UNIVERSIDAD DE TARAPACÁ



**FACULTAD DE INGENIERÍA**

Departamento de Ingeniería en Computación e Informática



**Avance del Proyecto  
“Scorpion Machine Gun”**

**Autor(es): -Alexis Yucra**

**-Francisco Silva**

**-Leandro Molina**

**-Sebastián Cayupi**

**-Arturo Rodríguez**

**Asignatura: Proyecto 1**

**Profesor(es): Leonel Alarcón Bravo**

ARICA, 27 Octubre 2022

**Historial de Cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor(es)** |
| 18/08/2022 | 1.0 | Versión preliminar del formato | Alexis Yucra  Francisco Silva  Sebastián Cayupi  Leandro Molina  Arturo Rodríguez |
| 25/08/2022 | 1.1 | Revisión y Finalización del plan | Alexis Yucra  Francisco Silva  Sebastián Cayupi  Leandro Molina  Arturo Rodríguez |
| 08/09/2022 | 1.2 | Arreglos para la presentación | Alexis Yucra |
| 15/09/2022 | 1.3 | Arreglo para la formulación de proyectos | Alexis Yucra |
| 02/11/2022 | 1.4 | Actualización para la Presentación 2 | Todos\* |

Tabla de Contenidos

[1. Panorama General 4](#_Toc116266633)

[1.1 Introducción: 4](#_Toc116266634)

[1.2 Objetivos: 4](#_Toc116266635)

[1.2.1 Objetivo general 4](#_Toc116266636)

[1.2.2 Objetivos Específicos: 4](#_Toc116266637)

[1.3 Restricciones: 4](#_Toc116266638)

[1.4 Entregables: 5](#_Toc116266639)

[2. Organización del Personal 5](#_Toc116266640)

[2.1 Descripción de roles: 5](#_Toc116266641)

[2.2 Personal que cumplirá los roles: 6](#_Toc116266642)

[2.3 Mecanismos de Comunicación: 6](#_Toc116266643)

[3. Planificación del Proyecto 7](#_Toc116266644)

[3.1 Actividades (nombre, descripción, responsable, producto): 7](#_Toc116266645)

[3.2 Asignación de tiempo (carta Gantt Redmine): 8](#_Toc116266646)

[3.3 Gestión de Riesgos (ver plantilla para el Tratamiento de los Riesgos): 8](#_Toc116266647)

[4. Planificación de los recursos 10](#_Toc116266648)

[4.1 Recursos Hardware-Software requeridos 10](#_Toc116266649)

[4.2 Estimación de costos (hardware, software, recursos humanos) 10](#_Toc116266650)

[5. Análisis y Diseño 12](#_Toc116266651)

[5.1 Especificación de requerimientos 12](#_Toc116266652)

[5.2 Arquitectura 12](#_Toc116266653)

[5.3 Interfaz 12](#_Toc116266654)

[6. Implementación 12](#_Toc116266655)

[6.1 Fundamentos de Proyectiles 12](#_Toc116266656)

[6.2 Descripción de los programas 12](#_Toc116266657)

[6.3 Diagramas 12](#_Toc116266658)

[7. Resultados 12](#_Toc116266659)

[7.1 Estado actual del Proyecto 12](#_Toc116266660)

[7.2 Problemas Encontrados y solución Propuesta 12](#_Toc116266661)

[8. Conclusión 12](#_Toc116266662)

[9. Referencia 12](#_Toc116266663)

# Panorama General

## Introducción:

En el presente informe se buscará poner en antecedentes acerca de cómo se trabajará a lo largo del proyecto, este consiste en la diseño, construcción y programación de un robot hecho con piezas del kit lego Mindstorms EV3, el cual como objetivo principal deberá ser capaz de disparar ligas. El kit de lego viene con un minicomputador denominado Brick que servirá como cerebro del robot, a través de este se ejecutarán los programas encargados de las acciones del robot.

## Objetivos:

### Objetivo general:

Construir un robot con el kit lego Mindstorms EV3 que sea capaz de realizar distintos movimiento y capaz de disparar ligas, además de desarrollar una interfaz capaz de controlar los movimientos y acciones del robot.

### Objetivos Específicos:

Los objetivos específicos que han surgido en la creación del proyecto son los siguientes:

* Diseñar y construir un robot capaz de moverse y disparar ligas.
* Programar código Python capaz de controlar las acciones del robot.
* Diseñar interfaz para controlar los movimientos y acciones del robot.
* Entrega del producto final.

## Restricciones:

A medida que íbamos avanzando en la construcción del robot y su configuración , Tuvimos las siguientes complicaciones:

* Falta de piezas del robot .
* Falta de experiencia a la hora de crear su algoritmo e interfaz.
* Programar en Python.

Las cuales fueron rápidamente resueltas ya que como dejamos a un encargado de ver las cosas faltantes del robot lo solicitamos y nos respondían rápidamente dándonos solución , Con la falta de experiencia a la hora de crear algoritmo la solución que le dimos fue juntarnos investigar más de cómo será programado, siguiendo ejemplos ya armados que salían en la página que se mostraba en el Manual del kit del robot.

## Entregables:

Durante la realización del proyecto habrá una serie de documentos que entregar con el motivo de actualizar la información con respecto a los avances y organización del proyecto.

Informe de Formulación del Proyecto

* Presentación de Formulación del Proyecto
* Informe de Avance del Proyecto
* Presentación de Avance del Proyecto
* Informe Final del Proyecto
* Manual de usuario
* Presentación Final del Proyecto
* Producto Final

# Organización del Personal

## Descripción de roles:

Durante el desarrollo del proyecto se necesito que los integrantes del grupo cumplan con cierto rol, con el propósito de tener una mejor administración en los tiempos de trabajo y comunicación obteniendo como resultado el cumplimiento de las tareas propuestas semanalmente respetando su tiempo de realización, Los roles que se seleccionaron son los siguientes:

* Programador: Es el que se encargara en la elaboración del algoritmo y interfaz para poder maniobrar al robot
* Líder de grupo: esta al pendiente de todo los roles y mira el progreso del proyecto
* Ensamblador: Encargado de armar el robot y ver su funcionamiento.
* Documentador: encargo de tener la bitácora y el informe al día realizar fotos videos de los avances del proyecto .
* Diseñador: encargado de seleccionar el modelo que se basara el robot del proyecto.

## Personal que cumplirá los roles:

|  |  |  |
| --- | --- | --- |
| **Roles** | **Encargado** | **Descripción** |
| Jefe de proyecto | Arturo Rodríguez | Representante del equipo de trabajo, toda decisión debe ser aprobada por el líder. |
| Ensamblador | Francisco Silva | Encargado de armar el modelo seleccionado por el grupo |
| Programador | Leandro Molina y Sebastián Cayupi | Desarrollar e implementar el algoritmo en Python , con el propósito de ejecutar todas las acciones solicitadas |
| Documentador | Alexis Yucra | Encargado de realizar los informes ,presentaciones ,bitácoras ,videos manual de usuario y wiki del proyecto |
| Diseñador | Alexis Yucra, Francisco Silva, Leandro Molina, Sebastián Cayupi y Arturo Rodríguez | Encargado de seleccionar el diseño del robot |

## Mecanismos de Comunicación:

Los medios de comunicación que se utilizaran para la elaboración del proyecto serán los siguientes:

* Discord (Software Libre) : este software se utilizará para realizar reuniones externas de las clases presenciales para la organización del proyecto.
* WhatsApp (Software Libre) : esta aplicación es utilizada para avisar la coordinación de reuniones y para avisar si es que uno falta.
* Zoom (Software Libre) : este software se utilizará para en caso de emergencia por si se llega a caer discord .

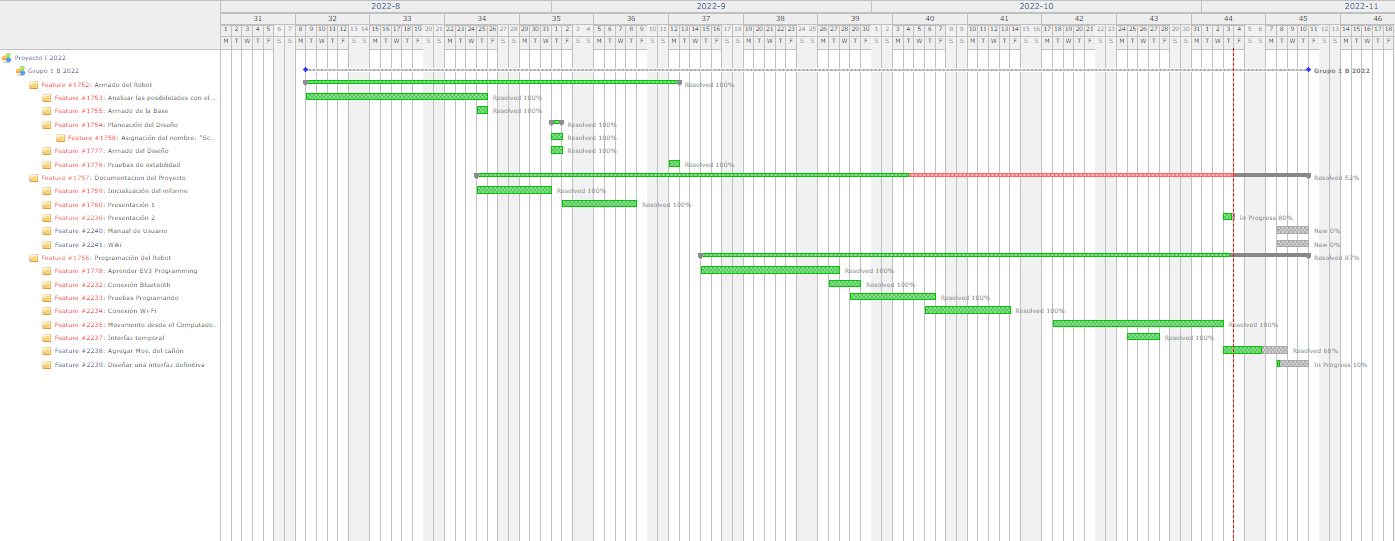
el objetivo de utilizar estos mecanismos de comunicación es más que todo para la organización que tendremos externas a las clases y justificar las ausencias de las reuniones, Donde también se comentan las ideas nuevas donde se vera si se aprueba o no por todos los integrantes .

# Planificación del Proyecto

## Actividades:



## Asignación de tiempo (carta Gantt Redmine):



Link: [Carta Gantt (imagen)](https://prnt.sc/jJ_1gEAzRB9X) o ver en [redmine](http://pomerape.uta.cl/redmine/projects/grupo-1-b-2022/issues/gantt?f%5B%5D=status_id&f%5B%5D=tracker_id&f%5B%5D=&gantt=1&month=8&months=4&op%5Bstatus_id%5D=o&op%5Btracker_id%5D=%3D&query%5Bdraw_progress_line%5D=0&query%5Bdraw_relations%5D=1&set_filter=1&utf8=✓&v%5Btracker_id%5D%5B%5D=2&year=2022&zoom=4)

## Gestión de Riesgos (ver plantilla para el Tratamiento de los Riesgos):

|  |  |  |  |
| --- | --- | --- | --- |
| **RIESGOS** | **PROBABILIDAD DE OCURRENCIA** | **NIVEL DE IMPACTO** | **ACCIÓN REMEDIAL** |
| Se necesitará cambiar las especificaciones del proyecto | 40% | 2 | Se buscará información para crear un proyecto de menor dificultad |
| Uno o más miembros del grupo se enferman | 20% | 3 | Se reorganizaran las tareas de los miembros afectados y así disminuir los riesgos |
| No se termina en el tiempo esperado | 30% | 2 | Se buscará una solución que permita terminar el proyecto los más antes posible |
| Uno o más miembros dejan el proyecto | 10% | 1 | Se reorganizaran las tareas |
| Falta de piezas/herramientas para el proyecto | 20% | 2 | Se buscará información de como terminar el proyecto buscando otro tipo de herramientas |
| Catástrofes Naturales | 40% | 1 | Se intentará resumir el proyecto lo más antes posible, sin poner en riesgo la vida y salud de los integrantes |
| Pérdida total de archivos o procesos | 10% | 1 | Recrear todo lo perdido lo más fiel posible, basándonos en nuestra memoria |
| Falta de información para realizar el proyecto | 50% | 3 | Informarse de forma completa antes de seguir con el proyecto |

**Factores de Riesgos:**

|  |  |
| --- | --- |
| **TIPO DE RIESGO** | **INDICADORES POTENCIALES** |
| Tecnología | No se cuenta con el software o herramientas necesarias |
| Salud | Algún integrante del grupo se contagia con COVID-19 |
| Organización | Ocurre algún contratiempo que complica la realización del proyecto |
| Estimación | Las expectativas no son realistas y se requerirá cambios en el proyecto |

# Planificación de los recursos

## Recursos Hardware-Software requeridos

los recursos que se utilizaron para la elaboración de este proyecto son los siguientes:

* Hardware: Lego Mindstorms EV3, Tarjeta SD 8 GB.
* Software: Lenguaje Python, Visual Studio Code .

## Estimación de costos (hardware, software, recursos humanos)

en este apartado se le mostrará detalladamente los recursos utilizados para llevar a cabo el proyecto:

Recursos utilizados:

1. Materiales:

|  |  |  |  |
| --- | --- | --- | --- |
| Recursos | Valor de recursos | Cantidad | Precio total |
| Notebook Msi Modern | $463.907 | 1 | $463.907 |
| micro SD | $5.900 | 1 | $5.900 |
| Kit Lego Ev3 | $1.233.999 | 1 | $1.233.999 |
| Notebook Asus Tuf Gaming | $649.990 | 1 | $649.990 |
| Adaptador USB inalámbrico TP-LINK TL-WN725N NANO | $9.900 | 1 | $9.900 |

Monto total de los materiales: $2.363.696

1. Estimación de costos (Trabajadores):
   * + - * Trabajo manual:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Función | Valor del trabajo | Cantidad de trabajadores | Horas trabajadas | Meses trabajados | Valor total |
| Programador | $7.500 | 2 | 19 | 4 | $1.140.000 |
| ensamblador | $7.000 | 1 | 4 | 4 | $112.000 |
| diseñador | $7.000 | 5 | 4 | 4 | $560.000 |

Monto total de trabajo manual: $1.812.000

* + - * + Trabajo por documento:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Función | Valor del trabajo | Cantidad de trabajadores | Cantidad de documentos | Valor total |
| Documentador | 7.000 | 1 | 8 | 56.000 |

Monto total de trabajo por documento: $56.000

Valor total del proyecto:

|  |  |  |  |
| --- | --- | --- | --- |
| trabajo por documento valor total | trabajo manual valor total | Materiales valor total | Monto total del proyecto |
| $56.000 | $1.812.000 | $2.363.696 | 4.231.696 |

# Análisis y Diseño

## Especificación de requerimientos

Especificación de requerimientos funcionales y no funcionales:

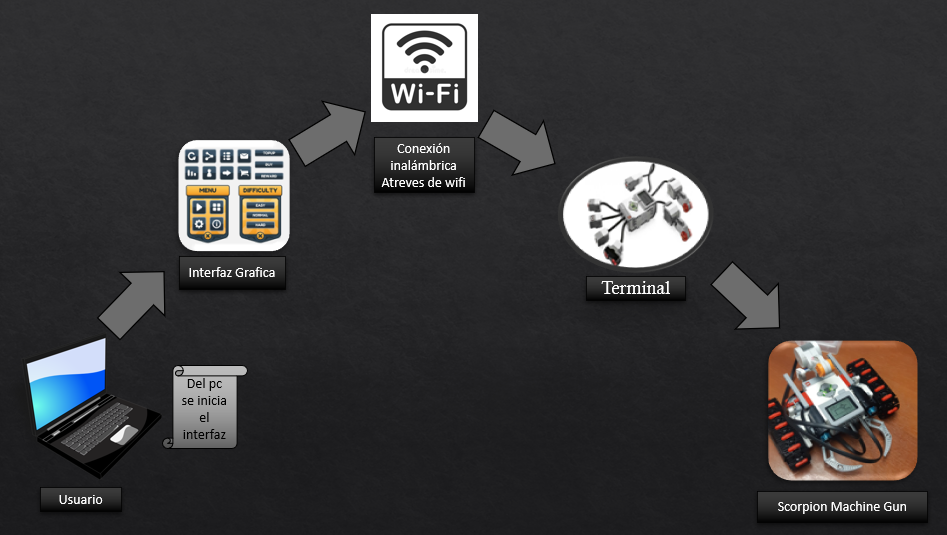
Funcionales:

* Construcción del robot conectado vía wifi.
* Que el usuario pueda decidir los movimientos del robot.
* Implementación de una interfaz grafica para la realización de los movimientos del robot donde se podrá controlar el robot , como por ejemplo el disparo del elástico del cañón que se pueda manipular de arriba para abajo o viceversa.

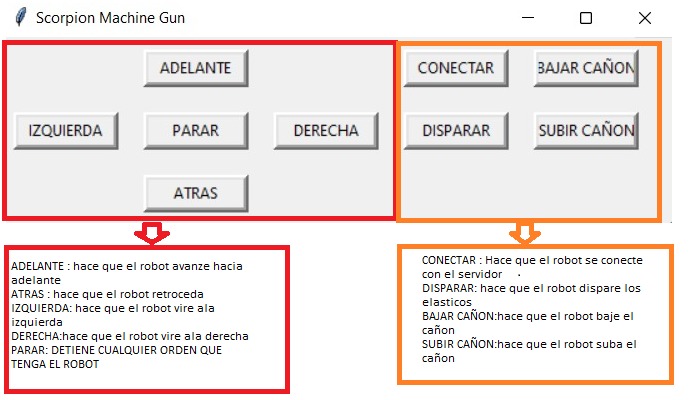
No Funcionales:

* La elaboración del algoritmo será a través de lenguaje de programación Python.
* Elaboración de un manual de usuario con el fin de explicar todas las funciones detalladamente.

## Arquitectura



## Interfaz



# Implementación

## Fundamentos de Proyectiles

El concepto físico que se presenta en nuestro proyecto a la hora de lanzar el elástico, se puede apreciar el lanzamiento de proyectil .

Que es?

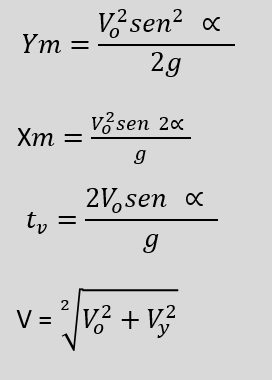
Un proyectil es cualquier objeto que se proyectará una vez que continúa en el movimiento por su propia inercia y es influenciado solamente por la fuerza hacia abajo de la gravedad.

Por definición, un proyectil tiene solamente una fuerza que actúa sobre él, esta es la fuerza de gravedad. Si hubiera alguna otra fuerza que actuara sobre un objeto, ese objeto no sería un proyectil. La gravedad actúa para influenciar el movimiento vertical del proyectil. El movimiento horizontal del proyectil es el resultado de la tendencia de cualquier objeto a permanecer en movimiento a velocidad constante.

Se denomina proyectil a cualquier objeto al que se le da una velocidad inicial y a continuación sigue una trayectoria determinada por la fuerza gravitacional que actúa sobre él y por la resistencia de la atmósfera. El camino seguido por un proyectil se denomina trayectoria.

El término proyectil se aplica por ejemplo a una bala disparada por un arma de fuego, a un cohete después de consumir su combustible, a un objeto lanzado desde un avión o en muchas actividades deportivas.

Se obtienen las ecuaciones cinemáticas del movimiento de un proyectil:



## Descripción de los programas

Interfaz:

from tkinter import \*

def enproceso():

print("falta")

ventppal = Tk()

ventppal.title('Scorpion Machine Gun')

ventppal.geometry('600x130')

Botonarriba = Button( text = "ARRIBA", width = 10, command = enproceso, bd= 4)

Botonarriba.grid(row = 1, column = 3, padx = 10, pady = 10)

botonizquierda = Button( text = "IZQUIERDA", width = 10, command = enproceso, bd= 4)

botonizquierda.grid(row = 2, column = 2, padx = 10, pady = 10)

botonatras = Button( text = "ABAJO", width = 10, command = enproceso, bd= 4)

botonatras.grid(row = 2, column = 3, padx = 10, pady = 10)

botonderecha = Button( text = "DERECHA", width = 10, command = enproceso, bd= 4)

botonderecha.grid(row = 2, column = 4, padx = 10, pady = 10)

botonbajarcañon = Button(text = "BAJAR CAÑON", width = 10, command = enproceso, bd= 4)

botonbajarcañon.grid(row = 2, column = 6, padx = (50, 0), pady = 10)

botonsubircañon = Button( text = "SUBIR CAÑON", width = 10, command = enproceso, bd= 4)

botonsubircañon.grid(row = 1, column = 6, padx = (50, 0), pady = 10)

botondisparar = Button(text = "DISPARAR", width = 10, command = enproceso, bd= 4)

botondisparar.grid(row = 2, column = 7, padx = (50, 0), pady = 10)

botondparar = Button(text = "Stop", width = 10, command = enproceso, bd= 4)

botondparar.grid(row = 1, column = 7, padx = (50, 0), pady = 10)

ventppal.mainloop()

funciones del robot(intento con RPYC actualizable):

#!/usr/bin/env python3

import rpyc

from rpyc.utils.server import ThreadedServer

from ev3dev.ev3 import \*

class MyService(rpyc.Service):

Motor\_izquierdo = LargeMotor('outB')

Motor\_derecho = LargeMotor('outC')

Motor\_cañon\_movimiento= LargeMotor('outA')

Motor\_cañon\_disparo= LargeMotor('outD')

def adelante():

Myservice.motor\_isquierdo.run\_forever(speed\_sp=500)

Myservice.motor\_derecho.run\_forever(speed\_sp=500)

def parar():

Myservice.motor\_isquierdo.run\_forever(speed\_sp=0)

Myservice.motor\_derecho.run\_forever(speed\_sp=0)

Myservice.Motor\_cañon\_movimiento.run\_forever(speed\_sp=0)

Myservice.Motor\_cañon\_disparo.run\_forever(speed\_sp=0)

def atras():

Myservice.motor\_isquierdo.run\_forever(speed\_sp=700)

Myservice.motor\_derecho.run\_forever(speed\_sp=700)

def isquierda():

Myservice.motor\_isquierdo.run\_forever(speed\_sp=400)

Myservice.motor\_derecho.run\_forever(speed\_sp=400)

def derecha():

Myservice.motor\_isquierdo.run\_forever(speed\_sp=400)

Myservice.motor\_derecho.run\_forever(speed\_sp=400)

def subircañon():

Myservice.Motor\_cañon\_movimiento.run\_forever(speed\_sp=300)

def bajarcañon():

Myservice.Motor\_cañon\_movimiento.run\_forever(speed\_sp=300)

def Disparar():

Myservice. Motor\_cañon\_disparo.run\_ forever (1000, 90000, wait=False)

wait(9000)

def exposed\_main(self, valor):

"""valor:char

esta fucnion es el main el cual recibe un valor"""

if valor== 'p':

MyService.parar()

if valor== 's':

MyService.atras()

if valor== 'a':

MyService.isquierda()

if valor== 'd':

MyService.derecha()

if valor== 'w':

MyService.adelante()

if valor== 't':

MyService.subircañon()

if valor== 'g':

MyService.bajarcañon()

else:

pass

if \_name\_=='\_main\_':

s=ThreadedServer(MyService,port=?????)

s.star()

help(MyService)

help(exposed\_main)

//cañon disparo

#!/usr/bin/env pybricks-micropython

from tkinter import \*

from pybricks.hubs import EV3Brick

from pybricks.hubs import MoveHub

from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,

InfraredSensor, UltrasonicSensor, GyroSensor)

from pybricks.parameters import Port, Stop, Direction, Button, Color

from pybricks.tools import wait, StopWatch, DataLog

from pybricks.robotics import DriveBase

from pybricks.media.ev3dev import SoundFile, ImageFile

# This program requires LEGO EV3 MicroPython v2.0 or higher.

# Click "Open user guide" on the EV3 extension tab for more information.

# Create your objects here.

hub = MoveHub()

ev3 = EV3Brick()

def Disparar():

m2.run\_time(1000, 90000, wait=False)

wait(9000)

m = Motor(Port.A)

m3 = Motor(Port.B)

m2 = Motor(Port.D, positive\_direction=Direction.COUNTERCLOCKWISE)

boton = Button.DOWN

# Write your program here.

#ev3.speaker.set\_speech\_options(language='es',voice=None, speed=None, pitch=None)

#ev3.speaker.say("hola gente")

#ev3.speaker.beep(frequency=1000, duration=200)

#ev3.speaker.play\_notes(['C4/4', 'G4/4', 'D4/4'])

m.run\_target(1000, -45, Stop.HOLD,wait=True)

#while True :

# wait(100)

m.hold()

m3.run\_time(1000,1000)

if(hub.boton.pressed()):

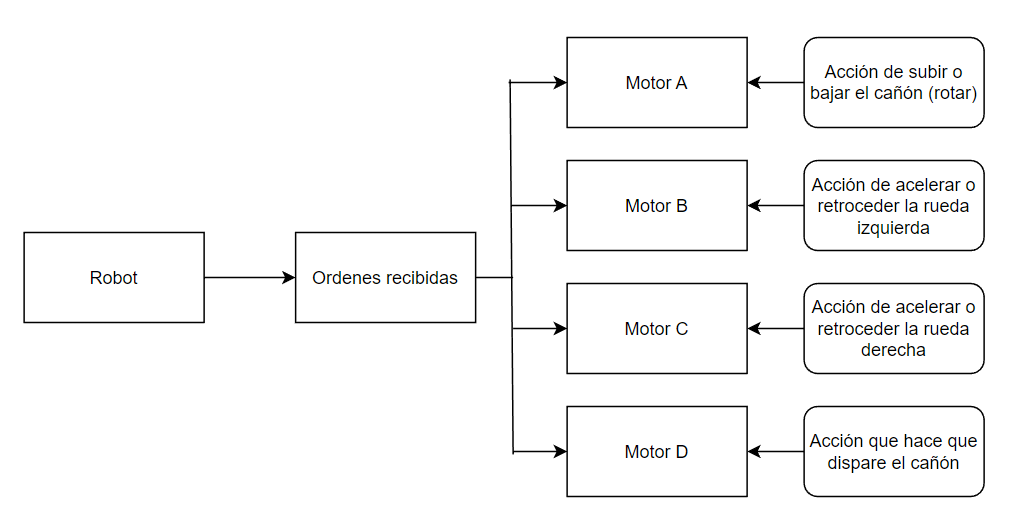
Disparar()

#wait(1500)

#Disparar()

#d.straight(0)

## Diagramas



# Resultados

## Estado actual del Proyecto

Estado actual del robot:

* + - * El diseño del robot está finalizado sin problemas a la hora de manipular.
      * Funciones definidas.
      * Interfaz de comando.
      * Movimiento del robot a la hora de girar

En desarrollo:

* La wiki del proyecto en desarrollo.
* La carta Gantt en proceso.
* Diseño en la interfaz.
* Movimiento del cañón de subida y bajada y disparo

## Problemas Encontrados y solución Propuesta

Problemas encontrados:

* Conectar la interfaz creada mediante tkinter a través de RPYC al ev3 en Lenguaje de Python
* Problema con el USB wifi para poder conectarnos con el robot ev3

Soluciones de los problemas:

* Conexión a través del programa PuTTY
* Se llego a la solución de implementar otra librería

# Conclusión

En la formulación del proyecto, a la hora de diseñar el robot es fundamental una buena comunicación con el equipo de trabajo ya que esto también influirá en la organización que se tendrá en el proyecto como la planificación de tareas a realizar que se serán acordadas por el equipo de trabajo.

Para la ejecución de cualquier tipo de proyecto es importante tener planes de emergencia en caso de que pase algún problema, para así brindar soluciones rápidas a dichos imprevistos.

Es necesario conocer bien las herramientas de trabajo que se estén utilizando en el proyecto.

# Referencia

enlaces de los que se recopilo información de los productos e investigaciones:

1. <https://www.solotodo.cl/products/146440-msi-modern-14-b5m-ryzen-5-5500u-8gb-256gb-ssd-w11h>

2.<https://www.falabella.com/falabella-cl/product/16110275/Laptop-M415DA-EK963W-AMD-Ryzen-7-8GB-RAM-512GB-SSD-14-FHD/16110275>

3.<https://articulo.mercadolibre.cl/MLC-984997712-mindstorm-ev3-core-set-45544-new-_JM#position=3&search_layout=stack&type=item&tracking_id=5b85e87d-9ae8-4de5-8be7-979235d32260>

4.<https://articulo.mercadolibre.cl/MLC-635353920-sony-memoria-micro-sdhc-8gb-clas10-cadaptador-70mb-ecoffice-_JM#position=3&search_layout=stack&type=item&tracking_id=4ec324ba-4667-4bb4-b7f1-cf1fad792243>