**UNIVERSIDAD DE TARAPACÁ**

**FACULTAD DE INGENIERÍA**

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

Imagen que contiene Forma

Descripción generada automáticamente

**Plan de Proyecto**

**“MP3K”**

**Alumno(os):** Karen Correa

Jhosep Marca

Fabián Orellana

Iván Viscarra

**Asignatura:** Proyecto 1

**Profesor:** Humberto Urrutia López

**DICIEMBRE – 2022**

**Historial de Cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor(es)** |
| 25/08/2021 | 1.0 | Primera Versión del informe, se incluye el Panorama General y la Organización del Personal. | Karen Correa, Jhosep Marca, Fabián Orellana, Iván Viscarra. |
| 06/09/2022 | 1.1 | Segunda Versión del Informe, se adiciona Planificación del Proyecto y Planificación de los recursos. | Karen Correa, Jhosep Marca, Fabián Orellana, Iván Viscarra. |
| 06/10/2022 | 2.0 | Actualización a avance del informe | Karen Correa, Jhosep Marca, Fabián Orellana, Iván Viscarra. |
| 08/11/2022 | 3.0 | Actualización Informe Final | Karen Correa, Jhosep Marca, Fabián Orellana, Iván Viscarra. |

Tabla de Contenidos

[1.1 Introducción 4](#_Toc121947831)

[1.2 Objetivos 4](#_Toc121947832)

[1.2.1 Objetivo General 4](#_Toc121947833)

[1.2.2 Objetivos Específicos 4](#_Toc121947834)

[1.3 Restricciones 4](#_Toc121947835)

[1.4 Entregables 4](#_Toc121947836)

[2 Organización del Proyecto 5](#_Toc121947837)

[2.1 Descripción de Roles 5](#_Toc121947838)

[2.2 Personal que cumplirá los Roles 5](#_Toc121947839)

[2.3 Mecanismos de comunicación 5](#_Toc121947840)

[3 Planificación de los Recursos 6](#_Toc121947841)

[3.1 Actividades 6](#_Toc121947842)

[3.2 Asignación de tiempo 6](#_Toc121947843)

[*3.3* Gestión de Riesgo 7](#_Toc121947844)

[4 Planificación de los Recursos 8](#_Toc121947845)

[4.1 Hardware 8](#_Toc121947846)

[4.2 Software 8](#_Toc121947847)

[4.3 Estimación de Costos 8](#_Toc121947848)

[5 Análisis y Diseño 8](#_Toc121947849)

[5.1 Especificación de Requerimiento 8](#_Toc121947850)

[5.2 Arquitectura 9](#_Toc121947851)

[5.3 Interfaz 9](#_Toc121947852)

[6 Implementación 10](#_Toc121947853)

[6.1 Fundamentos de Proyectiles 10](#_Toc121947854)

[6.2 Descripción de los programas 11](#_Toc121947855)

[6.3 Diagramas 17](#_Toc121947856)

[7 Resultados 17](#_Toc121947857)

[7.1 Estado Final del Proyecto 17](#_Toc121947858)

[7.2 Problemas Encontrados y Solución Propuesta 17](#_Toc121947859)

[7.3 Trabajo a Futuro 18](#_Toc121947860)

[8 Conclusiones 18](#_Toc121947861)

[9 Referencias (Estándar IEEE) 19](#_Toc121947862)

Panorama General

## Introducción

En este informe se presentarán los objetivos del proyecto, la planificación para lograr terminar en el plazo de tiempo determinado el diseño del robot y que cumpla sus funciones. Además, se describirán las actividades y roles que cada integrante del grupo cumple, se analizarán los posibles riesgos que compliquen la realización del trabajo.

Adicionalmente se incluirán los costos estimados del proyecto.

El Kit de Lego Mindstorms EV3 está diseñado para el aprendizaje de estudiantes que deseen adquirir experiencia en la codificación, contiene una pequeña computadora programable que puede controlar sus motores y recopilar información de sensores.

En este proyecto se busca construir un robot con legos que ejecute la función de disparo de proyectiles mediante un programa creado en lenguaje Python, este robot deberá finalmente con sus funciones otorgadas intentar atacar a otros oponentes.

## Objetivos

### Objetivo General

Diseñar una estructura de un robot que disparará proyectiles hacia un objetivo, la función será ejecutada a través de una interfaz gráfica codificada en lenguaje Python, esta conexión será realizada con un servidor a través de protocolo SSH.

### Objetivos Específicos

* Construir una torreta capaz de expulsar objetos disparando a un objetivo.
* Construir la base del robot que servirá para permitir el desplazamiento de este.
* Unir una estructura estable que conecte la base y la torreta del robot.
* Crear una interfaz gráfica en lenguaje Python que permita dirigir las funciones del robot de forma remota.
* Establecer una conexión SSH que permita ejecutar funciones mediante una red wifi.
* Implementar librería EV3DEV 2 con funciones que otorgarán el movimiento y la reproducción del sonido en el robot.

## Restricciones

* Trabajar de acuerdo con las piezas de legos disponibles.
* El código de la interfaz gráfica debe estar en lenguaje de programación Python.
* El robot debe estar conectado a la misma red wifi del equipo.
* Hay una distancia limitada a la que el robot se puede desplazar.
* El robot cuenta con una batería, la cual tiene un tiempo limitado.

## Entregables

* Informes del Proyecto, donde se documentará el proceso de la construcción del robot.
* Bitácoras Semanales, en las cuales se informará los avances semanales del proyecto.
* Carta Gantt, la que contiene la planificación del proyecto y los plazos en los que debe cumplirse cada actividad.
* Wiki, en la plataforma de Redmine y debe contener los datos importantes sobre el proyecto.
* Manual de Usuario, el cual debe contener instrucciones de uso para que cualquier usuario pueda comprender como utilizar el robot.

# Organización del Proyecto

## Descripción de Roles

Para la realización de este proyecto fue necesario y de suma importancia la distribución de roles, definiendo las tareas que cada integrante debe cumplir, esto para llevar un orden en lo que cada uno debe realizar, evitando así tanto la desorganización del grupo, como la improductividad. La organización en este trabajo fue crucial debido a los límites de tiempos establecidos. Los 3 aspectos principales considerados para los roles fueron: La programación del código que permite el funcionamiento del robot, el armado del diseño del robot elegido con las piezas disponibles y la realización de la documentación requerida para este proyecto. Además, se cuenta con un jefe de grupo que es encargado de tomar las decisiones finales y de asignar los demás roles.

## Personal que cumplirá los Roles

|  |  |  |  |
| --- | --- | --- | --- |
| **Actividad** | **Descripción** | **Involucrados** | **Responsable** |
| Programación | Crear el programa que ejecute las funciones para la torreta. | Karen Correa, Jhosep Marca, Fabián Orellana, Iván Viscarra. | Fabián Orellana  Iván Viscarra |
| Armado | Armar el robot con las piezas de lego y buscar alternativas para las piezas faltantes. | Karen Correa, Jhosep Marca, Fabián Orellana, Iván Viscarra. | Jhosep Marca |
| Documentación | Hacer informes de la Formulación del proyecto, bitácoras semanales, presentaciones. | Karen Correa, Jhosep Marca, Fabián Orellana, Iván Viscarra. | Karen Correa |
| Jefe de Grupo | Toma de Decisiones, Representa al equipo de trabajo. |  | Jhosep Marca |

## Mecanismos de comunicación

En su mayoría se intenta resolver las problemáticas dentro del horario de las clases, ya que así se pueden solucionar de forma más expedita, sin embargo, se cuenta con diferentes medios de comunicación para complementar el trabajo realizado en horario de clases.

* WhatsApp: Facilita la comunicación en caso de que a algún integrante le surja alguna idea, inconveniente o duda. El horario designado para ocupar este medio es de lunes a viernes, entre las 8 AM. y 10 PM.
* Discord: Ocupado para reuniones semanales fuera del horario de clases, en las reuniones se discuten detalles sobre el proyecto que no necesitan presencialidad. Las reuniones semanales ocurren los viernes a las 6 PM.
* Drive: En esta plataforma se actualizan los avances del informe y las presentaciones sobre el proyecto. El horario para trabajar en esta plataforma es de lunes a viernes, entre las 8 AM y las 10 PM.

# Planificación de los Recursos

## Actividades

Para la realización del proyecto se han asignado algunas actividades que deben ser cumplidas para poder llevar a cabo una correcta planificación y cumplir el objetivo principal a tiempo. A continuación, se presentan las actividades:

* Codificación de un programa Python: Se debe crear un programa provisional que ejecute movimientos en el robot como prueba de la estabilidad de la estructura y el correcto funcionamiento de funciones.
* Codificación de una Interfaz Gráfica: Se debe realizar una interfaz gráfica que permita a un usuario desde un equipo ejecutar acciones en el robot.
* Desarrollo de la documentación del proyecto: Se debe desarrollar la documentación correspondiente a este proyecto, tal como los informes donde se detalla el avance del proyecto y todas sus especificaciones, además de las bitácoras que se entregarán semanalmente e incluirán lo realizado en cada una.
* Proceso de Armado del robot: El robot debe tener una estructura que sea adecuada con las funciones que se le serán implementadas, para esto se armará tanto la torreta, como una base para el robot mediante piezas de lego que se unirán para formar una construcción estable.
* Actualización del proyecto en la plataforma Redmine: El proyecto debe ser documentado además en la plataforma Redmine y estar constantemente actualizado, para esto se usará principalmente la Carta Gantt y la Wiki presentes en la plataforma, así como también se subirán los informes y bitácoras con el avance de cada semana en el apartado de documentos.

## Asignación de tiempo

Para tener una estimación del tiempo que se utilizará en cada actividad del proyecto se ha confeccionado una Carta Gantt, con el propósito de organizar el tiempo entre actividades para una mayor eficiencia.

*Carta Gantt:*

Escala de tiempo

Descripción generada automáticamente

## Gestión de Riesgo

El nivel de impacto de los riesgos pronosticados se clasificó según el nivel:

|  |  |
| --- | --- |
| **CATEGORÍA DE RIESGOS** | |
| 1 | CATASTRÓFICO |
| 2 | CRÍTICO |
| 3 | MARGINAL |
| 4 | DESPRECIABLE |

El cual se le asignará al nivel Catastrófico como el más grave y el Despreciable el más leve.

|  |  |  |  |
| --- | --- | --- | --- |
| Riesgos | Probabilidad de Ocurrencia | Nivel de Impacto | Acciones Remediables |
| Que se rompa o extravíe el robot. | 5% | 1 | Hablar con el profesor para buscar el arreglo del robot o pedir otro. |
| Pérdida del código del programa con el que se ejecuta las funciones del robot. | 40% | 3 | Si se pierde todo el material, se debe empezar nuevamente, aunque los conocimientos adquiridos facilitarán la recreación del código. |
| Enfermedad o imposibilidad de asistir a clases de algún integrante. | 70% | 3 | Los demás integrantes deben intentar avanzar con las tareas de él o los integrantes ausentes para no atrasarse en la planificación. |
| Que se desarmen las piezas del robot ya construido. | 30% | 2 | Volver a armar el robot y reemplazar las piezas si es que alguna se dañó. |

# Planificación de los Recursos

## Hardware

|  |  |
| --- | --- |
| Hardware | Cantidad |
| Robot EV3 | 1 |
| Tarjeta SD | 1 |
| Dongle | 1 |
| Computador | 4 |

## Software

|  |  |
| --- | --- |
| Software | Cantidad |
| Visual Studio Code | 4 |
| Python | 4 |
| PuTTY | 4 |

## Estimación de Costos

A continuación, se presenta la Estimación de Costos para este proyecto, incluyendo los recursos materiales, así como también los humanos.

|  |  |
| --- | --- |
| **Producto** | **Costo Estimado** |
| Kit de Lego Mindstorms EV3 | $450.000 |
| Tarjeta SD (32 GB) | $8.000 |
| Dongle | $9.000 |
| Piezas de Lego extras | $20.000 |
| Equipos (Notebooks-4) | $3.000.000 |
| Software de desarrollo Python | Gratis |
| Librería EV3DEV 2 | Gratis |
| Sueldo Ing. Programador | $8.615/hora |
| Sueldo Ing. Proyectos | $8.000/hora |
| Sueldo Ensamblador | $5.769/hora |

# [Análisis y Diseño](#_heading=h.3j2qqm3)

## Especificación de Requerimiento

Requerimientos funcionales:

* El robot debe desplazarse al apretar un botón específico en la interfaz gráfica.
* El robot debe disparar piezas de lego al presionar un botón designado en la interfaz gráfica.
* El robot se debe conectar a un equipo mediante una conexión wifi.

Requerimientos no funcionales:

* Robot con estructura estable que al realizar las funciones solicitadas no presenta daños.
* Se debe programar la interfaz gráfica en lenguaje de programación Python.
* El robot debe estar construido con las piezas del Kit de lego Mindstorms ev3.

## Arquitectura

Diagrama

Descripción generada automáticamente

(1) Tanto el robot, como el equipo a ser utilizado deben estar conectados a la misma red wifi.

(2) Se debe iniciar el archivo del servidor para establecer la conexión remota.

(3) Se debe iniciar la Interfaz Gráfica para que el usuario pueda ejecutar las funciones del robot.

(4) Se establece la conexión del servidor del robot y la interfaz permitiendo que el usuario pueda usar la aplicación.

## Interfaz

A continuación, se muestra la Interfaz Gráfica, la cual contiene una única ventana principal. En esta ventana se encuentran los botones que ejercerán movimientos y sonidos en el robot.

Icono

Descripción generada automáticamente

* Movimiento: El movimiento de desplazamiento del robot está controlado por los botones **↑, ↓,** ←, →. Así al presionarlos el usuario puede controlar si el robot se mueve hacia delante, hacia atrás, a la derecha o izquierda.
* Inclinación: Con los botones▲, ▼ se puede controlar la inclinación la elevación a la que la torreta posteriormente disparará los proyectiles.
* Sonido: El botón con el símbolo **🔊**, es para que al presionarlo se reproduzca en el robot sonidos seleccionados que fueron incluidos en el código.
* Disparo: Al presionar este botón la torreta comenzará el disparo de proyectiles, expulsando la munición cargada previamente en el robot.
* Sensor: Con este botón se puede calcular a través del sensor la distancia que hay entre la torreta y el objetivo a disparar.
* Establecer Conexión: Al presionar estos botones se logra establecer o terminar la conexión entre el servidor y el robot.
* Manual: Este botón contiene un manual para facilitar al usuario con la información de cómo funciona el programa.

# Implementación

## [Fundamentos de Proyectiles](#_heading=h.3whwml4)

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

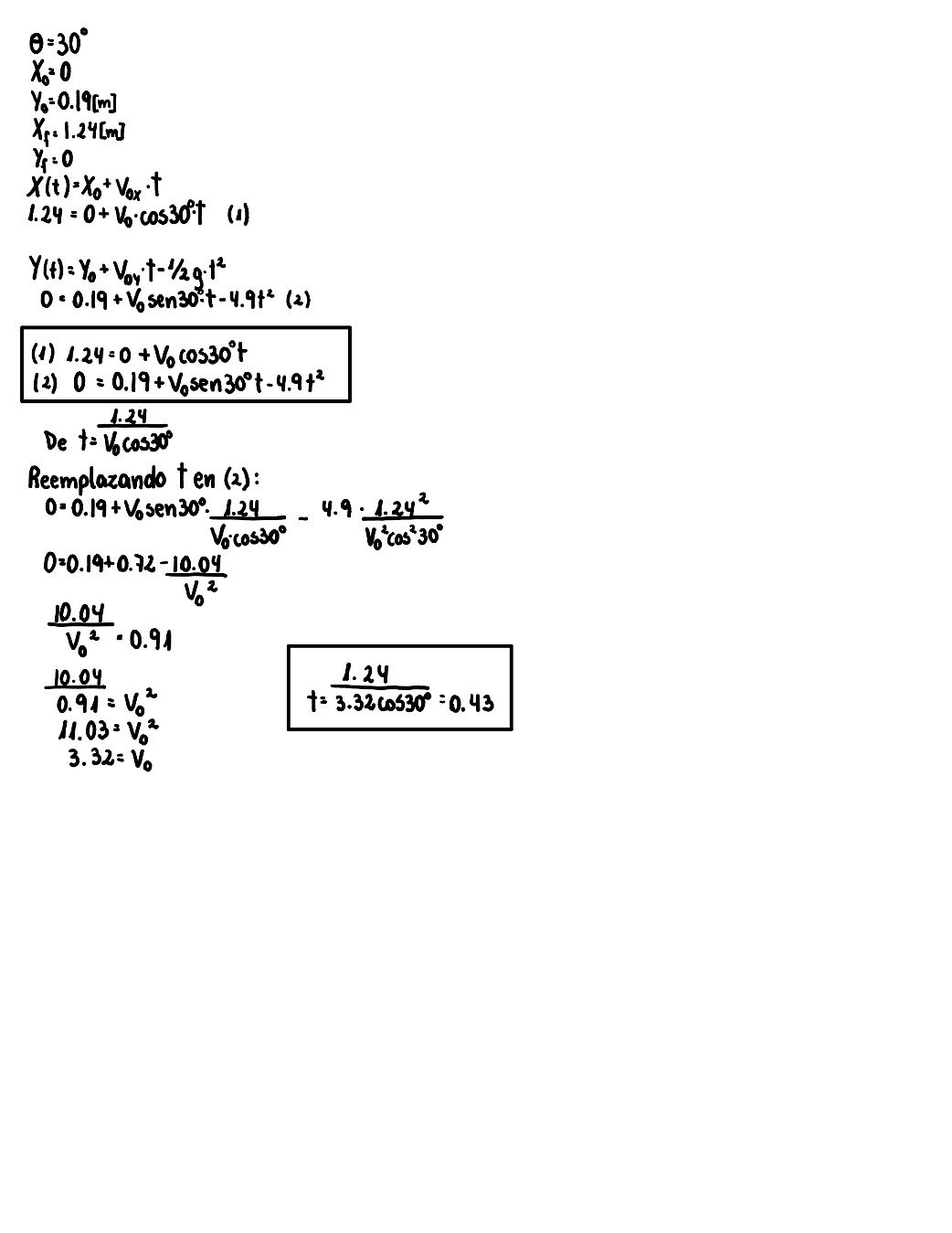
x = + ∗

= ∗ = cte

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

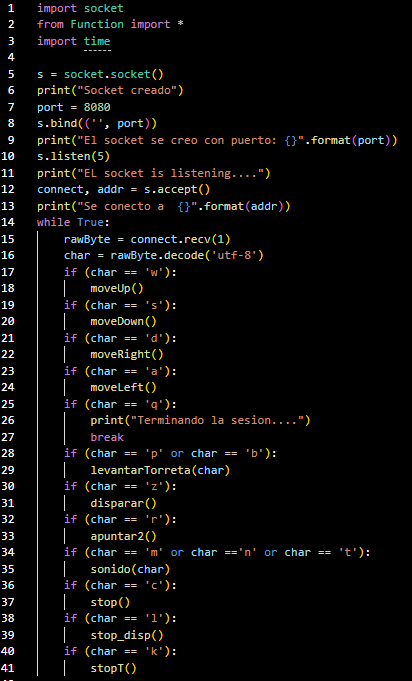
= + ∗ +∗ ∗

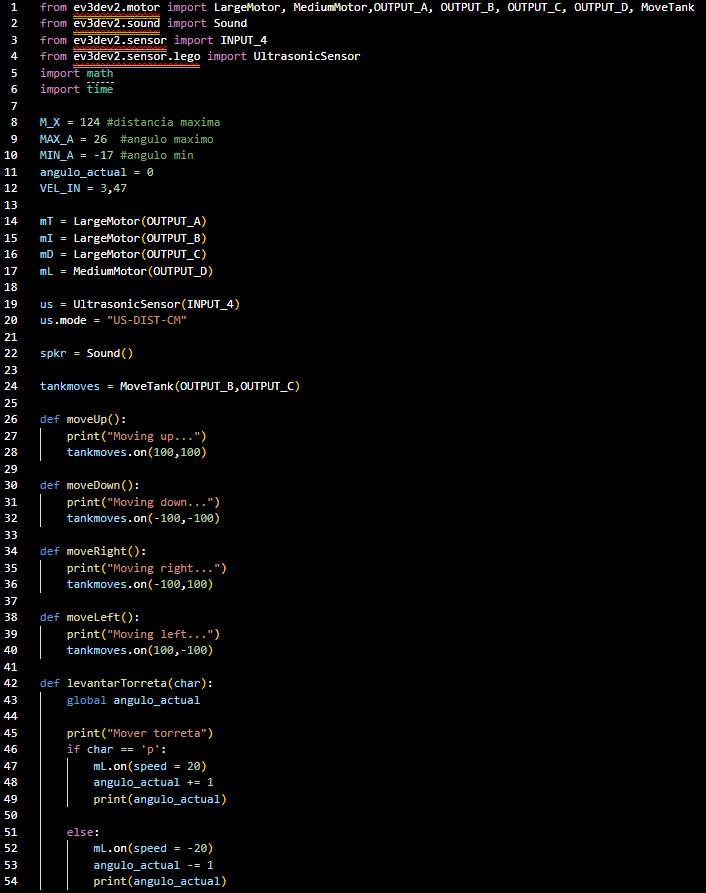
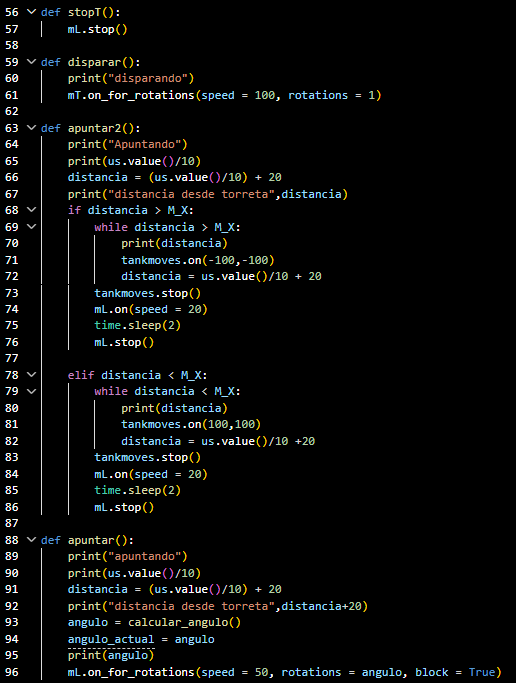
= ∗ − ∗



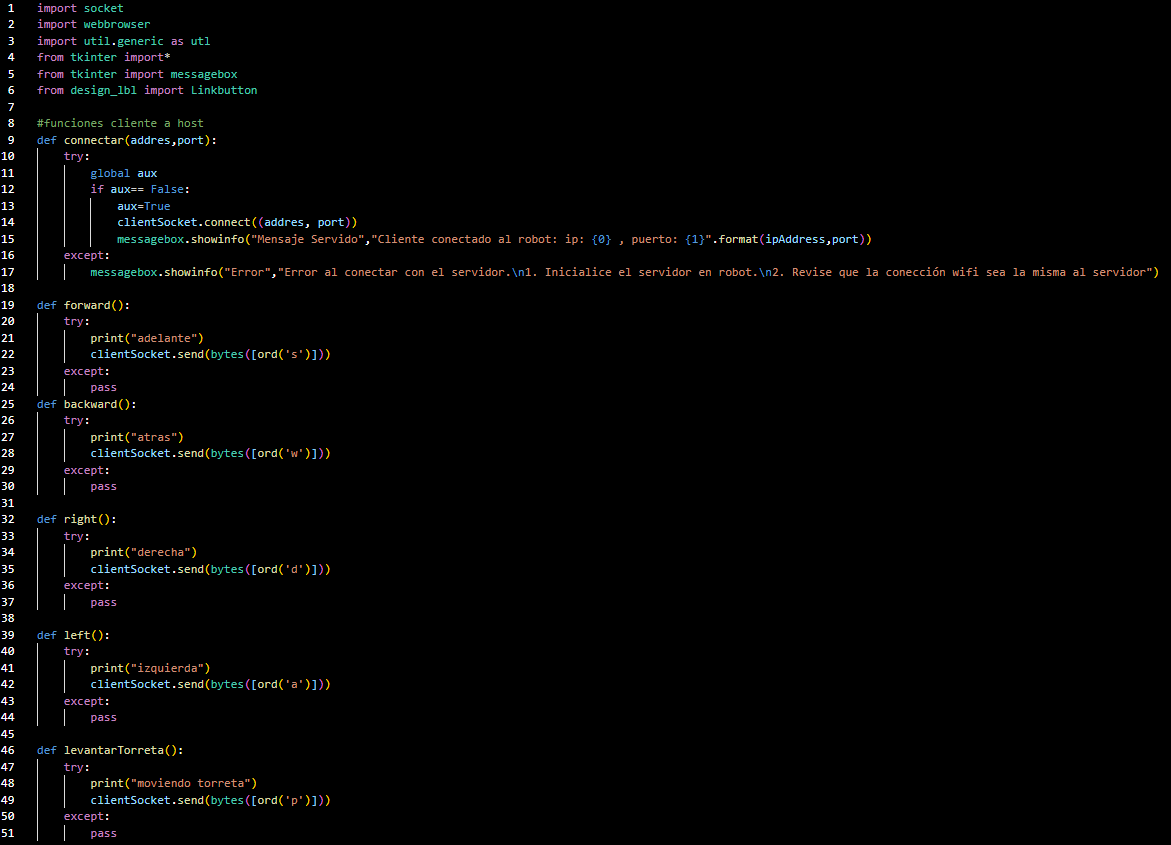
## [Descripción de los programas](#_heading=h.2bn6wsx)

* Visual Studio Code: Es un editor de código fuente. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.
* Microsoft Office: Es una suite ofimática de aplicaciones para computadoras de escritorio, servidores y servicios para los sistemas operativos Microsoft Windows, Mac OS X, iOS, Android y Linux.
* Drive: Es un servicio de alojamiento de archivos que guarda los datos de aquellos archivos en una nube.
* PuTTY: Es un emulador de terminal gratuito que admite varios protocolos de red tal como SSH, está disponible de licencia libre.



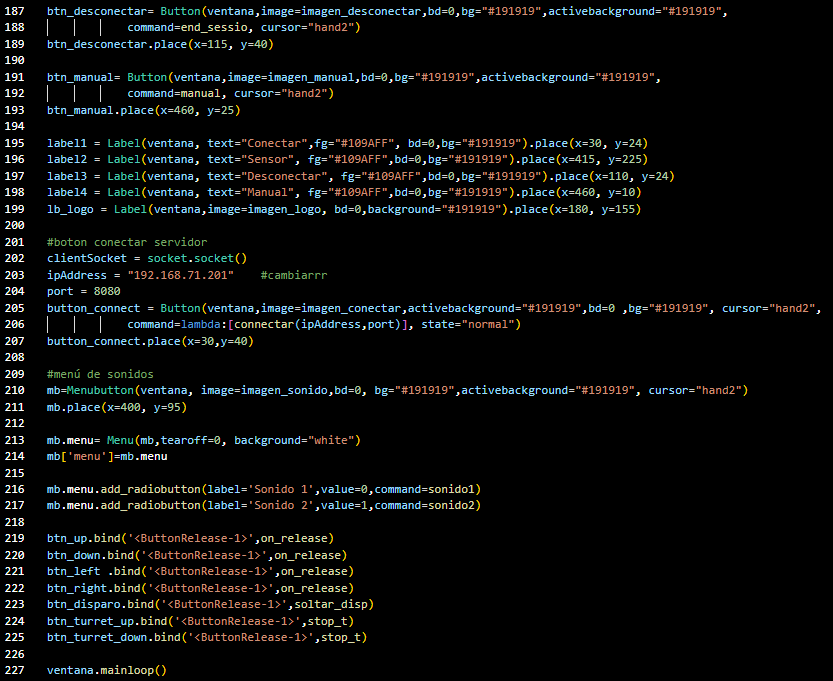




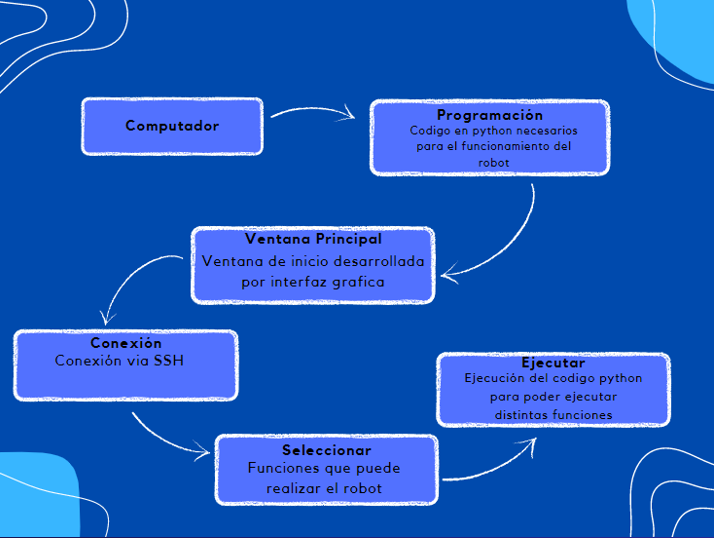








## Diagramas



# Resultados

## [Estado Final del Proyecto](#_heading=h.1pxezwc)

* La estructura del robot ya está en su versión final, es estable al desplazarse de un punto a otro, su inclinación de posicionamiento para realizar los disparos de la munición es correcta.
* La interfaz gráfica está finalizada y actualizada con los últimos detalles de diseño y funcionalidad.
* La wiki está actualizada, ya que se encuentran subidas todas las bitácoras, informes, fotos del proyecto e información del proyecto.
* El manual de usuario y el documento de los estándares de calidad está finalizado.

## [Problemas Encontrados y Solución Propuesta](#_heading=h.49x2ik5)

|  |  |
| --- | --- |
| **Problema** | **Solución** |
| Cumplir con las fechas de la Carta Gantt | Re-agendar las fechas de actividades programadas e ir avanzando de antemano con las que se pueda para compensar los plazos estipulados y no atrasar las entregas de avances. |
| Falta de Piezas | Buscar alternativas que cumplan las mismas funciones que las piezas que no están a disposición. |
| Estabilidad de la estructura | Buscar distintas piezas que aporten la estabilidad a la torreta para que, al realizar los momentos debidos, no se ocasione ningún percance. |

## Trabajo a Futuro

Se planea seguir mejorando la eficiencia de los códigos para lograr un programa más pulcro y legible que facilite a todo aquel programador que desee modificar y/o mejorar el programa actual.

Además, se busca idear e implementar más funciones que puedan aportar una mayor entretención a los usuarios de nuestro robot, a parte de su función principal que es el disparo de proyectiles a través de la torreta.

# Conclusiones

A lo largo del proceso de desarrollo del proyecto, la habilidad más reforzada fue el trabajo en equipo, trabajamos en escuchar todas las ideas de los integrantes para realizar el proyecto, y si surgían problemas se solucionaban de acuerdo con un consenso logrado por el equipo.

A nivel de programación, este proyecto permitió un mayor trabajo con el lenguaje de programación Python, junto con librerías, como la EV3DEV 2 y además el trabajo con la interfaz gráfica tkinter.

También tuvimos que aprender cómo realizar los distintos documentos solicitados que contienen una estructura específica a cumplir, tal como la carta Gantt, informes de avance y el manual de usuario.

Finalmente, en el aspecto laboral este trabajo permitió acercarnos más a las actividades que realizaremos en un futuro al ejercer nuestra carrera, mediante tener que crear y seguir una planificación para cumplir plazos específicos, además de tener que lograr objetivos solicitados con relación al robot y su documentación.

# Referencias (Estándar IEEE)

* PuTTY: Cliente gratuito que **permite el uso de protocolos de red Telnet**, rlogin, SSH o SCP con notables opciones de personalización. Está disponible en las plataformas Windows PC, Linux y macOS, permitiendo hacer conexiones a servidores remotos.

"Download PuTTY - a free SSH and telnet client for Windows". Download PuTTY - a free SSH and telnet client for Windows. https://www.putty.org/ (accedido el 14 de diciembre de 2022).

* Visual Code Studio: Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.  
  Microsoft. "Visual Studio Code - Code Editing. Redefined". Visual Studio Code - Code Editing. Redefined. https://code.visualstudio.com/ (accedido el 14 de diciembre de 2022).
* LEGO Technic Tower Defense – Instructions: Video que contiene las instrucciones con las que se creó la base del robot.

Thoroughfare production. LEGO Technic Tower Defense - Instructions. (30 de abril de 2020). Accedido el 14 de diciembre de 2022. [Video en línea]. Disponible: https://www.youtube.com/watch?v=uzBoo\_1Z8JM