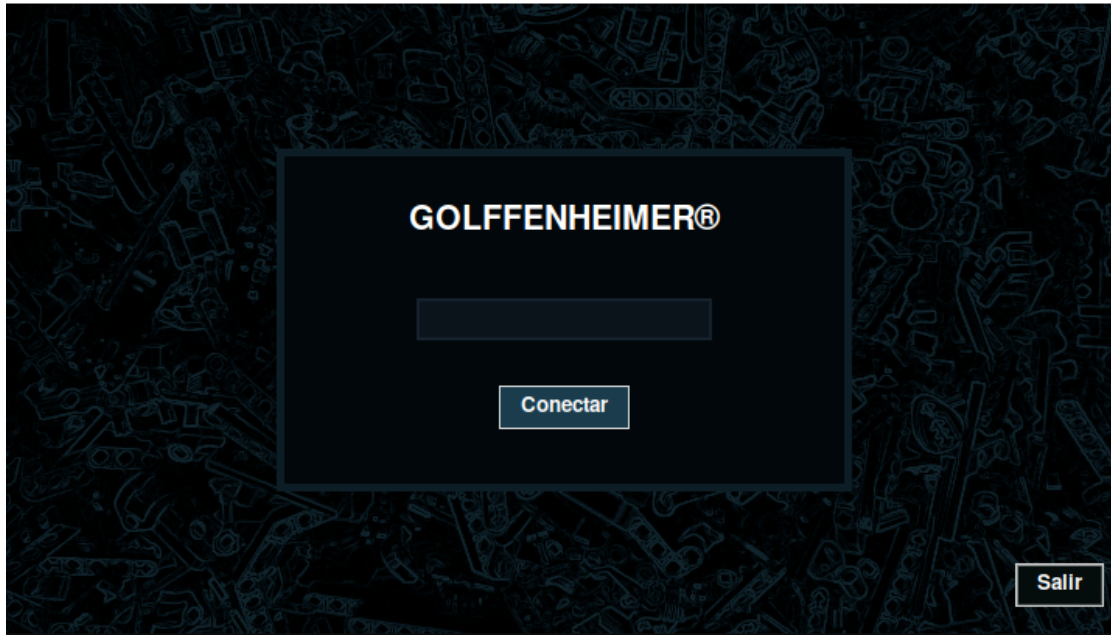


Ilustración 3: Ventana inicial.



Ilustración 4: Mensaje de error.



Ilustración 5: Mensaje de conexión exitosa.

Luego de que la conexión se haya establecido exitosamente, se abrirá la ventana que permite al usuario controlar los movimientos del robot. Los botones que se pueden encontrar son:

- Botón “ADELANTE” desplaza al robot hacia adelante.
- Botón “ATRÁS” desplaza al robot hacia atrás.
- Botón “IZQUIERDA” gira al robot hacia la izquierda.
- Botón “DERECHA” gira al robot hacia la derecha.
- Botón “GOLPE BAJO” el cual tiene la funcionalidad de hacer que el robot realice el golpe con una intensidad mínima.
- Botón “GOLPE MEDIO” el cual tiene la funcionalidad de hacer que el robot realice el golpe con una intensidad media .
- Botón “GOLPE FUERTE” el cual tiene la funcionalidad de hacer que el robot realice el golpe con una fuerte intensidad.
- Botón “SALIR” permite salir del programa en cualquier momento.

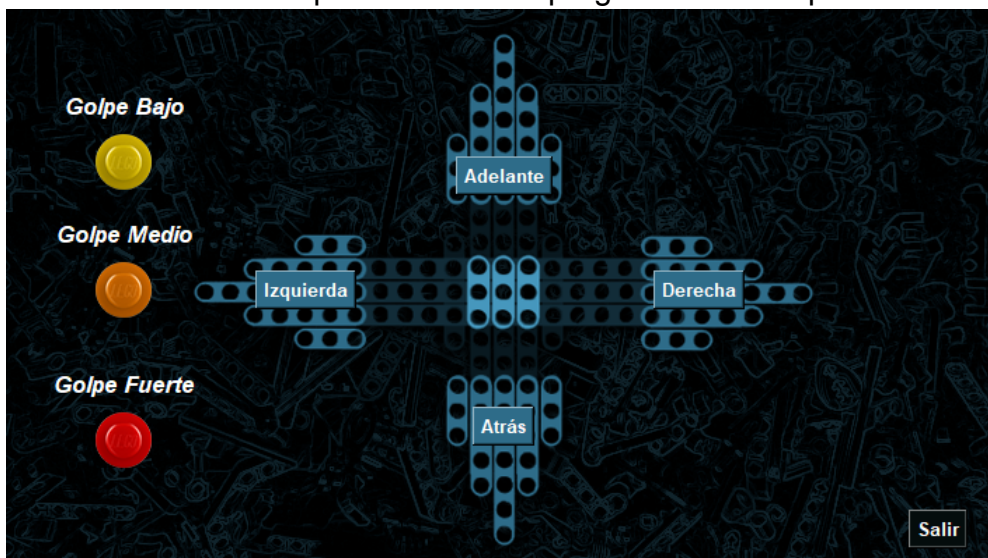


Ilustración 6: Ventana funciones.



6. Implementación

6.1. Fundamentos de proyectiles.

Se tiene en cuenta de que el brazo del robot sigue un movimiento armónico simple, el cual se deja caer desde cierta altura para luego golpear la pelota de golf. Para calcular la velocidad del brazo con la que se golpea la pelota se utilizará la ley de conservación de la energía.

- Energía potencial:

$$U = m \times g \times h$$

- Energía cinética:

$$K = \frac{1}{2} \times m \times v^2$$

Datos:

$$h = 0.5m$$

$$g = 9.8 \frac{m}{s^2}$$

$$K_i + U_i = K_f + U_f$$

- Despejando v y reemplazando los datos:

$$K_i + U_i = K_f + U_f$$

$$U_i = K_f$$

$$m \times g \times h = \frac{1}{2} \times m \times v^2$$

$$v = \sqrt{2gh}$$

$$v = 3.1 \frac{m}{s}$$

- Luego, se calcula la velocidad de la pelota al momento del impacto.

Datos:

$$m_b = 0.1 kg$$

$$M_p = 0.003kg$$

$$m_b \times v_{ib} + M_p \times v_{ip} = v(m_b + M_p)$$

$$m_b \times v_{ib} = v(m_b + M_p)$$

$$v = \frac{m_b \times v_{ib}}{(m_b + M_p)}$$

$$v = 3 \frac{m}{s}$$

Con estos datos calculados podemos analizar el movimiento de la pelota de golf el cual es un movimiento parabólico bidimensional, es decir, el proyectil se desplaza en dos dimensiones: horizontalmente (en el eje x) y verticalmente (en el eje y).

Las fórmulas utilizadas para predecir trayectoria de la pelota son:

- Movimiento Rectilíneo Uniforme (M.R.U): En el eje horizontal

$$x_f = x_0 + v_{0x} \times t$$

$$v_{0x} = v_0 \cos\theta = \text{cte}$$

- Movimiento Rectilíneo Uniforme Acelerado (M.R.U.A): En el eje vertical

$$y_f = y_0 + v_{0y} \times t + \frac{1}{2} \times g \times t^2$$

$$v_{0y} = v_0 \times \text{sen}\theta$$

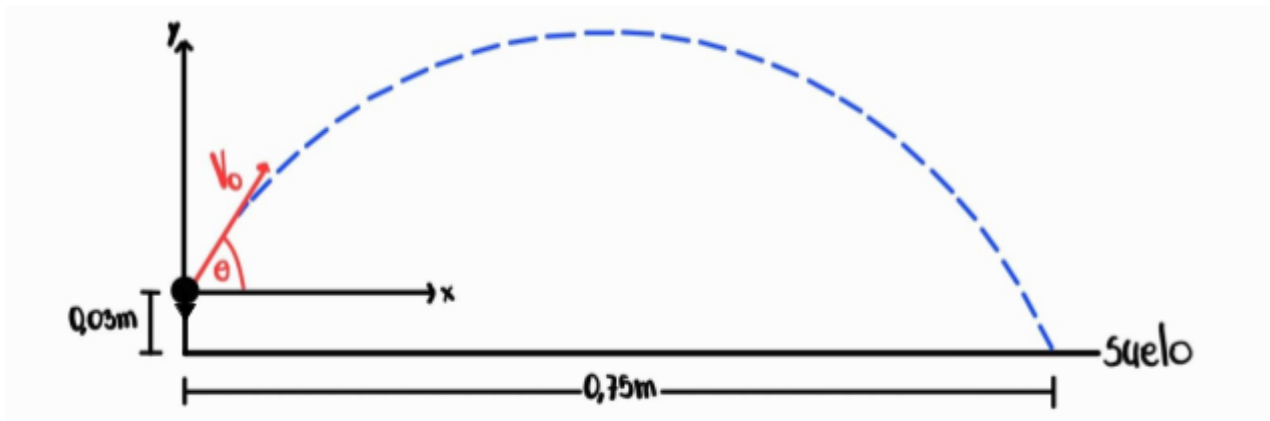


Ilustración 7: Movimiento pelota de golf.

Datos:

$$x_0 = 0m$$

$$y_0 = 0m$$

$$\theta = 45^\circ$$

$$x_f = 0.75m$$

$$y_f = -0.03m$$

$$v_0 = 3 \frac{m}{s}$$

$$g = -9.8 \frac{m}{s^2}$$

- Reemplazando y despejando t:

$$x_f = x_0 + v_{0x} \times t$$

$$0.75 = v_0 \cos 45^\circ \times t$$



$$t = \frac{0.75}{3 \times \cos 45^\circ}$$

$$t = \frac{0.75}{3 \times \cos 45^\circ}$$

$$t = 0.35s$$

- El tiempo transcurrido desde que se golpea la pelota hasta que llega al suelo es de 0.35s
- Calculando la altura máxima:

$$y_f = y_0 + v_{0y} \times t + \frac{1}{2} \times g \times t^2$$

$$y_f = 3 \times \text{sen } 45^\circ \times 0.35 - 4.9 \times (0.35)^2$$

$$y_f = 3 \times \text{sen } 45^\circ \times 0.35 - 4.9 \times (0.35)^2$$

$$y_f = 0.14m$$

- La altura máxima que alcanzará la pelota será de 0.14m cuando la velocidad inicial de la pelota es $3 \frac{m}{s}$

6.2. Descripción de los programas

Las funciones principales son las del movimiento del robot y son las siguientes:

- def adelante(): el robot se mueve en dirección hacia adelante.
- def atras(): el robot se mueve en dirección hacia atrás.
- def derecha(): el robot se mueve en dirección hacia la derecha.
- def izquierda(): el robot se mueve en dirección hacia la izquierda.
- def golpe_bajo(): el robot mueve el brazo para realizar un golpe con intensidad baja.
- def golpe_medio(): el robot mueve el brazo para realizar un golpe con intensidad media.
- def golpe_fuerte(): el robot mueve el brazo para realizar un golpe con intensidad máxima.
- def stop(): Detiene al robot luego de realizar una función.

Luego, están las funciones implementadas en la interfaz gráfica (GUI):

- def adelante(): envía el carácter "w" al socket.
- def atras(): envía el carácter "s" al socket.
- def derecha(): envía el carácter "d" al socket.
- def izquierda(): envía el carácter "a" al socket.
- def golpe_bajo(): envía el carácter "r" al socket.

- def golpe_medio(): envía el carácter “t” al socket.
- def golpe_fuerte(): envía el carácter “y” al socket.
- def on_release(): el robot se parará cuando el cliente deje de hacer click a algún botón.
- def abrirventana(): se abre una nueva ventana con todas las funciones luego de que el usuario logre conectarse al servidor.

6.3. Diagramas.

Los movimientos del robot se muestran a partir de los siguientes diagramas:

- A y D son los motores conectados al puerto OUTPUT_A y OUTPUT_D respectivamente.

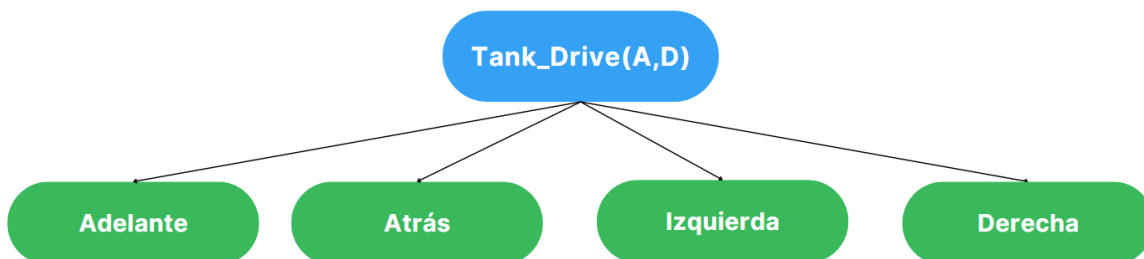


Ilustración 8: Diagrama movimientos.

- C es el motor conectado al puerto OUTPUT_C.

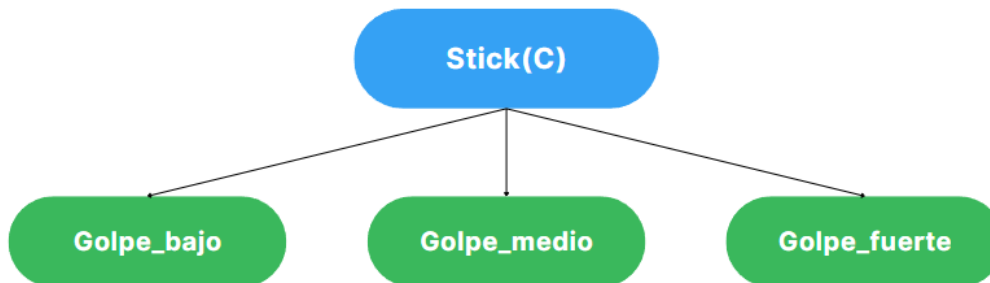


Ilustración 9: Diagrama movimiento.



7. Resultados

7.1. Estado actual del proyecto

- El diseño del robot ya se encuentra construido en su totalidad.
- Cuenta con funciones de movimiento definidas (Dirección y movimiento Péndulo)
- Control de este a través de una interfaz gráfica mediante Tkinter y mediante la creación del servidor
- Wiki del proyecto actualizada
- Carta Gantt actualizada
- Bitácoras entregadas

7.2. Problemas encontrados y solución propuesta

Problemas encontrados	Soluciones
Movimientos del robot	Se reinstaló el SO EV3Dev para limpiar la cantidad de archivos dentro del bloque, lo que provocaba el problema.
Bloque EV3 no detecta los motores	Se hicieron modificaciones al código para que fueran compatibles con nuestro diseño de robot.
El diseño del robot contaba con más de cuatro motores	Se rediseñó el robot para que ocupara dos motores grandes y uno mediano.

Tabla 6: Problemas encontrados y solución propuesta.



8. Pruebas

8.1. Descripción de las pruebas realizadas

- **Prueba de motores:** Prueba de funcionamiento de motores en la fase inicial del robot para ver si sus puertos funcionan de manera correcta.
- **Pruebas de movimiento:** Verificar que el robot avance, retroceda, gire a la derecha y a la izquierda correctamente.
- **Pruebas de golpeo:** Comprobar que el brazo del robot gire correctamente y golpee la pelota según la intensidad elegida por el usuario.

8.2. Resultado de las pruebas

- **Prueba de motores:** Al realizar la prueba el brick EV3 no detectaba los motores.
- **Pruebas de movimiento:** Luego de solucionar el problema con los motores se logra que el robot se mueva correctamente.
- **Pruebas de golpeo:** El movimiento del brazo del robot, golpea la pelota según la intensidad que se indique.



9. Conclusión

Siendo este el último informe de progreso, se puede declarar que el proyecto "Golffenheimer" ha sido completado de manera exitosa, con todos los objetivos que se plantearon en las etapas de planeación siendo cumplidas y alcanzadas.

La documentación (que incluye la wiki, las bitácoras, la carta Gantt y el manual de usuario) está completa y al día, al igual que el robot sobre el cual gira el proyecto. Todos los ajustes que quedaron pendientes del informe anterior han sido aplicados sin mayores contratiempos, lo que significa que nuestro robot es capaz de moverse y golpear una pelota pequeña, pudiendo controlar el nivel de impacto del golpe.

Además, como se propuso en el último informe, "Golffenheimer" es de fácil uso para cualquier tipo de usuario, incluyendo a quienes no tienen experiencia con este tipo de software. Recursos como la extensa wiki y el manual de usuario hacen aún más accesible este producto al usuario común.

El proyecto "Golffenheimer" ha cumplido todas las expectativas, y se declara finalizado.



10. Referencias

[1] *LEGO Mindstorm EV3. Versiones y kits, la guía definitiva.* (n.d.).

Juegos Robótica. <https://juegosrobotica.es/lego-mindstorm/>