

**UNIVERSIDAD DE TARAPACÁ**



**FACULTAD DE INGENIERÍA**

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

**Ingenierí@**  
Computación e Informática

**Plan de Proyecto  
“P1NW1N3CH0 H1R4N3”**

**Alumno(os): Daniel Alday  
Benjamín Gómez  
Francisco Pantoja  
Tomás Silva**

**Asignatura: Proyectos I**

**Profesor: Humberto Urrutia**

**Arica, 2 de septiembre, 2022**

### Historial de Cambios

<b>Fecha</b>	<b>Versión</b>	<b>Descripción</b>	<b>Autor(es)</b>
25/10/2021	1.0	Formulación del Proyecto	

## Tabla de Contenidos

1. Panorama General	4
1.1. Introducción	4
1.2. Objetivos	4
1.2.1. Objetivo General	4
1.2.2. Objetivos Específicos	5
1.3. Restricciones Generales	6
1.4. Entregables	6
2. Organización del Personal	7
2.1. Descripción de los Roles	7
2.2. Personal que cumplirá los Roles	8
2.3. Mecanismos de Comunicación	8
3. Planificación del Proyecto	9
3.1. Actividades	9
3.2. Asignación de Tiempo	10
3.3. Gestión de Riesgos	11
4. Planificación de los Recursos	13
4.1. Hardware	13
4.2. Software y Servicios	13
4.3. Estimación de Costos	14
5. Análisis y Diseño	15
5.1. Especificación de Requerimiento	15
5.2. Arquitectura	16
5.3. Interfaz (Conexión con el robot)	21
6. Implementación	22
6.1. Fundamentos de Projectiles	22
6.2. Descripción de los programas	24
7. Resultados	26
7.1. Estado Actual del Proyecto	26
7.2. Problemas Encontrados y Solución Propuesta	26
8. Conclusión	27
9. Referencias	28



# 1. Panorama General

## 1.1. Introducción

El mundo científico tiende a descubrir problemas cada vez más complejos a medida que avanza el tiempo. Paralelamente, la sociedad siempre ha estado dispuesta a aceptar la ayuda de la tecnología para resolver sus problemas; en un principio con la creación de herramientas simples como los martillos o cuchillos, seguido muchos años después en la revolución industrial, llegando hasta los días actuales, donde la robótica y la inteligencia artificial forman parte fundamental de la vida.

El objetivo del presente proyecto es dar una solución a un problema planteado. Este problema puede ser resuelto fácilmente por un ser humano, pero surge la pregunta de cómo resolverlo sin la ayuda directa de uno. Para ello, usando conocimientos adquiridos durante los primeros dos años de carrera y con ayuda del robot LEGO MINDSTORM EV3, se conformarán las directrices del proyecto, pasando por todos sus respectivos procesos, los cuales son: planteamiento del problema, planificación, distribución de tareas, creación del diseño básico, armado y programación del robot, fase de pruebas y su despliegue.

De forma adicional, el uso de recursos externos para la resolución del problema, específicamente, conocimientos básicos de física y mecánica, también están considerados.

## 1.2. Objetivos

### 1.2.1. Objetivo General

Aplicar los conocimientos informáticos e ingenieriles necesarios para construir un robot que pueda controlarse de manera remota, sea capaz de moverse y pueda también lanzar un proyectil.

### 1.2.2. Objetivos Específicos

- Comunicarse remotamente con un robot a través de instrucciones dadas por código
- Implementar librería ev3.lang de Python
- Aplicar conocimientos adquiridos en la asignatura de Mecánica Clásica para que el robot lance un proyectil
- Ensamblar partes del robot de tal forma que sea resistente y cumpla su objetivo

### 1.3. Restricciones Generales

Realizar el proyecto no es tan sencillo como llegar y trabajar. Un análisis previo de las cosas con las que se trabaja es esencial para evitar problemas logísticos y de funcionamiento más adelante.

- a) Al realizarse la conexión con el robot mediante Wifi, implica que la misma por la cual se comunica el robot en algún momento puede volverse inestable, ya sea por su distancia con la fuente o lo saturada que la señal esté.
- b) El proyectil lanzado por el robot será casi siempre del mismo tipo (conservando más que nada su volumen y forma) y el ángulo de lanzamiento será constante, por lo que valores como distancia recorrida por el proyectil y velocidad inicial de disparo serán los mismos para todo lanzamiento. El cálculo de cuánto debe moverse el robot para acertar a su objetivo depende solamente del sensor de movimiento, el cual determinará si este se encuentra o no a la distancia apropiada para disparar.
- c) Dado el no-despreciable peso de los componentes Lego, es importante considerar al momento de la construcción todos los efectos adversos que el peso pueda ocasionar, para así no condicionar su movilidad ni su correcto funcionamiento.
- d) Como se debe trabajar con movimiento, dentro del mundo de los legos la única forma de realizarlo es usando engranajes de lego. Debido a su naturaleza, no todos los movimientos mecánicos (de la vida real) son posibles, y dependemos fuertemente de las piezas que se nos faciliten.
- e) La duración de la batería del robot es relativamente corta, por lo que su buena gestión es indispensable al momento de testear y presentar.

### 1.4. Entregables

En este proyecto, se piden como entregables los siguientes documentos:

- 1) Informes y Presentaciones:
  - a) Formulación del Proyecto
  - b) Ejecución del Proyecto
  - c) Fase Final del Proyecto
- 2) Bitácoras Semanales y Wiki del Proyecto
- 3) Robot en su estado final
- 4) Manual de Usuario

## 2. Organización del Personal

Una buena planificación de un proyecto requiere una adecuada gestión del personal, ya que mejora la organización y la facilidad de enfrentar problemas.

### 2.1. Descripción de los Roles

Por el número de participantes dentro del proyecto, se optó por un modelo de cuatro roles, que suplen las necesidades de liderazgo, programación, infraestructura y análisis.

- Jefe de Grupo:

Dentro del proyecto, es fundamental supervisar que los otros roles cumplan con su debida función además de influir en la planificación, gestión y asignación de las tareas. Al mismo tiempo, sirve de ayuda para los demás y es el encargado poseer la visión general que mantiene alineadas los quehaceres del equipo.

- Programador:

Su tarea es trabajar en el código general del proyecto. Tiene que estructurar y modularizar cada una de las funciones del robot por medio de codificación en Python y entender las relaciones que existan entre los sistemas presentes.

- Ensamblador:

Encargado de ingeniar y ensamblar cada uno de los mecanismos que conformarán al robot, siempre teniendo en cuenta las distintas ventajas y desventajas que los sistemas construidos aportan, además del mantenimiento y mejoramiento de los sistemas una vez terminados.

- Analista Logístico:

Se encarga de revisar que las ideas propuestas dentro del armado y programación del robot sean viables y efectivas. Trabaja en cercanía con el ensamblador y programador, realizando constantemente preguntas que corroboren que siempre se tomen buenas decisiones.

## 2.2. Personal que cumplirá los Roles

Para el proyecto, la asignación de roles se determinó de esta manera:

- a) Jefe de Grupo: Francisco Pantoja
- b) Programador: Daniel Alday
- c) Ensamblador: Benjamín Gómez
- d) Analista Logístico: Tomás Silva

## 2.3. Mecanismos de Comunicación

Dentro de un proyecto, es de vital importancia poder compartir información, así, se facilita mantenerse al tanto de lo que se trabaja y del progreso en general que el proyecto tiene.

Para el proyecto, fueron seleccionados cuatro vías preferentes de comunicación, que son: WhatsApp, Discord, Google Drive y de manera presencial. Los medios de comunicación descritos a continuación fueron escogidos debido a su simpleza de uso y extensión de funcionalidades.

- WhatsApp:  
Preferentemente utilizado para la comunicación por vía escrita y difusión de archivos multimedia. El personal usa esta aplicación para gestionar juntas en horario fuera de clases, además de avisar cosas importantes o emergencias que surjan repentinamente.
- Discord:  
Preferentemente utilizado para la comunicación por llamadas de voz y difusión de archivos en general. Es utilizado en largas sesiones principalmente para la redacción de documentos entregables como informes, bitácoras y presentaciones.
- Google Drive:  
Pensado para el resguardo de archivos importantes, los cuales (en su mayoría) serán subidos a la plataforma Redmine.
- Clases presenciales:  
Mecanismo más efectivo para avanzar físicamente en el proyecto, ya que cada uno de los integrantes se encuentra presente para realizar avance y aportar ideas en conjunto. Es indispensable la asistencia para que este mecanismo sea más efectivo.

## 3. Planificación del Proyecto

### 3.1. Actividades

El proyecto se dividió en cuatro partes fundamentales para su correcta concreción: Documentación, Entregas, Código y Armado del Robot.

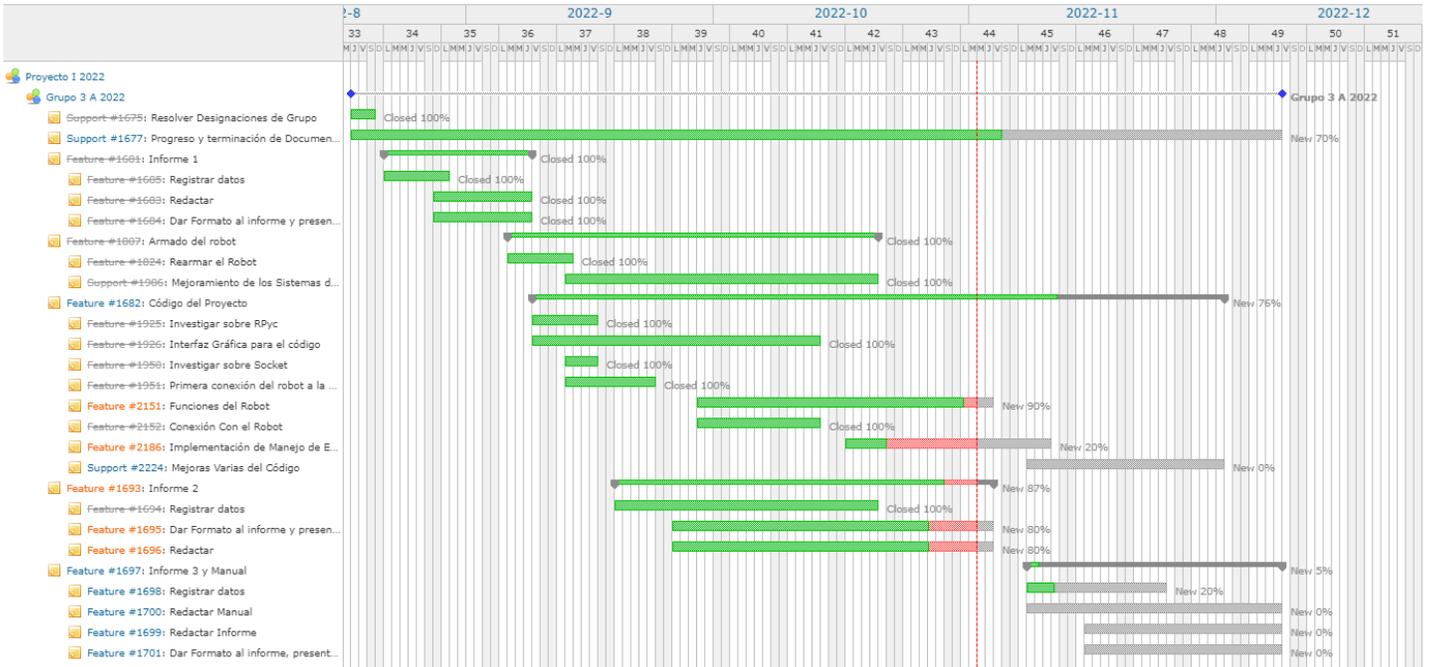
Las actividades que se planean realizar durante el desarrollo del proyecto para completar cada una de las partes anteriores son:

Tipo	Nombre	Responsable
Documentación	Progreso y Terminación de Documentos	Tomás Silva
Entregas	Redacción y edición del informe 1	Francisco Pantoja
	Redacción y edición del informe 2	Francisco Pantoja
	Redacción y edición del informe 3	Francisco Pantoja
	Redacción del Manual de usuario	Francisco Pantoja Daniel Alday
Código	Código del proyecto	Daniel Alday
	Interfaz Gráfica para el programa	Daniel Alday
Armado del Robot	Armado de la estructura final del robot	Benjamín Gómez Tomás Silva
	Mejoramiento de los sistemas del robot	Benjamín Gómez Tomás Silva
	Creación y fabricación de proyectiles	Benjamín Gómez

### 3.2. Asignación de Tiempo

Una herramienta muy importante al momento de gestionar tareas a lo largo del tiempo es la carta Gantt, la cual permite visualizar de forma continua cada una de las tareas propuestas.

A continuación se muestra la carta Gantt elaborada en conjunto:



### 3.3. Gestión de Riesgos

Los riesgos presentados aquí se categorizan según su tipo y según su gravedad, que está en la siguiente escala:

1	2	3	4	5
Insignificante	Muy Leve	Leve	Levemente Grave	Grave

6	7	8	9	10
Muy Grave	Gravísimo	Urgente	Extremadamente Urgente	Absolutamente Inmediato

Tipo de Riesgo	Riesgo	Gravedad (1-10)	Gestión
Documentación	Pérdida del formato o información relacionada a la documentación	1 – 4	Dependiendo de la extensión del documento, considerar primero restaurar (desde un punto en concreto) el documento; si es imposible, dedicar tiempo adicional lo más pronto posible para recrear (de 0) el documento.
	Imposibilidad de subida o transmisión de los archivos	3	De haber un problema con la subida física de los documentos a la plataforma que sea, considerar primero su necesidad: si es prescindible, considerar una nueva fecha posterior para su subida; si es imperioso para el avance del proyecto, considerar otros medios móviles para su subida y/o transmisión
Entregas	Pérdida del formato o información relacionada a los entregables	3 – 6	Dependiendo de lo próximo que se encuentre el entregable de su fecha de entrega, se determinará en el momento de cuánto tiempo dedicar a reparar los daños producidos al documento.
	Imposibilidad física (de cualquier tipo) de presentación de los entregables	6	Dependiendo de la proximidad a presentar del entregable, se consideran dos opciones: si no se está tan próximo a presentar, se deberá intentar conseguir nuevo implemento que permita su correcta presentación; si el problema se presenta directamente en su presentación, se deberá recurrir también a implemento habilitado para la presentación, pidiendo disculpas por la demora y procurar no llenar de vacíos la presentación en cuestión.

## Proyecto Plan de Proyecto Avance

Código	Batería del Robot insuficiente	3	Preocuparse de cargar siempre que sea posible al robot.
	Fallo en el programa que produzca que las funciones tengan un comportamiento incontrolable e indeseado	4	Detener las pruebas aplicadas en el momento, revisar, analizar y aplicar manejo de excepciones en donde se crea que ocurre el problema.
	Necesidad de cambio de librerías por funcionamiento indeseado, ininteligibilidad del código o no prestación de las funciones requeridas	5 – 7	Dependiendo del avance hecho antes de descubrir el problema, se pueden tomar dos caminos: Sin haber tenido mucho avance con la librería problemática, se pueden reescribir brevemente las funciones y rehacer en distinto código las mismas funciones requeridas; si el problema se descubrió realizado ya un avance considerable, se considerará primero reescribir el código y funciones de nuevo, con otro código, y si no es plausible, se analizará el grado de dificultad o uso de la librería en cuestión, si es muy compleja o impráctica, se considera el paso anterior.
Armado	Rotura o mal funcionamiento de engranado ocasionado por exceso, insuficiencia o mal aplicación de torque	4 - 8 (Levemente grave a Urgente)	A corto plazo, reparar el daño y analizar si la rotura fue provocada por una mala construcción, por un impedimento físico de la pieza, o por excesiva/insuficiente fuerza aplicada por el motor. De ser la primera opción, bastaría con un leve análisis afianzar la estructura actual; de ser la segunda, reconsiderar la estructura actual y plantear una mejor a largo plazo; de ser la tercera, bastaría con cambiar de manera inversa al problema la velocidad (si el problema fue mucha velocidad, disminuirla y viceversa), analizar y determinar si el problema radicaba en un mal uso o en alguno de los problemas anteriores.
	Rotura o mal funcionamiento de algún sistema funcional del robot	6 – 9 (Muy Grave a Extremadamente Urgente)	Como rearmar el robot significa dedicar muchísimo tiempo analizando, construyendo y probando, se consideran las siguientes acciones dependiendo de la proximidad del éste a su presentación:  Si el problema se presenta mucho antes de la presentación y: 1) El problema radica en el engranado del sistema: revisar el punto anterior. 2) El problema radica en un impedimento físico de alguna(s) pieza(s): Analizar el daño o mal funcionamiento del sistema y repararlo, con un nuevo diseño, lo más pronto posible. 3) No cumple con las expectativas previstas: Analizar los errores cometidos, ventajas y desventajas del sistema construido y revisar cómo mejorarlo (añadir a la construcción) o cambiarlo para conservar las ventajas y minimizar las desventajas que proveía el diseño anterior.  Si el problema se presenta muy próximo a la presentación y: 1) El problema radica en algún error o falla de construcción (sea engranado o impedimento físico): Intentar recuperar el diseño funcional anterior más próximo al actual, reconstruyendo sin intentar modificar grandemente el diseño para evitar la mayor cantidad de fallos posibles. Si no existe uno, dedicar tiempo adicional para su reparación. 2) No cumple con los requerimientos de la exposición: Intentar idear una solución temporal y mínimamente efectiva que cumpla con los requerimientos, priorizando que no sea compleja para su posterior mejoramiento o reconstrucción.

## 4. Planificación de los Recursos

Además de la planificación de personal y de las actividades, dentro de un proyecto también se tiene que tener en cuenta los recursos de los que se dispone para trabajar.

A continuación se detalla cada recurso requerido para la realización del proyecto:

### 4.1. Hardware

En recursos de hardware, se detalla lo siguiente:

- a) Un Kit LEGO MINDSTORMS EV3
- b) Piezas adicionales que provengan de otros kits EV3
- c) Computadores públicos y propios (para codificar y generar documentos)
- d) Un Adaptador de Wifi para el robot
- e) Una Tarjeta MicroSD clase 10 de 8 GB

### 4.2. Software y Servicios

En recursos de software y servicios, se detalla lo siguiente:

- a) Lenguaje de programación Python
- b) Putty
- c) Sistema operativo ev3\_dev
- d) Visual Studio Code y Nano para la implementación del código
- e) Plataforma Redmine

## 4.3. Estimación de Costos

### 4.3.1. Costos de Recursos

Categoría	Recurso	Cantidad	Costo
Hardware	Notebook propio	1	CLP \$1.000.000
	Kit LEGO MINDSTORMS EV3	1	CLP \$800.000
	Tarjeta Micro SD (Clase 10) 8Gb	1	CLP \$10.000
	Adaptador Dongle USB Wifi	1	CLP \$8.000
	Piezas adicionales LEGO	Ind.	Gratuito
Software	Visual Studio Code	-	Gratuito
	Discord (Sin licencia Nitro)	-	Gratuito
	WhatsApp	-	Gratuito
	Nano	-	Gratuito
	S.O. ev3_dev	-	Gratuito
Servicios	Plataforma Redmine	-	Gratuito (Por la institución)
	Google Workspace (Gmail, Drive, etc.)	1 plan	Gratuito (Por la institución)
	Licencia Microsoft 365 Empresa Básico	1 plan	USD \$ 6.00 por mes (≈ CLP \$ 5.668,2) Aprox. CLP \$ 34.009,2 en 6 meses
Costo Total			CLP \$ 1.852.009,2

### 4.3.2. Costos de trabajo

Rol Encargado	Horas Asignadas	Semanas de Trabajo	Valor hora	Costo Total
Jefe de Grupo	8 horas por semana	16 semanas	CLP \$ 20.000	CLP \$ 2.560.000
Ensamblador			CLP \$ 22.000	CLP \$ 2.816.000
Programador			CLP \$ 25.000	CLP \$ 3.200.000
Analista Logístico			CLP \$ 18.000	CLP \$ 2.304.000
Costo total				CLP \$ 10.880.000

### 4.3.3. Costo Total del Proyecto

Considerando las dos previas tablas, se puede obtener un estimado de **\$12.732.000** (doce millones, setecientos treinta y dos mil pesos chilenos) destinados para la completación del proyecto

## 5. Análisis y Diseño

### 5.1. Especificación de Requerimiento

El robot deberá ser capaz de moverse y disparar a un objetivo determinado. El disparo se efectuará utilizando un proyectil y su trayectoria tendrá que ser calculada siguiendo el modelo de lanzamiento parabólico (ideal) visto en la asignatura de mecánica clásica.

A continuación, se listan los requerimientos funcionales (tareas que debe cumplir el robot) y no funcionales (condiciones que deben darse para que pueda realizar sus tareas) que se deben cumplir para lograr el objetivo previsto:

Requerimientos Funcionales:

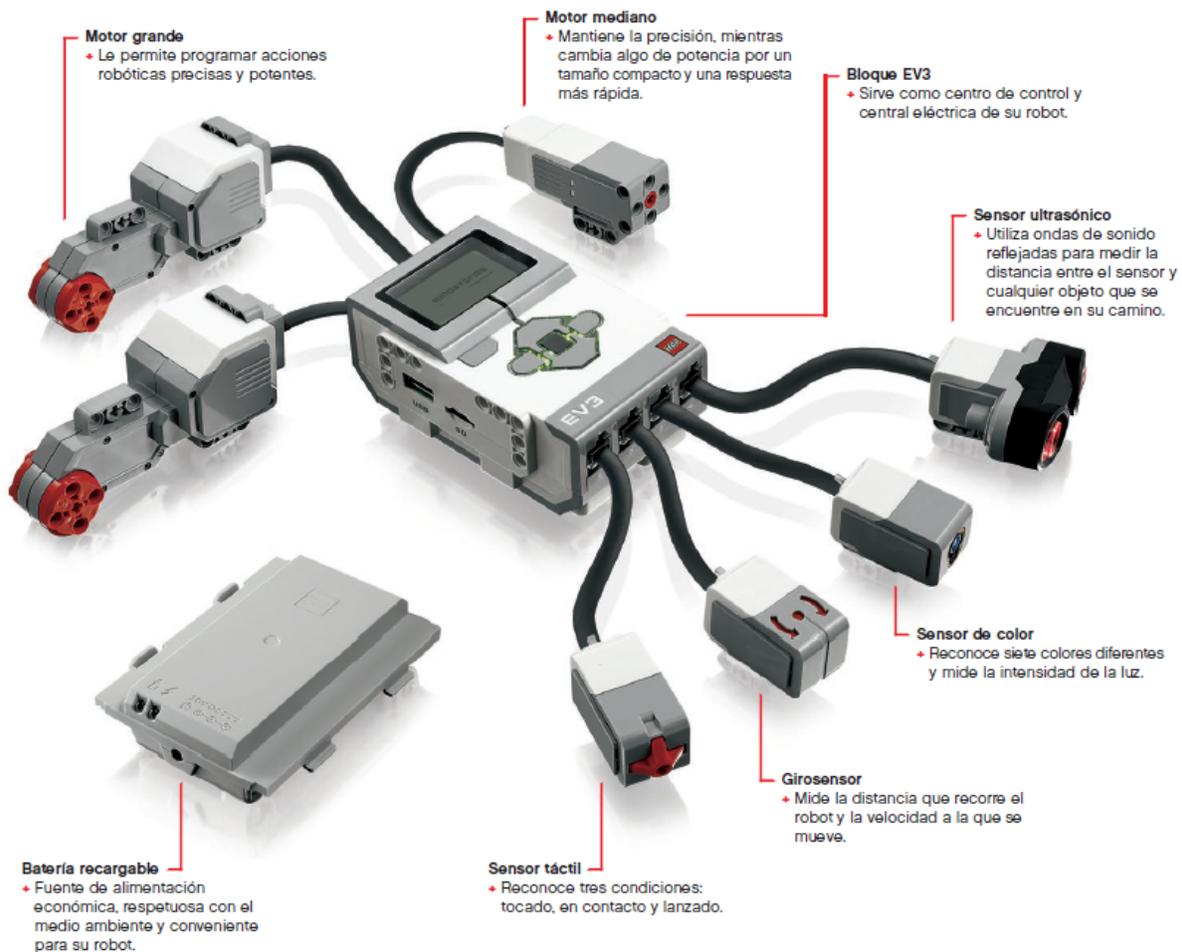
- Crear un servidor para poder comunicarse con el medio cliente
- Mediante una interfaz gráfica:
  - Moverse por los ejes x e y
  - Disparar un proyectil (siguiendo el modelo de lanzamiento parabólico ideal)

Requerimientos No Funcionales:

- Mecánicos (físicos):
  - Mantener la estabilidad en todo momento
  - Ser capaz de soportar el mecanismo de disparo
- De software:
  - El robot debe estar construido únicamente por las piezas del kit LEGO MINDSTORMS ev3 y por las adicionales prestadas.
  - La interfaz gráfica deberá ser escrita en el lenguaje de programación Python, utilizando la librería Tkinter (los algoritmos y funciones internas del robot pueden estar escritos en algún otro lenguaje conocido (ej: Java, Dart))

## 5.2. Arquitectura

El bloque EV3 internamente se compone de un procesador ARM9, una memoria SD que soporta su sistema operativo EV3-Dev, basado en Linux; un dongle de conexión Wifi y 8 puertos RJ12 modificados para sensores y motores.

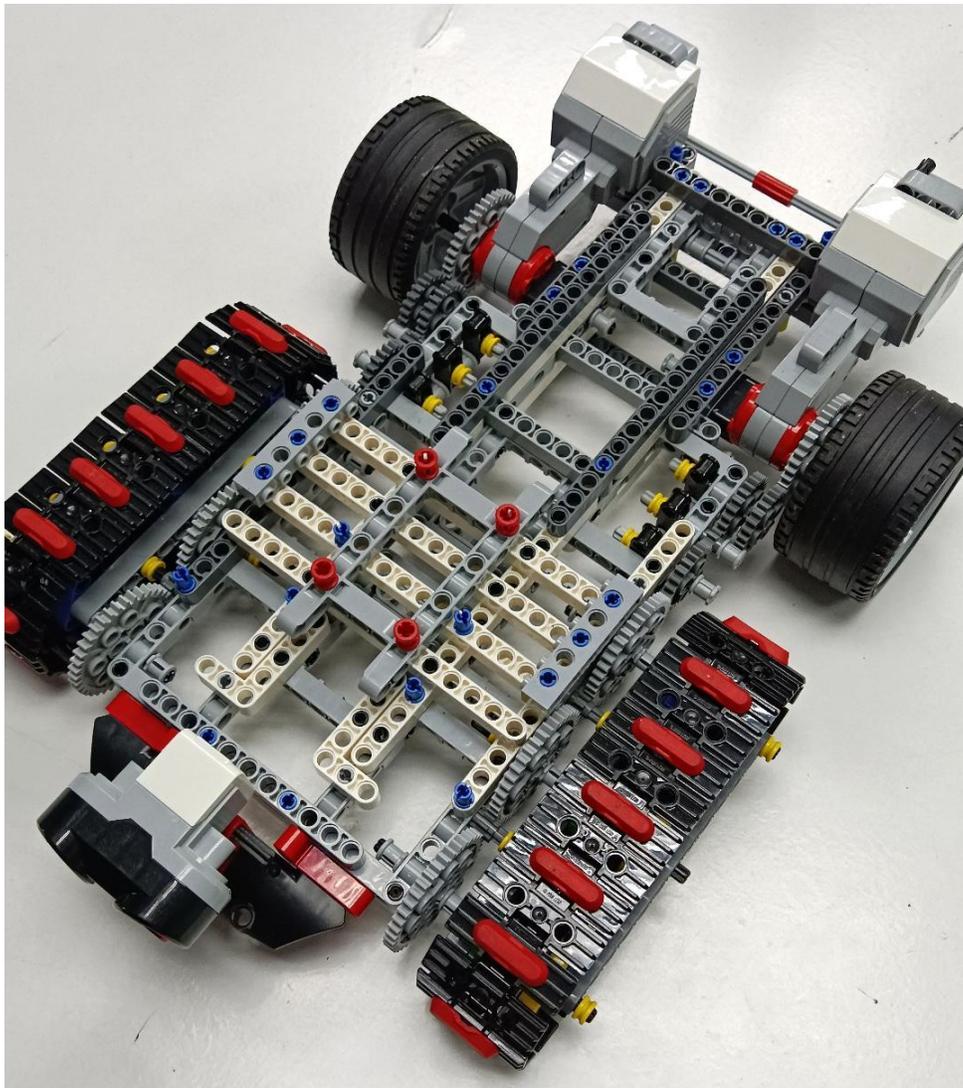


El presente robot usará tres puertos para motores y uno para sensores.

Los distintos sistemas presentes en el robot se detallan a continuación:

### 5.2.1. Sistema de Movimiento

El movimiento del robot se lleva a cabo gracias a una base sólida, la cual consta de dos ruedas traseras y dos orugas. Las ruedas traseras aprovechan la potencia de 2 LargeMotor que impulsan a las orugas, y éstas, a su vez, facilitan su movimiento sobre superficies irregulares. La conexión entre ruedas y orugas es mediante un sistema de engranado, el cual transmite uniformemente el movimiento a la base sólida.

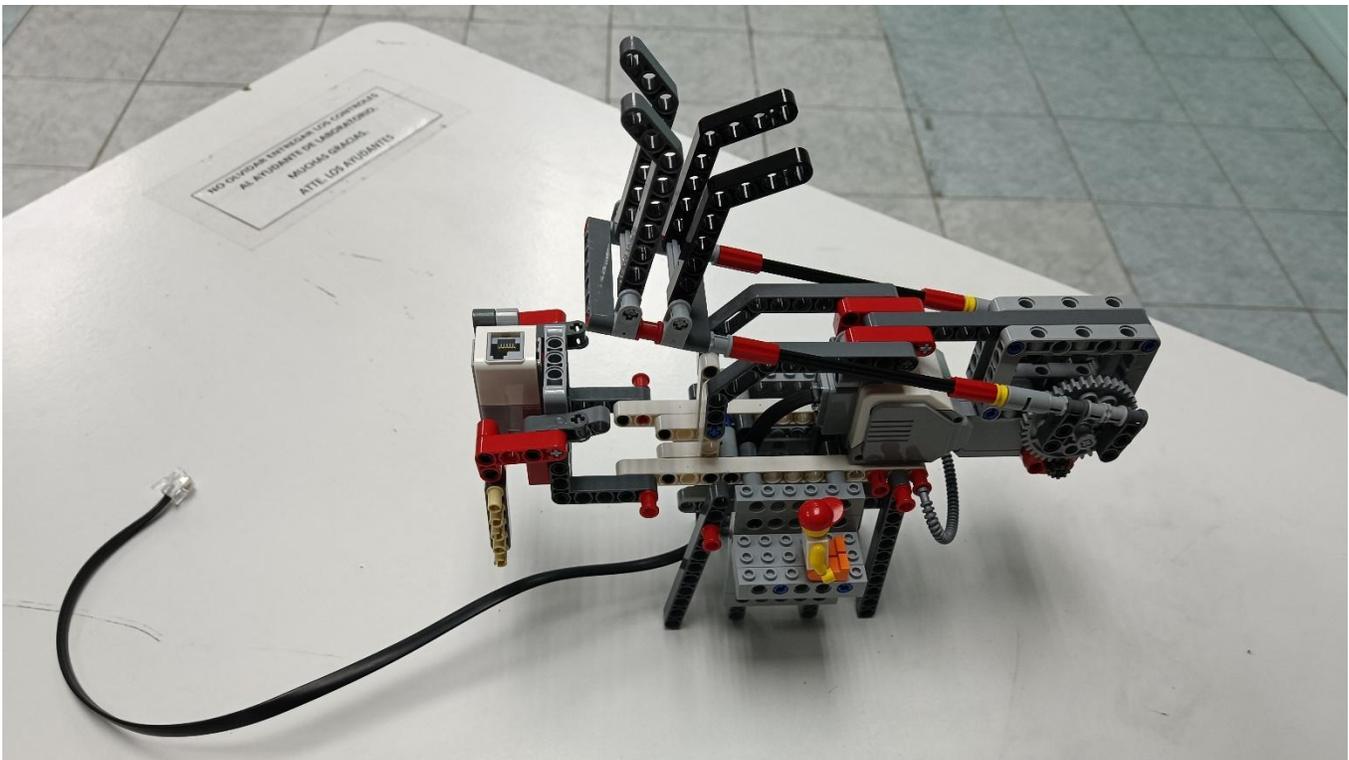


### 5.2.2. Sistema de disparo

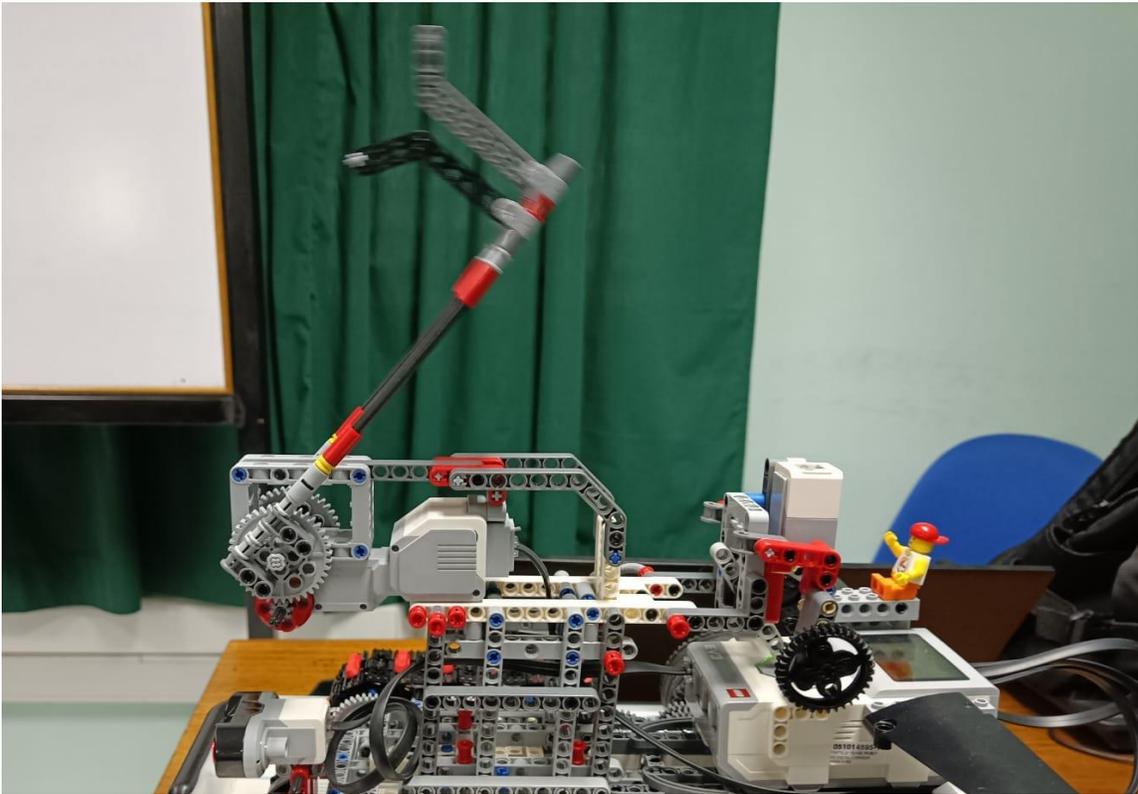
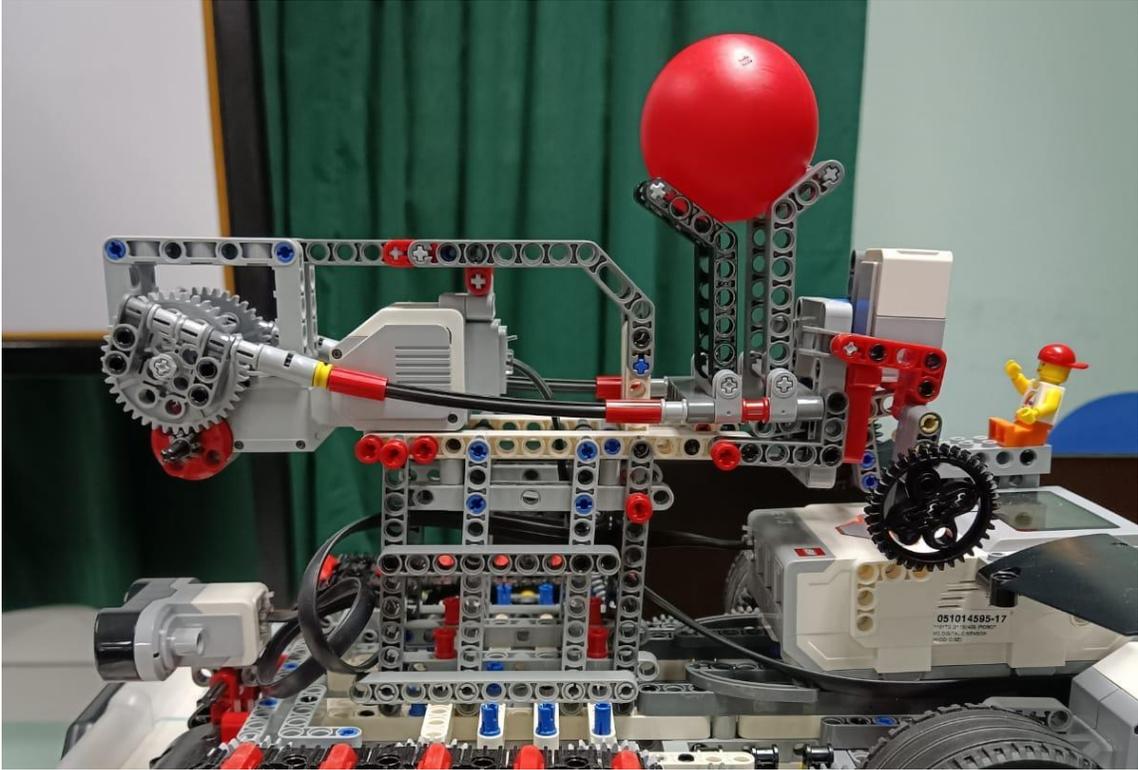
El proyectil es lanzado por una catapulta, la cual funciona con un LargeMotor. La catapulta consta de dos fases, fase de carga y fase de disparo.

La fase de carga es considerada como el estado neutro del robot, su ángulo de inclinación es  $180^\circ$ . La fase de disparo comienza con el giro del LargeMotor, el cual gira al 100% de su capacidad y genera flexión en las varillas gracias al tope que las mantiene en su lugar por un corto tiempo determinado.

Al tensar, se produce una transformación de una energía cinética a una potencial elástica, la cual vuelve a ser transformada nuevamente a cinética al momento del lanzamiento.



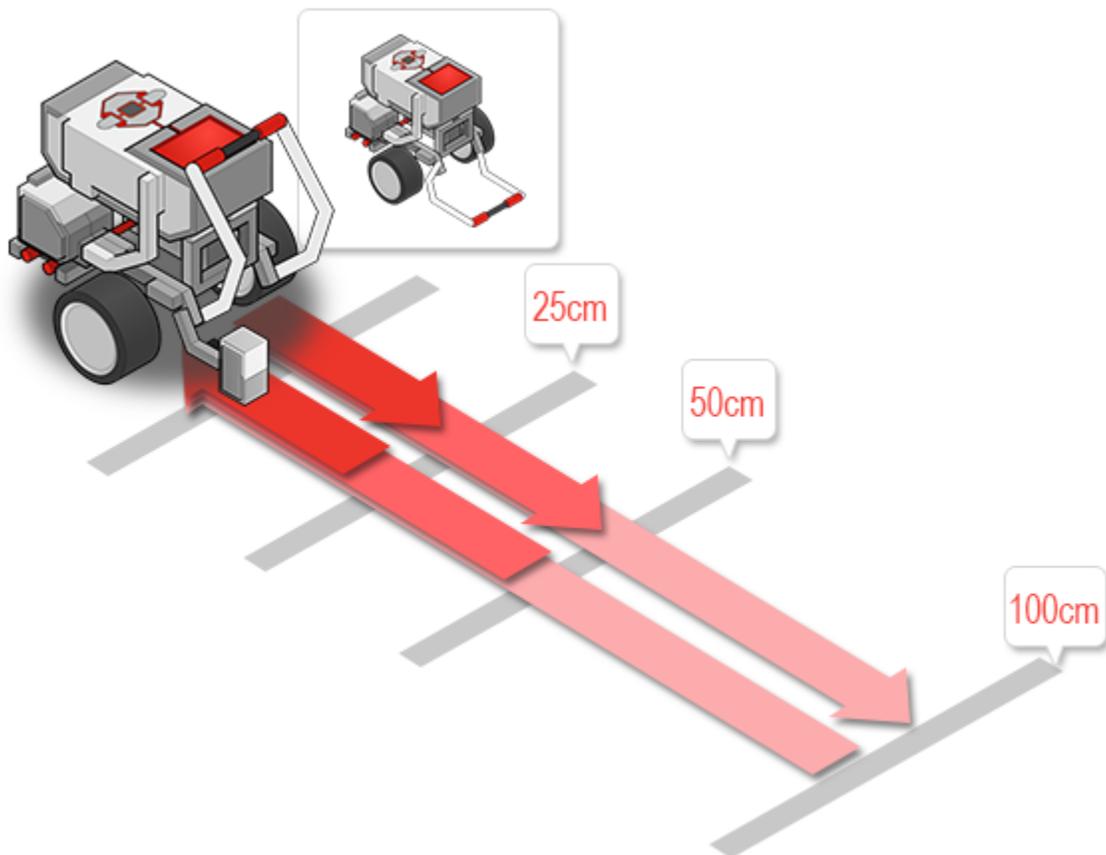
Proyecto Plan de Proyecto Avance



### 5.2.3. Sistema de sensores

Este apartado engloba al sensor de distancia ultrasónico, el cual utiliza ondas de sonido reflejadas para medir distancia entre el sensor y cualquier objeto que se encuentre en su camino. La distancia máxima que registra el sensor es de 2500 milímetros (2.5 metros).

Una vez detectada la distancia óptima de disparo (1.2 metros), el robot emitirá un sonido, indicando que es seguro disparar a esa distancia.

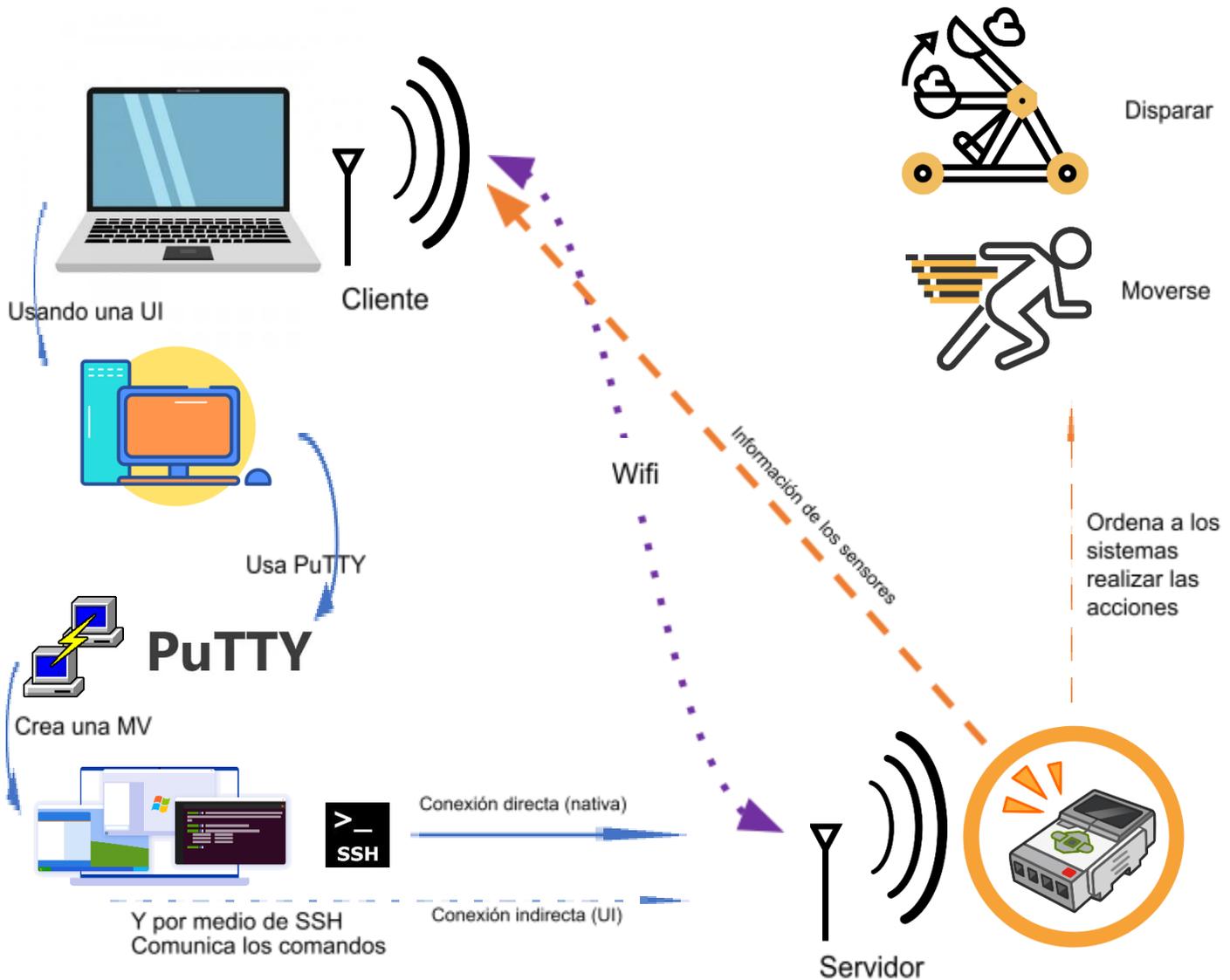


### 5.3. Interfaz (Conexión con el robot)

La comunicación con el Robot se realiza mediante Wifi. El cliente y conexión son posibles gracias a la plataforma Putty para Windows. Esta plataforma crea una máquina virtual en la cual corre un terminal de Linux, donde se ejecuta el programa SSH, en el que nativamente se puede interactuar con el sistema operativo EV3-Dev, es aquí que el robot a través del lenguaje Python, recibe las instrucciones a ejecutar.

Para controlar el proceso de la conexión, se utiliza otro archivo de Python donde se encuentra la interfaz gráfica, la cual se utiliza para enviar las instrucciones de forma remota sin tener que lidiar con los procesos internos.

El siguiente diagrama ilustra la conexión de manera pictórica:



## 6. Implementación

### 6.1. Fundamentos de proyectiles

Los proyectiles lanzados por la catapulta realizan un movimiento denominado «*movimiento parabólico ideal*», el cual es estudiado a profundidad en la rama de la física Mecánica Clásica.

El movimiento parabólico se caracteriza por ser en dos dimensiones únicamente. Es así, porque la variación de movimiento (velocidad) en el eje x es siempre la misma, ya que se considera que no hay ninguna fuerza de oposición al movimiento en ese eje. Por el contrario, la variación de movimiento en el eje y sí se ve modificada uniformemente debido a la aceleración de la gravedad presente en todo tiempo.

Inicialmente hay un ángulo de inclinación inicial, el cual determina, junto de la velocidad de lanzamiento inicial, la distancia total que recorrerá el proyectil.

Las siguientes ecuaciones describen el movimiento parabólico de la siguiente forma:

*Sea  $\theta$  el ángulo de lanzamiento y  $v_i$  la velocidad con la que se lanza el proyectil:*

*Posición en eje x, en función del tiempo*       $x(t) = v_i \cdot \cos(\theta) \cdot t$

*Velocidad en eje x, en función del tiempo*       $v_x(t) = v_i \cdot \cos(\theta)$

*Posición en eje y, en función del tiempo*       $y(t) = v_i \cdot \sin(\theta) \cdot t - \frac{1}{2} g \cdot t^2$

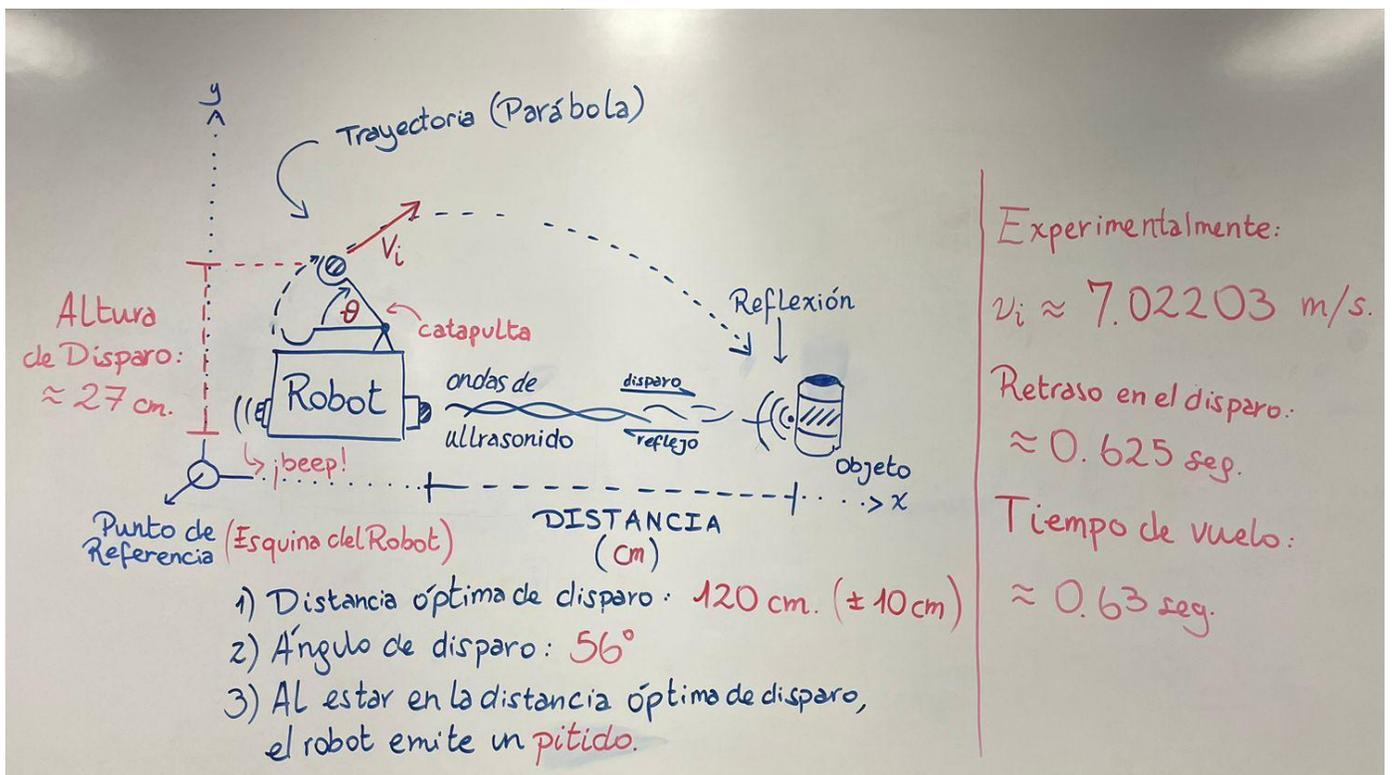
*Velocidad en eje y, en función del tiempo*       $v_y(t) = v_i \cdot \sin(\theta) - g \cdot t$

## Proyecto Plan de Proyecto Avance

En este proyecto se determina que la catapulta es un muy buen ejemplo al momento de explicar el movimiento parabólico, aunque cabe recalcar que para este sistema de disparo, el ángulo de inclinación inicial al cual sale eyectado el proyectil será siempre el mismo.

Experimentalmente, la distancia recorrida del proyectil en el eje x es de 1.2 metros, por lo que el único trabajo del robot será colocarse a esa distancia de su objetivo para impactarlo.

El siguiente diagrama modela el lanzamiento del proyectil efectuado por el robot:



La velocidad inicial es muy importante al momento de describir el movimiento, ya que es el impulso que recibe el proyectil, dándole capacidad de llegar a una distancia específica.

El cálculo de la velocidad inicial está dado por el siguiente despeje:

Es prudente dar cuenta también de los distintos desfases que se tienen en el sistema (en relación con el sistema de referencia), a pesar del problema que supone, dentro de los cálculos, no se omite ninguno.

## Cálculo de velocidad inicial

Distancia recorrida en eje x: 1,2 [m]

Tiempo de vuelo experimental: 0,63 [s]

Ángulo de disparo: 56°

$$x(t) = V_0 \cdot \cos\theta \cdot t \quad v_x(t) = V_0 \cdot \cos\theta$$

$$y(t) = V_0 \cdot \sin\theta \cdot t - \frac{g t^2}{2} \quad v_y(t) = V_0 \cdot \sin\theta - 9,8 t$$

\* Al momento del impacto,  $v_y(t) = 0$ :

$$0 = V_0 \cdot \sin\theta - 9,8 t \quad ; \quad t = \frac{V_0 \cdot \sin\theta}{9,8} \quad ; \quad V_0 = \frac{9,8 \cdot t}{\sin\theta}$$

$$V_0 = \frac{9,8 \cdot 0,63}{0,82}$$

\* Sabiendo que la distancia recorrida es 1,2 [m]:

$$1,2 = V_0 \cdot \cos\theta \cdot t \quad \Rightarrow \quad 1,2 = V_0 \cdot \cos\theta \cdot 0,63 \quad ; \quad V_0 = \frac{1,2}{\cos\theta \cdot 0,63}$$

$$\therefore \vec{V} = V_0 + V_0 \quad ; \quad |\vec{V}| = \sqrt{V_0^2 + V_0^2} = \sqrt{57} \approx 7,03 \text{ [m/s]}$$

## 6.2. Descripción de los programas

La mayor parte de los sistemas de comunicación se implementaron con la librería *socket*. La comunicación del robot se basa en la dinámica de servidor-cliente que consta de la comunicación bidireccional entre dos dispositivos, siempre y cuando ambos estén conectados a la misma red Wifi.

### 1) Dentro del Robot

#### a) SERVIDOR (main.py):

Es el programa encargado de permitir la conexión mediante un servidor creado por la librería *socket* de Python. Su función principal es leer las instrucciones que son enviadas desde el cliente. Las instrucciones están formateadas por caracteres dentro de un bucle que se ejecuta infinitamente hasta que termina la conexión.

#### b) FUNCIONES (Funciones.py):

En este archivo de Python se encuentran definidas las funciones encargadas de comunicarse con el robot. En ella también están definidas cada una de las funciones que el robot puede realizar.

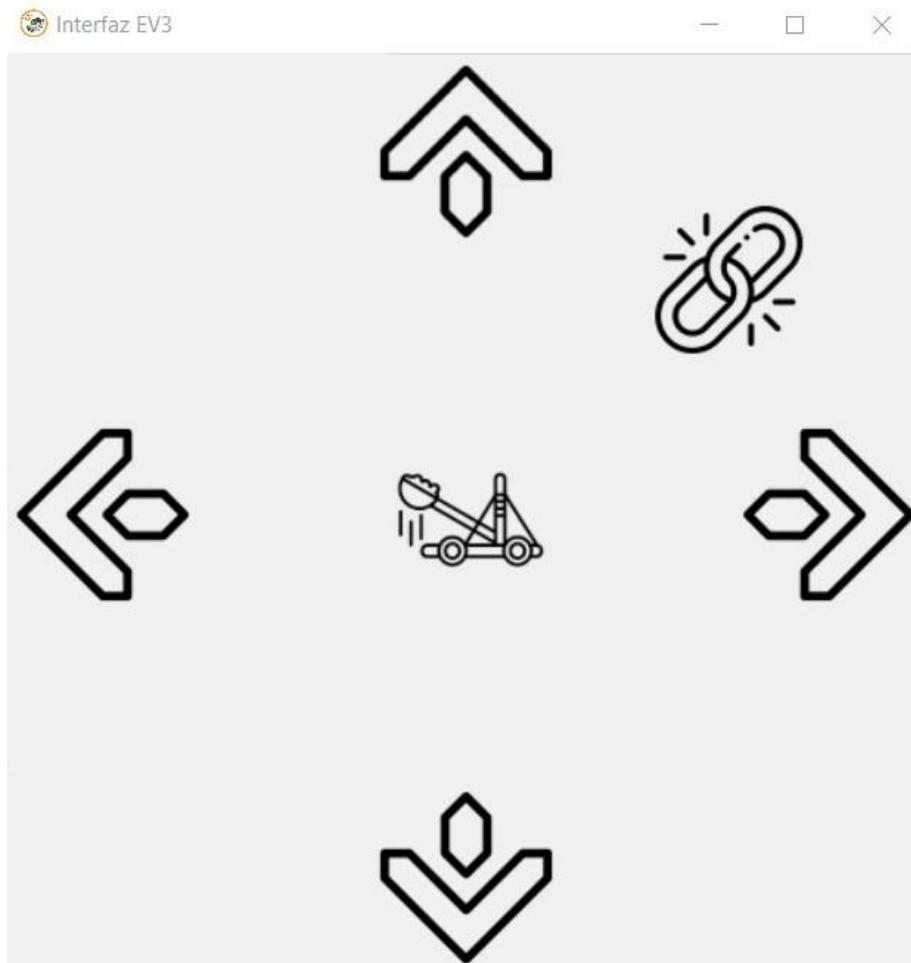
Para todo esto, se decidió utilizar las clases y métodos proporcionados por la librería *ev3\_dev2*.

2) Dentro del Computador

a) CLIENTE (interfaz.py):

Despliega una interfaz gráfica creada con la librería Tkinter de Python. En ella están los botones de los movimientos básicos del robot, además del botón de conexión y de disparo.

El botón de conexión está implementado con la librería *socket* que permite enlazar al computador directamente con el robot. En caso de fallar la conexión o la existencia de otro problema, se manejan excepciones que hacen la interacción y uso del programa mucho más seguro.



## 7. Resultados

### 7.1. Estado Actual del Proyecto

A la fecha de hoy el robot cumple con todas las funcionalidades descritas con anterioridad. El ensamblaje básico está igualmente implementado, por lo que se ha cumplido con lo propuesto en la fase de formulación del proyecto. Por el lado de la programación se ha cumplido con lo propuesto igualmente. Finalmente, en la plataforma Redmine, se ha ido actualizando los avances grupales en la carta Gantt, faltando solamente actualizar la Wiki, la cual será más tomada en cuenta en la fase final del proyecto.

### 7.2. Problemas Encontrados y Solución Propuesta

A lo largo de la fase de planificación y ejecución del proyecto se han encontrado los siguientes problemas:

<b>Problema</b>	<b>Solución</b>
Base de movimiento con cuatro motores	Bajo este diseño, no se daba lugar a la catapulta, ya que el robot posee entradas limitadas para motores: Se rediseñó la base de tal forma que el movimiento fuera realizado solo con dos motores, dejando espacio a el motor de la catapulta.
Ruedas chocando con los engranajes debido al peso de la catapulta	Se aligeró la base de la catapulta, permitiendo una mayor movilidad.
Fallas en la comunicación con el robot debido a una conexión a internet débil en ciertas circunstancias, provocando	Testear el robot en momentos que no haya otros grupos conectados, de lo contrario, generar conexión con red Wifi proveniente de un dispositivo móvil.

## **8. Conclusión**

A medida que se avanza dentro del proyecto, se han encontrado una serie de dificultades, que, a pesar de haber sido un problema, no fueron impedimento para el desarrollo íntegro de cada una de las partes del proyecto. Además, es importante destacar que los resultados obtenidos son los esperados según la planificación que el grupo ha establecido en la fase anterior. En la próxima fase del proyecto se dará más énfasis en hacer llegar el proyecto al usuario común, creando el manual de usuario y mejorando la interfaz gráfica.

## 9. Referencias

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

<https://www.hostmath.com/>

[http://ceca.uaeh.edu.mx/informatica/oas\\_final/OA6\\_lego/componentes.html](http://ceca.uaeh.edu.mx/informatica/oas_final/OA6_lego/componentes.html)