**UNIVERSIDAD DE TARAPACÁ**

**FACULTAD DE INGENIERÍA**

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

Imagen que contiene Forma

Descripción generada automáticamente

Formulación de Proyecto

“Machine EV3 G”

Integrantes:

* Jean Pierre Duran
* Valentina Álvarez
* Cristian Sánchez
* Cesar Jiménez
* Sergio Rabanal

Docente: Humberto Urrutia

Modulo: Proyecto I – Semestre IV

Fecha – Inicio Documentación: 14 de Agosto del 2023

Fecha – Término Documentación: 30 de Noviembre del 2023

**Historial de cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Fecha*** | ***Versión*** | ***Descripción*** | ***Autores*** |
| 16/08/2023 | 1.0.1 | Planificación del proyecto “Machine EV3 G” | -Valentina Alvarez  -Cesar Jimenez  -Sergio Rabanal  -Jean Piere Durán  -Cristhian Sanchez |
| 18/08/2023 | 1.0.1 | Definición de roles | -Valentina Alvarez  -Cesar Jimenez |
| 18/08/2023 | 1.0.2 | Elaboración de informe; bitácora y carta Gantt. | -Cesar Jimenez  -Sergio Rabanal  -Cristhian Sanchez |
| 23/08/2023 | 1.0.3 | Construcción de la base del robot. | -Valentina Alvarez |
| 23/08/2023 | 1.0.4 | Instalación del sistema EV3 y desarrollo del código. | -Cristhian Sanchez  -Jean Piere Duran |
| 13/11/2023 | 1.0.5 | planificación del informe de avance | -Cristhian Sanches  -Jean Pierre Duran  -Cesar Jimenez -Sergio Rabanal |
| 19/11/2023 | 1.0.6 | revisión del informe de avance | -Cristhian Sanches  -Jean Pierre Duran  -Cesar Jimenez -Sergio Rabanal  Valentina Alvarez |

Tabla de contenido

[1.)Panorama General 5](#_heading=h.30j0zll)

[1.1) Introducción: 5](#_heading=h.1fob9te)

[1.2) Objetivo General: 6](#_heading=h.3znysh7)

[1.3) Objetivo Específicos 6](#_heading=h.2et92p0)

[1.4) Restricciones: 7](#_heading=h.tyjcwt)

[1.5) Entregables: 8](#_heading=h.3dy6vkm)

[2.Organización del Personal 9](#_heading=h.1t3h5sf)

[2.1) Descripción de roles: 9](#_heading=h.4d34og8)

[2.2) Medios de Comunicación:](#_heading=h.2s8eyo1) 11

[3.Planificación del proyecto 1](#_heading=h.17dp8vu)1

[3.1) Actividades: 1](#_heading=h.3rdcrjn)1

[3.2) Asignación de tiempo (Gantt): 1](#_heading=h.26in1rg)3

[3.3) Gestión de riesgos: 1](#_heading=h.lnxbz9)4

[4)Planificación de los recursos 1](#_heading=h.35nkun2)6

[4.1) Recursos (Hardware-Software): 1](#_heading=h.1ksv4uv)6

[4.2) Estimación de costos: 1](#_heading=h.44sinio)6

[5)Análisis y Diseño 1](#_heading=h.z337ya)9

[5.1) Especificación de Requerimientos funcionales y no funcionales. 1](#_heading=h.3j2qqm3)9

[5.2) Arquitectura](#_heading=h.1y810tw) 20

[5.3) Interfaz 2](#_heading=h.4i7ojhp)1

[6)Implementación 2](#_heading=h.2xcytpi)2

[6.1) Fundamentos del movimiento parabólico. 2](#_heading=h.1ci93xb)2

[6.2) Descripción de los programas 2](#_heading=h.3whwml4)4

[6.3) Diagramas 3](#_heading=h.2bn6wsx)9

[7)Resultados 38](#_heading=h.qsh70q)

[7.1)Estado actual del proyecto.](#_heading=h.3as4poj) 40

[7.2) Problemas encontrados y soluciones propuestas](#_heading=h.1pxezwc) 40

[8)Conclusión 4](#_heading=h.49x2ik5)2

[9)Referencias 4](#_heading=h.qpeemrh1wscq)3

## 1.)Panorama General

### 1.1) Introducción:

Desde hace muchos años, se soñó con crear robots humanoides o máquinas que fueran capaces de realizar todas las tareas que nosotros los humanos no podemos realizar por las limitaciones físicas, psicológicas o intelectuales que tiene una persona, sin embargo, los científicos de la época tenían pronósticos muy acertados acerca del futuro prometedor que nos podría brindar la tecnología y estos seres creados por la raza humana.

Hoy en día existen diferentes inventos tecnológicos que cubren diferentes tipos de necesidades con relación a la comunicación a distancia, tareas domésticas, transportes, etc.…

Pero desde entonces se creía que esto de crear robots era solo para científicos y personas con el conocimiento necesario para realizarlo, entonces, en el año 1998 se creó una forma más lúdica de fomentar este mundo de la tecnología con el “Lego Mindstorms” es una línea de robótica para niños que utiliza conocimientos básicos de robótica.

Como grupo, utilizaremos este medio “Lego Mindstorms” elaborar un proyecto en base a una temática específica este será programado con un lenguaje de programación, y cada una de las etapas del desarrollo del proyecto serán registradas en diferentes fuentes de información



### 1.2) Objetivo General:

Diseño y construcción de un robot mediante el uso del set Lego Mindstorms EV3 con la capacidad de movilizarse y lanzar una pelota de forma precisa y controlada a través de una interfaz gráfica. El programa para controlar el robot tendrá un desarrollo en Python, y la interfaz gráfica permitirá al usuario enlazar cada función del robot a botones, emulando un control remoto para dirigir la dirección y ajustar la fuerza del disparo del objetivo.

### 1.3) Objetivo Específicos

1)Diseñar un robot capaz de moverse en todas direcciones, que incluya un mecanismo para golpear una pelota con precisión.  
2) Controlar el movimiento parabólico presente en el lanzamiento de una pelota, realizando los cálculos requeridos aplicando conocimientos previos.  
3) Diseñar y programar una interfaz gráfica intuitiva junto a sus respectivas funciones para controlar el movimiento del robot, la potencia y el golpeo del mecanismo.  
4) Crear un servidor SSH para conectar el robot a la interfaz gráfica y las funciones que controlan el movimiento y las acciones a realizar por el robot.

### 1.4) Restricciones:

Hace referencia a las limitaciones que se puedan presentar a cabo de llevar a cabo el proyecto:

* **Temática**: el proyecto debió ser realizado basándonos en un modelo específico de robot con relación al golf, cosa que podría realizarse o no todo depende del dominio que tenga el programador y diseñador sobre el tema.
* **Duración**: Este proyecto debió ser llevado a cabo en un lapso de tiempo determinado, entonces para dicha fecha límite el proyecto debe estar concretado y finiquitado.
* **Piezas**: dado que el kit “Lego Mindstorms EV3” tiene piezas limitadas debimos solicitar más piezas para la construcción de nuestro prototipo de robot.
* **Interconexión**: uno de los principales problemas que se tuvo fue con el manejo del robot fue el no poder establecer la conexión entre el ordenador y la caja EV3 en el tiempo estimado en la semana predestinada.
* **Set**: El proyecto se llevó a cabo con el estudio de otros modelos de robot, ya que en el grupo no se tenía conocimiento base sobre el uso del Kit “Lego Mindstorms EV3”.
* **Interfaz**: Este prototipo debe tener una interfaz gráfica diseñada con el lenguaje de programación Python que pueda ser capaz de dar instrucción de forma remota a la caja EV3 y posteriormente el robot pueda realizarlas.
* **Lenguaje Programación:** Uno de los inconvenientes que tuvimos al inicio fue el desconocimiento de la librería de lego mindstorms con el lenguaje de programación.
* **Sistema Operativo:** Este proyecto se tiene que realizar con un sistema operativo específico, este es Linux Ubuntu 23.04 y es un sistema de código abierto. Se utilizó para optimizar la interconexión entre el dispositivo y la caja EV3 del kit.

### 1.5) Entregables:

 - Informe: Presentación general del proyecto, este formato se basa en el “todo”, es uno de los más completos, en cuanto a información, ya que se registra el proyecto consolidado. Se enfoca en cada uno de los requerimientos de forma minuciosa para llevar a cabo el proyecto, tales como: Estimación de Costos, Software y Hardware requeridos, detalles del funcionamiento del Robot,

- Bitácoras: Es un formato en el cual se presentan todas las actividades diarias realizadas en el transcurso del semestre para el desarrollo del Robot “Machine EV3 G” dando descripción breve sobre lo realizado en la semana.

- Manual de Usuario: Es un Instructivo que indica cómo se puede usar de forma correcta el robot y también nos indica que precauciones debemos tener en cuenta al uso del mismo. Este Formato, da instrucciones sobre el armado de cada una de las piezas para darle forma al robot, y también viene con imágenes originales sobre la construcción del Robot.

- Presentación: Diagrama visual que se usará para presentar el proyecto, este “formato” es más dinámico ya que no se encarga solo de registrar información para el lector, sino que también de exponerla a un público objetivo con tal de dar ideas sobre un proyecto de robótica u informar sobre el estado del producto.

- Wikipedia: Es donde se entrega un resumen o descripción breve sobre el asunto, en este caso del proyecto. Este formato contiene información del informe, pero se entrega solo lo más relevante para captar la atención del público objetivo.

## 2.Organización del Personal

### 2.1) Descripción de roles:

2023-08-19:

|  |  |  |
| --- | --- | --- |
| Ocupación | Descripción | Responsables |
| Jefe de Grupo | Dirige el proyecto del grupo y se encarga de ver el avance del proyecto. | Jean Piere Duran |
| Programador | Programan y desarrollan el software del proyecto | CristhianSánchez |
| Ensamblador | Construyen y manipulan el hardware del proyecto. | Valentina Álvarez |
| Documentador | Encargado de realizar y supervisar la elaboración de los informes, bitácoras, manual, etc. | Cesar Jiménez  Sergio Rabanal |

2023-10-22 (actualización de roles):

|  |  |  |
| --- | --- | --- |
| Ocupación | Descripción | Responsables |
| Jefe de Grupo | Dirige el proyecto del grupo y se encarga de ver el avance del proyecto. | Jean Piere Duran |
| Programador | Programan y desarrollan el software del proyecto | Cesar Jimenez  Valentina Álvarez |
| Ensamblador | Construyen y manipulan el hardware del proyecto. | Sergio Rabanal |
| Documentador | Encargado de realizar y supervisar la elaboración de los informes, bitácoras, manual, etc. | CristhianSánchez  Jean Piere Duran |

### 

### 

### 

### 

### 2.2) Medios de Comunicación:

WhatsApp: utilizamos Para establecer los horarios de las reuniones semanales creando un grupo, este es el principal medio de comunicación que utilizamos a lo largo del desarrollo de proyecto.

Discord: es una aplicación de comunicación versátil que se utiliza para reuniones y conversaciones en grupo, en nuestro caso, especialmente fuera del horario académico o en situaciones de urgencia.

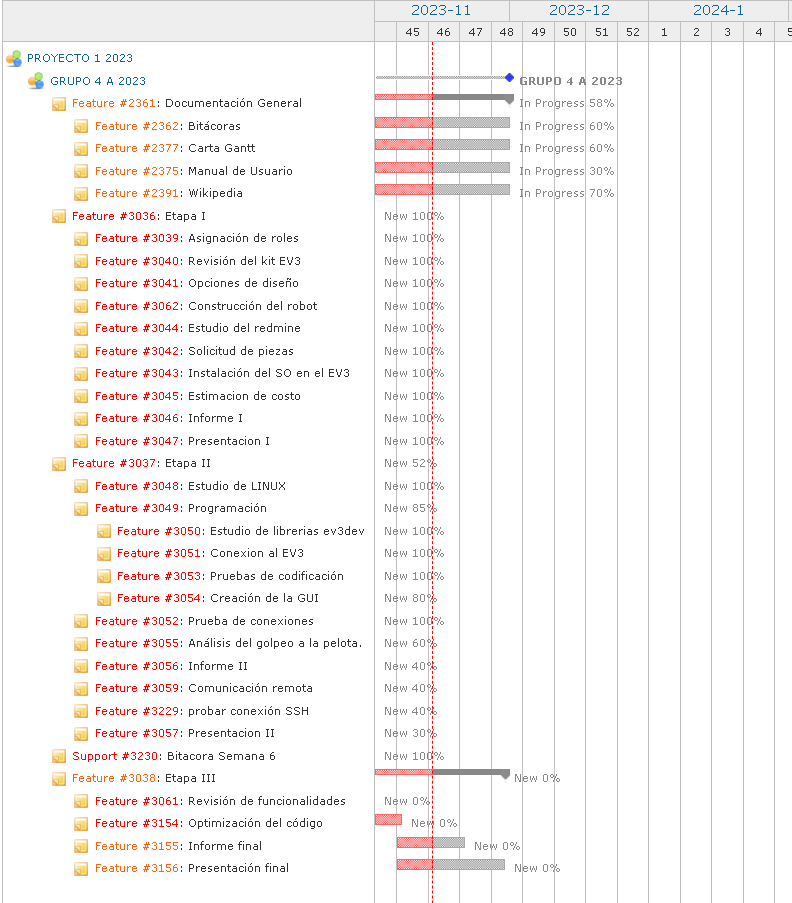
Redmine: es una plataforma de la Universidad de Tarapacá, cumple la función como medio de “comunicación” no apto para entablar una conversación, sino que más para informar al docente encargado sobre los avances realizados en cuanto al proyecto.

## 3.Planificación del proyecto

### 3.1) Actividades:

|  |  |  |
| --- | --- | --- |
| Actividades | Descripción | Asignaciones |
| Bitácora Semanal | Es un registro semanal de todas las actividades realizadas. | César Jiménez |
| Carta Gantt | Es un tipo de registro cronológico que organiza el tiempo y fechas estimadas por actividad. | Cristhian Sánchez |
| Manual de Usuario | Es un instructivo para orientar al usuario sobre la construcción del robot. | Valentina Álvarez |
| Diseño General | Se encargará de proponer mejoras estéticas para el proyecto. | Valentina Álvarez  Sergio Rabanal |
| Wikipedia | Es una presentación breve del proyecto por la plataforma redmine | César Jiménez |
| Prototipo | Será el encargado de estudiar las capacidades y limitantes que tiene el kit “LEGO MINDSTORMS EV3”. | Valentina Álvarez |
| Informe Final | Se dará registro detallado de cada una de las etapas, pros y contras del proyecto en sí. | Sergio Rabanal  César Jiménez |
| Redmine | Plataforma web de la Universidad de Tarapacá para dar registro al Proyecto. | César Jiménez  Sergio Rabanal  Cristhian Sánchez |
| Código | Son los encargados de estudiar e implementar el código y la interfaz gráfica, teniendo conocimiento del lenguaje de programación y las librerías del kit en cuestión. | Cristhian Sánchez  Jean Piere Duran |
| Contabilidad | Se encargarán de realizar los cálculos matemáticos. Que se requieran en cada área. | Sergio Rabanal  César Jiménez  Cristhian Sánchez |
| Presentaciones | Se encargarán de exponer el prototipo de robot con el kit “LEGO MINDSTORMS EV3” | Sergio Rabanal  César Jiménez  Valentina Álvarez  Jean Piere Duran  Cristhian Sánchez |

### 3.2) Asignación de tiempo (Gantt):



### 3.3) Gestión de riesgos:

Tabla con posibles riesgos que se tienen a consideración al realizar el proyecto. El nivel de impacto se ordena de la siguiente forma:

1. Catastrófico.

2. Crítico.

3. Marginal.

4. Despreciable.

|  |  |  |  |
| --- | --- | --- | --- |
| **Riesgo** | **Probabilidad**  **De ocurrencia** | **Nivel de impacto** | **Acción remedial** |
| **Escasez de Piezas** | **10%** | **4** | Comprar más piezas |
| **Mal estimación de tiempo** | **5%** | **3** | Dividir aún más las tareas para desarrollarlas en los tiempos determinados |
| **Limitaciones de conocimiento** | **40%** | **3** | Investigar más sobre el tema, sino designar a una persona que conozca sobre el tema |
| **Ausencia de personal** | **20%** | **1** | Se realiza un relevo temporal para la tarea específica. |
| **Ausencia en las reuniones** | **10%** | **2** | Realizar reuniones por medios de comunicación a distancia |
| **Poco compromiso del personal** | **15%** | **2** | Hallar la causa del problema con la persona y replantear la estrategia |
| **Perdida de avance del proyecto** | **5%** | **4** | Tener respaldos de los documentos en diferentes plataformas |
| **Problemas con la conexión** | **8%** | **2** | Crear un enlace de conexión entre los dispositivos con una red inalámbrica por medio de un celular |
| **Reconstrucción total o parcial del**  **robot** | **15%** | **4** | Buscar ideas novedosas e implementarlas en el robot |
| **Dañar el equipo** | **10%** | **3** | Buscar alternativas para poder realizar la misma tarea. |

## 

## 4)Planificación de los recursos

### 4.1) Recursos (Hardware-Software):

Se realizará una explicación breve de todos los implementos utilizados en el desarrollo del proyecto, en cuanto a:

Hardware:

* Computador: Dirige el proyecto del grupo y se encarga de ver el avance del proyecto.
* Piezas lego (Mindstorms EV3): Programan y desarrollan el software del proyecto.
* Controlador Maestro: Documenta y registra el avance del proyecto.
* Motores y sensores: Construyen y manipulan el hardware del proyecto.

Software:

* Microsoft office: Programa para la realización de bitácoras e informes en PPT y la carta Gantt.
* Redmine: Página dada por el profesor para subir los avances e informes del proyecto de forma semanal.

### 4.2) Estimación de costos:

Es una estimación por los 4 meses trabajados en el proyecto.

Costo por Gestión:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Encargado | Persona (s) | Valor hora (por trabajador) | Horas mensuales | Sueldo mensual (por trabajador) | Total por Semestre  (4 meses) |
| Jefe de  grupo | 1 | $9.000 | 38,5 | $346.500 | $1.386.000 |
| Programador | 1 | $8.500 | 38,5 | $327.250 | $1.309.000 |
| Ensamblador | 1 | $8.000 | 38,5 | $308.000 | $1.232.000 |
| Documentador(es) | 2 | $7.000 | 38,5 | $269.500  ($539.000) | $1.078.000  ($2.156.000) |
| Totales | | | | $1.520.750 | $6.083.000 |

Costo Software o Servicio:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Servicio o Producto | Descripción | Meses | Costo Mensual | Costo Semestral (4 meses) |
| Internet | El acceso a internet se realizó mediante una red local desde el celular. | 4 | $16.990 | $67.960 |
| Microsoft Word | Se pagó la licencia de Microsoft Office para documentar el Proyecto. | 4 | $7.500 | $60.000 c/u |
| Transporte | Se realizaron gastos de movilización (como grupo). | 4 | $160.000 | $640.000 |
| Total | | | | $ 767.960 |

Costo Hardware o Material:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Producto | Descripción | Cantidad | Costo Unitario | Costo Total | |
|  | Arriendo notebook Lenovo por 4 meses | 3 | 75.000 | 900.000 | |
| Kit Lego | Set de construcción LEGO Mindstorms EV3 robótica | 1 | $1.200.000 | $1.200.000 | |
| Memoria | Micro SD (8 GB)  Kingston | 1 | $10.000 | $10.000 | |
| Router | ZTE - BAM Mifi MF920U | 1 | 49.990 | $49.990 | |
| Set de expansión | Set de Expansión LEGO Mindstorm EV3 | 1 | $250.000 | $250.000 | |
| Antena WIFI | Tarjeta de Red USB WiFi Tp-Link I330WR725N | 1 | $12.000 | 12.000 | |
|  | | | Total | | 2.421.990 |

|  |  |
| --- | --- |
| Costos Proyecto | Costos Totales |
| Costo Software | $ 767.960 |
| Costo Gestión | $ 6.083.000 |
| Costo Hardware | $ 2.421.990 |
| Total | $ 9.272.950 |

# 

## 5)Análisis y Diseño

### 5.1) Especificación de Requerimientos funcionales y no funcionales:

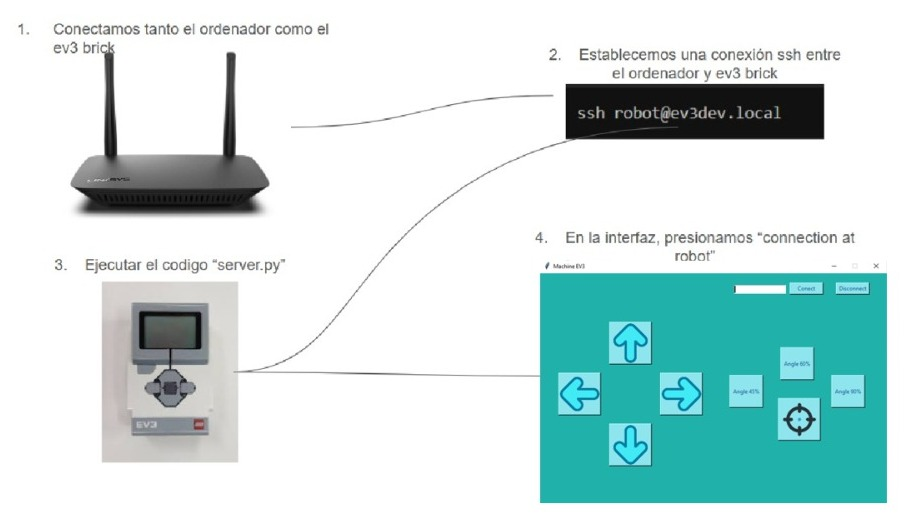
**Requerimientos funcionales:**

* El robot tiene que ser capaz de jugar al golf mediante una palanca mecánica que lance la pelota, por lo cual requiere de funciones para interactuar con la pelota.
* El robot debe poder ser capaz de moverse en distintas direcciones.
* Se requiere de un servidor que sea capaz de comunicar información entre la interfaz y las funciones del robot.

**Requerimientos no funcionales:**

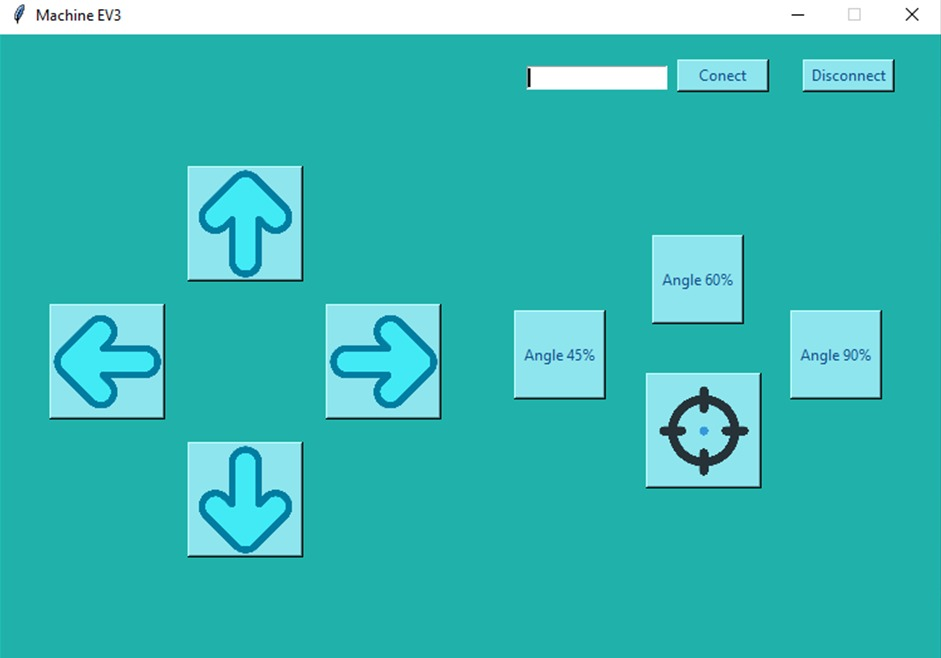
* Las construcciones del código junto con sus respectivos algoritmos serán hechos desde el lenguaje de programación Python por las comodidades que este presenta.
* La interfaz gráfica ha de ser sencilla para el manejo del usuario, de modo que pueda realizar movimientos de forma fluida y poder ajustar el potencial de disparo de la pelota.
* El entorno de desarrollo y diseño en el aspecto de la programación debe ser llevado en el sistema operativo Linux, debido a las comodidades que este presenta en el tema de sus librerías.
* El robot tiene que tener una ejecución fluida de sus movimientos o acciones.

### 5.2) Arquitectura:



* El ladrillo EV3 y el notebook deben estar conectados a la misma red WIFI.
* La interfaz gráfica permite controlar al robot en todos sus apartados disponibles mediante una conexión remota.
* La conexión remota se lleva a cabo mediante la ejecución del programa Server.py, enlazando las funciones del programa con el robot.
* El robot ejecutará las acciones indicadas por el usuario que opera en el notebook mediante la interfaz gráfica.

### 5.3) Interfaz:



El usuario tendrá acceso a una pantalla principal con una interfaz, similar a la que se muestra en la imagen. En esta pantalla, encontrará las funciones y opciones principales.

* El botón "Connection at robot" se encarga de enlazar el cliente con el servidor a través de la dirección IP del EV3.
* El botón "Disconnect" se encarga de desenlazar la conexión remota.
* El botón "Shoot" mueve el brazo mecánico a una potencia del 50%.
* Flechas para controlar el movimiento del robot: hacia adelante, hacia atrás, giro a la izquierda y giro a la derecha.
* Los botones de potencias en la parte superior indican el % de fuerza a aplicar en el lanzamiento de la pelota de golf.

## 

## 

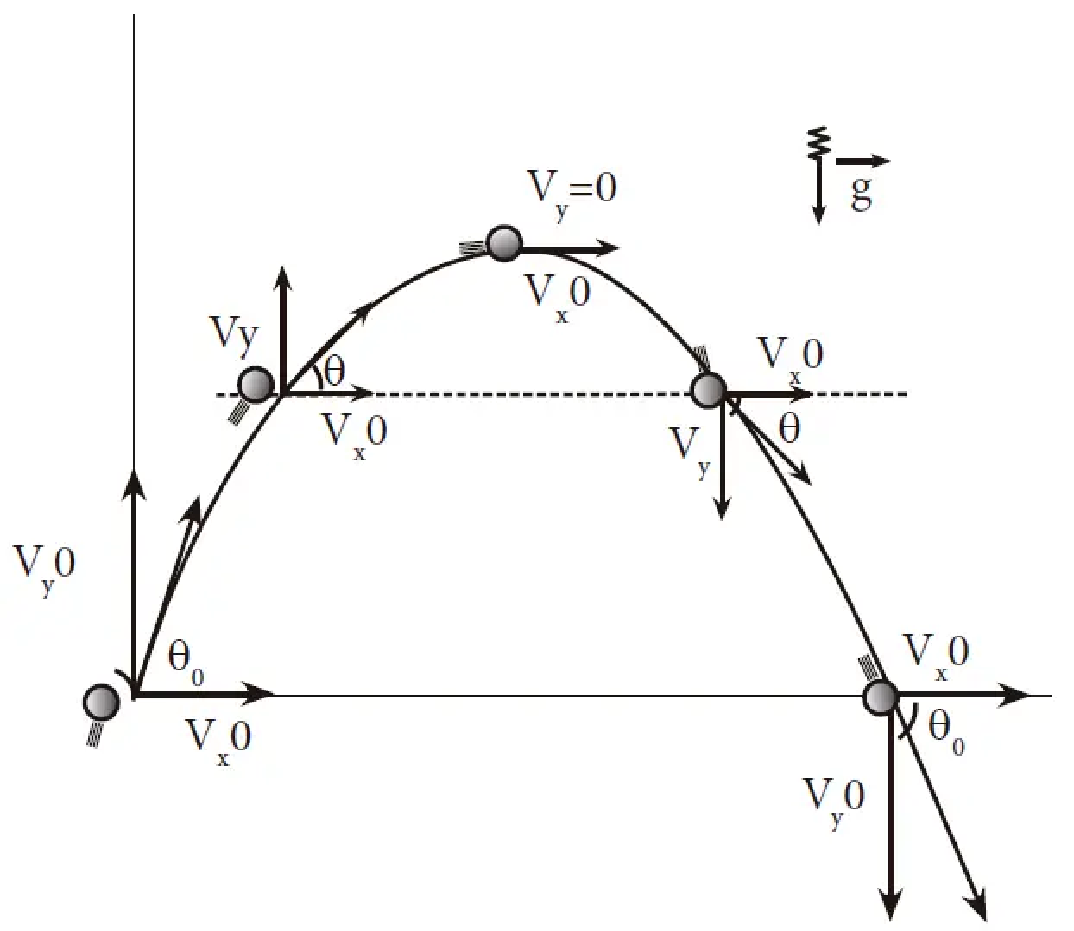
## 

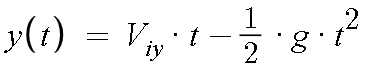
## 6)Implementación

### 6.1) Fundamentos del movimiento parabólico:

Para realizar un tiro preciso, analizamos los principios físicos involucrados en un parabólico. Esto nos permite predecir con precisión el comportamiento de la pelota. Es esencial considerar dos tipos de movimientos para calcular el tiempo de vuelo de la pelota lanzada por el brazo mecánico.

En términos más detallados, estas fórmulas permitirán determinar con precisión cuánto tiempo la pelota estará en el aire antes de tocar el suelo. Esto implica considerar la influencia de la gravedad en el movimiento de la pelota y calcular el intervalo de tiempo en el cual la pelota experimenta una trayectoria parabólica descendente. Este enfoque es crucial para comprender y prever el comportamiento del proyectil lanzado, lo que puede ser esencial para optimizar su rendimiento o realizar ajustes necesarios en el diseño del brazo mecánico.





En base a los siguientes datos X**f** = 0.8m ; T = 0.5s

X(T) = V**ix\***T Y(T) = V**iy**\*T - (1/2)\*G\*T**2** Teniendo V**ix** y V**ix**  encontramos el

0.8 = V**ix**\*(0.5) 0 = V**iy**\*(0.5) - 4.9\*(0.5)**2** módulo de la velocidad inicial.

V**ix** = 1.6 m/s V**iy** = 2.45 m/s √((1.6)**2**+(2.45)**2**) = 2.93 m/s

Con estos datos podemos obtener la altura máxima y un aproximado bastante cercano al ángulo de lanzamiento de la pelota.

Θ = arctan(2.45/1.6) V**y**(T) = V**i**\*sin(Θ) - 9.8\*T Y(T) = V**iy**\*T - (1/2)\*G\*T**2**

Θ = 56.85° 0 = 2.45 - 9.8\*T Y(0.25) = (2.45)(0.25) - (4.9)(0.25)

T = 0.25 s H**max** = 0.30625 m

Estos cálculos proporcionan las componentes de la velocidad inicial v0x y v0y y la altura máxima “H” para el lanzamiento parabólico con los datos proporcionados.

### 6.2) Descripción de los programas:

* **Funciones:**

from ev3dev.ev3 import Sound

from ev3dev2.motor import LargeMotor,OUTPUT\_A, OUTPUT\_B, OUTPUT\_C, OUTPUT\_D, MoveTank

#Agregar Funciones del ROBOT

rueda\_left = LargeMotor(OUTPUT\_C)

rueda\_right = LargeMotor(OUTPUT\_B)

motor\_A = LargeMotor(OUTPUT\_A)

motor\_D = LargeMotor(OUTPUT\_D)

def Mov\_Right():

    motor\_A.run\_forever(speed\_sp=-80)

def Mov\_Left():

    motor\_A.run\_forever(speed\_sp=80)

def Mov\_Down():

    rueda\_left.run\_forever(speed\_sp=500)

    rueda\_right.run\_forever(speed\_sp=500)

def Mov\_Up():

    rueda\_left.run\_forever(speed\_sp=-500)

    rueda\_right.run\_forever(speed\_sp=-500)

def Stop\_Motor():

    rueda\_left.stop()

    rueda\_right.stop()

    motor\_A.stop()

def Angle45():

    motor\_D.on\_for\_degrees(speed=10, degrees = 45)

    motor\_D.on\_for\_degrees(speed=100, degrees = -105)

    motor\_D.on\_for\_degrees(speed=10, degrees = 60)

    motor\_D.off()

def Angle60():

    motor\_D.on\_for\_degrees(speed=10, degrees = 60)

    motor\_D.on\_for\_degrees(speed=100, degrees = -120)

    motor\_D.on\_for\_degrees(speed=10, degrees = 60)

    motor\_D.off()

def Angle90():

    motor\_D.on\_for\_degrees(speed=10, degrees = 90)

    motor\_D.on\_for\_degrees(speed=100, degrees = -150)

    motor\_D.on\_for\_degrees(speed=10, degrees = 60)

    motor\_D.off()

* **Server:**

import socket

from Funciones import \*

# Crear un socket del servreceptor

server = socket.socket()

# Vincular el socket a una dirección IP y un puerto

server.bind(('',4999))

print("CONECTADO")

# Poner el socket en modo de escucha

server.listen(5)  # Permite hasta 5 conexiones pendientes

print("Esperando conexiones entrantes...")

# Aceptar una conexión entrante

socket\_client, address\_client = server.accept()

print("Conexion entrante desde 192.168.47.147")

while True:

    recept = socket\_client.recv(1).decode("utf-8")

    if (recept == 'w'): # arriba

        Mov\_Up()

    if (recept == 's'): # abajo

        Mov\_Down()

    if (recept == 'a'): # izquierda

        Mov\_Left()

    if (recept == 'd'): # derecha

        Mov\_Right()

    if (recept.isdigit()): # disparo por angulo

        if (recept == '1'):

            Angle45()

        elif (recept == '2'):

            Angle60()

        elif (recept == '3'):

            Angle90()

    if (recept == 'z'): # Cierre Ventana

        break

    if (recept == ' '): # Detener Movimiento

        Stop\_Motor()

* **Interfaz:**

import tkinter as tk

from tkinter import ttk

from tkinter import \*

from tkinter import messagebox

from tkinter import PhotoImage

import socket

conection = False

def stop(Event):

    try:

        server.send(bytes([ord(' ')]))

    except BrokenPipeError:

        messagebox.showwarning("Lo siento","Intentelo Nuevamente")

def Shoter():

    try:

        server.send(bytes([ord('r')]))

    except BrokenPipeError:

        messagebox.showwarning("Lo siento","Intentelo Nuevamente")

def Up():

    try:

        server.send(bytes([ord('w')]))

    except BrokenPipeError:

        messagebox.showwarning("Lo siento","Intentelo Nuevamente")

def Down():

    try:

        server.send(bytes([ord('s')]))

    except BrokenPipeError:

        messagebox.showwarning("Lo siento","Intentelo Nuevamente")

def Left():

    try:

        server.send(bytes([ord('a')]))

    except BrokenPipeError:

        messagebox.showwarning("Lo siento","Intentelo Nuevamente")

def Right():

    try:

        server.send(bytes([ord('d')]))

    except BrokenPipeError:

        messagebox.showwarning("Lo siento","Intentelo Nuevamente")

def Disconnect():

    if conection:

        server.send(bytes([ord('z')]))

        ventana.destroy()

    else:

        ventana.destroy()

def Power(Pow):

    if (Pow == 45):

        server.send(bytes([ord('1')]))

    elif (Pow == 60):

        server.send(bytes([ord('2')]))

    elif (Pow == 90):

        server.send(bytes([ord('2')]))

    else:

        server.send(bytes([ord('3')]))

def Validar():

    pts = 0

    numero = True

    for k in ip.get():

        if (k == '.'):

            pts += 1

        elif (k.isdigit()):

            continue

        else:

            numero = False

    if (pts == 3 and numero):

        conectar()

    else:

        messagebox.showwarning("Error","Ingrese una dirección valida")

    ip.set("")

def conectar():

    puerto = 4999

    direccion = ip.get()

    try:

        server.connect((direccion,puerto))

        messagebox.showinfo("Conectado","Se ha establecido la conexion exitosamente")

        conection = True

    except BrokenPipeError:

        messagebox.showwarning("Desconectado","Ingrese la dirección correcta")

        conection = False

    ip.set("")

# Ventana principal

ventana = tk.Tk()

ventana.geometry("750x500")

ventana.title("Machine EV3")

ventana.configure(bg="light sea green")

ventana.resizable(0,0)

ip = StringVar()

# Ingresar Imagenes

up = PhotoImage(file="up.png")

down = PhotoImage(file="down.png")

left = PhotoImage(file="left.png")

right = PhotoImage(file="right.png")

cross = PhotoImage(file="cross.png")

# Dimensiones de las Imagenes

up = up.subsample(6, 6)

down = down.subsample(6, 6)

left = left.subsample(6, 6)

right = right.subsample(6, 6)

cross = cross.subsample(6, 6)

# Iniciar Servidor

server = socket.socket()

# Boton Arriba

boton\_arriba = Button(None, repeatdelay=20,repeatinterval=20,text="UP", command=Up, background="cadetblue2", image=up)

boton\_arriba.place(x=150, y=105)

# Boton Derecha

boton\_derech = Button(None, repeatdelay=20,repeatinterval=20 ,text="RIGHT", command=Right, background="cadetblue2", image=right)

boton\_derech.place(x=260, y=215)

# Boton Izquierda

boton\_izq = Button(None, repeatdelay=20,repeatinterval=20, text="LEFT", command=Left, background="cadetblue2", image=left)

boton\_izq.place(x=40, y=215)

# Boton Abajo

boton\_abajo = Button(None, repeatdelay=20, repeatinterval=20,text="DOWN", command=Down, background="cadetblue2", image=down)

boton\_abajo.place(x=150, y=325)

# Boton Disparar

boton\_disparar = Button(text="SHOT", command=lambda:[Power(50)], background="cadetblue2", image=cross)

boton\_disparar.place(x=515, y=270)

# Boton Power de 30°

boton\_p30 = Button(text="Angle 45%", command=lambda:[Power(45)], background="cadetblue2",fg = "DodgerBlue4")

boton\_p30.config(width=7, height=4)

boton\_p30.place(x=410, y=220)

# Boton Power de 60°

boton\_p60 = Button(text="Angle 60%", command=lambda:[Power(60)], background="cadetblue2",fg = "DodgerBlue4")

boton\_p60.config(width=7, height=4)

boton\_p60.place(x=520, y=160)

# Boton potenca de 90°

boton\_p100 = Button(text="Angle 90%", command=lambda:[Power(90)], background="cadetblue2",fg = "DodgerBlue4")

boton\_p100.config(width=7, height=4)

boton\_p100.place(x=630, y=220)

# Desconectar Brick

boton\_exit = Button(text="Disconnect",command = Disconnect,background="cadetblue2",fg = "DodgerBlue4")

boton\_exit.config(width=9, height=1)

boton\_exit.place(x=620, y=20)

#Ingreso ip

entrada\_IP = Entry(ventana,textvariable = ip, width = 18)

entrada\_IP.focus\_set()

entrada\_IP.place(x = 320, y=25)

# Conexión

boton\_conectar = Button(text="Conect",background="cadetblue2",fg = "DodgerBlue4",command = Validar)

boton\_conectar.config(width=9, height=1)

boton\_conectar.place(x=490, y=20)

# Detener Motores

boton\_arriba.bind('<ButtonRelease-1>',stop)

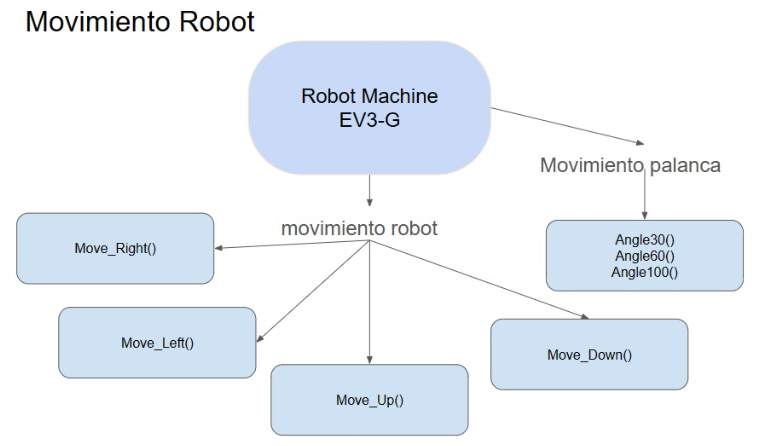
boton\_abajo.bind('<ButtonRelease-1>',stop)

boton\_izq.bind('<ButtonRelease-1>',stop)

boton\_derech.bind('<ButtonRelease-1>',stop)

ventana.mainloop()

### 6.3) Diagramas:

****

## 7)Resultados

### 7.1) Estado actual del proyecto:

Hasta el momento, hemos logrado resultados muy positivos en nuestro trabajo. En primer lugar, hemos establecido con éxito la conexión entre la interfaz en el ordenador y nuestro robot EV3. Ahora podemos enviar al robot instrucciones básicas de movimiento. Además, hemos confirmado que el diseño de disparo funciona correctamente.

* La conexión remota entre el cliente y el robot ha sido exitosamente establecida, esto mediante la creación de un servidor que utiliza la librería socket para establecer una conexión entre el robot y el notebook.
* Se cuenta con una interfaz gráfica, desarrollada con la librería Tkinter de Python, ha sido programada para permitir al cliente operar las funciones desde cualquier dispositivo compatible con Python. La interfaz es completamente funcional en la actualidad.
* El robot ahora puede desplazarse en diversas direcciones y ejecutar el lanzamiento de pelotas de golf de manera efectiva.
* La estructura del robot se encuentra cómodamente finalizada para ejecutar sus acciones básicas.

### 7.2) Problemas encontrados y soluciones propuestas:

|  |  |
| --- | --- |
| **Problemas encontrados** | **Soluciones** |
| Movimientos toscos y poco fluidos del robot a nivel mecánico. | Se integró un mecanismo mejorado a base de dos ruedas delanteras que giran en torno a cierta cantidad de grados. |
| Hacer que el brazo mecánico se mueva con precisión al ejecutar varios disparos. | Se utilizaron métodos relacionados al manejo del motor con grados para ejecutar 180 grados de forma constante. |
| Poco manejo en las nuevas librerías necesarias para el funcionamiento del robot. | Se investigaron los métodos de las librerías EV3 y socket para realizar funciones deseadas por parte de robot y establecer una conexión remota del usuario al robot. |
| Algoritmos poco eficientes para el movimiento del robot . | Se utilizaron técnicas de lectura y microlecturas en tiempo real para el movimiento perdurado si se deja pulsado la opción de movimiento. |

## 8)Conclusión

Durante el desarrollo del proyecto, hemos enfrentado diversos desafíos, particularmente en la fase final del diseño de nuestro robot EV3, encargado de lanzar pelotas de golf. En las etapas iniciales, evaluamos dos modelos con configuraciones estéticas considerablemente distintas antes de concretar la construcción del intrincado mecanismo de lanzamiento. Actualmente, contamos con un diseño robusto y programable que permite la emisión de pelotas en diversas direcciones, todo bajo el control preciso del usuario. Es trascendental subrayar que nuestro robot EV3 no solo exhibe la capacidad de lanzar pelotas de golf, sino que también incorpora la facultad de ajustar el ángulo de inclinación de los lanzamientos, característica que aporta versatilidad y precisión a su funcionalidad. Este logro ha sido posible gracias a la aplicación de conocimientos avanzados en el lenguaje de programación Python, adquiridos y perfeccionados tanto en ambientes académicos como domésticos.

En el ámbito de los desafíos, a pesar de las dificultades individuales y grupales, hemos cultivado la habilidad para superar cada obstáculo con resiliencia y creatividad. Este proceso no solo representó una oportunidad para perfeccionar habilidades técnicas, sino también para fomentar un espíritu de colaboración y respeto en nuestro equipo. A medida que enfrentamos y resolvemos problemas, hemos experimentado un crecimiento sustancial en nuestra capacidad para organizar tareas, una competencia que indudablemente se traducirá en un activo valioso para proyectos futuros, donde la coordinación efectiva y la habilidad para abordar desafíos se erigen como elementos cruciales para el éxito.

## 

## 9)Referencias

Documentación de la librería ev3:

<https://www.ev3dev.org/>

Modelo:

<https://www.youtube.com/watch?v=6Ck1uTM0J2A> y <https://www.youtube.com/watch?v=IKLf6zoXD6E>

Servidor:

<https://www.youtube.com/watch?v=nJYp3_X_p6c>

Métodos:

[https://sites.google.com/site/ev3basic/ev3-basic-programming/ev3-basic-manual/manualev3-basic](https://sites.google.com/site/ev3basic/ev3-basic-programming/ev3-basic-manual/manual-ev3-basic)