**UNIVERSIDAD DE TARAPACÁ**

**FACULTAD DE INGENIERÍA**

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

Imagen que contiene Forma

Descripción generada automáticamente

**Plan de Proyecto**

**“Blitz”**

**Alumno(os): Cristóbal Hernández**

**Jhon Alarcón**

**Felipe Lira**

**Fernando Garrido**

**Ana Gutiérrez**

**Asignatura: Proyecto l**

**Profesor: Humberto Urrutia López**

**Arica, Septiembre 2024**

Historial De Cambios

| **Fecha** | **Versión** | **Descripción** | **Autor(es)** |
| --- | --- | --- | --- |
| 28/08/2023 | 1.0 | Concepción del Documento | Ana Gutiérrez |
| 06/09/2024 | 1.1 | Primer avance | Ana Gutiérrez  Jhon Alarcón |
| 30/09/2024 | 1.2 | Segundo avance | Ana Gutiérrez |

###### Tabla 1: Historial de cambios.

# 

# Tabla de Contenidos

[**1. Panel General 4**](#_heading=h.tyjcwt)

[1.1. Introducción 4](#_heading=h.3dy6vkm)

[1.2. Objetivos 4](#_heading=h.1ksv4uv)

[1.2.1. Objetivo General 4](#_heading=h.44sinio)

[1.2.2. Objetivos Específicos 4](#_heading=h.2jxsxqh)

[1.3. Restricciones 5](#_heading=h.3j2qqm3)

[1.4. Entregables 5](#_heading=h.1y810tw)

[**2. Organización del Personal 5**](#_heading=h.4i7ojhp)

[2.1. Descripción de los Roles 5](#_heading=h.2xcytpi)

[2.2. Personal que Cumplirá los Roles 5](#_heading=h.1ci93xb)

[2.3. Mecanismos de comunicación 6](#_heading=h.3whwml4)

[**3. Planificación del Proyecto 7**](#_heading=h.2bn6wsx)

[3.1. Actividades (Carta Gantt) 7](#_heading=h.qsh70q)

[3.2. Asignación de tiempo 7](#_heading=h.99xzomj1xd8y)

[3.3. Gestión de Riesgos 7](#_heading=h.1pxezwc)

[**4. Planificación de los Recursos 9**](#_heading=h.49x2ik5)

[4.1. Hardware 9](#_heading=h.2p2csry)

[4.2. Software 9](#_heading=h.147n2zr)

[4.3. Estimación de Costos 9](#_heading=h.3o7alnk)

[4.3.1 Costos de Hardware 9](#_heading=h.iyoaf2r5992)

[4.3.2 Costos de Software 10](#_heading=h.88k4infjmwwe)

[4.3.4 Costos de Gestión 10](#_heading=h.tbm08nwe3sfc)

[4.3.5 Costos Totales 10](#_heading=h.9evvu7copjzt)

[**5. Análisis y diseño 11**](#_heading=h.2qrv1fdoa7d)

[5.1. Especificación de requerimientos 11](#_heading=h.t6us68ytcs85)

[5.1.1. Requerimientos funcionales 11](#_heading=h.u5gn30q9fd5q)

[5.1.2. Requerimientos no funcionales 11](#_heading=h.3zlwj31grjnu)

[5.2 Arquitectura 11](#_heading=h.z8qxbom0om9k)

[5.3. Interfaz gráfica 12](#_heading=h.ul5m1edkifqf)

[**6. Implementación 12**](#_heading=h.55ml959utk4)

[6.1. Fundamentación 12](#_heading=h.2n5b19vbyziu)

[6.2. Descripción de los programas 12](#_heading=h.p2hbfry2jilo)

[6.3. Diagramas 12](#_heading=h.4g67kpntbxfv)

[**7. Resultados 12**](#_heading=h.exdck5sb5iph)

[7.1. Estado actual del proyecto 12](#_heading=h.udxys52eap3o)

[7.2. Problemas encontrados y solución propuestas 12](#_heading=h.f19xi65r0ahp)

[**5. Conclusión 12**](#_heading=h.1hmsyys)

[**6. Referencias 13**](#_heading=h.41mghml)

# Índice de tablas

1[. Tabla 1: Historial de cambios](#_heading=h.771cchp7qj5s) 2

2[. Tabla 2: Roles](#_heading=h.rq4z4341k50u) 6

3[. Tabla 3: Riesgos](#_heading=h.pkkxyvybfapk) 9

4[. Tabla 4: Costos Hardware 1](#_heading=h.mxyy3kab3ze9)0

5[. Tabla 5: Costos Software 1](#_heading=h.bo0noyhyee5k)0

[6. Tabla 6: Costos Gestión](#_heading=h.sem36pa1hdyr) 10

7[. Tabla 7: Costos totales](#_heading=h.41mghml) 11

8[. Tabla 8: Problemas y soluciones](#_heading=h.41mghml) 19

# Índice de figuras

1[. Ilustración 1: Carta Gantt](#_heading=h.b52xnsw66z5r) 7

2[. Ilustración 2: Arquitectura 1](#_heading=h.sq41zufpg32n)2

3[. Ilustración 3: Interfaz conexión. 1](#_heading=h.3xwik427tvnl)3

4[. Ilustración 4: Interfaz botones 1](#_heading=h.xfle19aqm01r)3

5[. Ilustración 5: Interfaz teclado 1](#_heading=h.gpnnljhntpex)4

6. [Ilustración 6: Interfaz joystick 1](#_heading=h.j4gxbey1eh8a)5

7. [Ilustración 7: server.py](#_heading=h.skly1niecw7j) [1](#_heading=h.j4gxbey1eh8a)6

8. [Ilustración 8: library.py 1](#_heading=h.5tw08qgi3uba)7

6[. Ilustración 8: Diagrama 1 1](#_heading=h.20not3fhdtpb)8

7[. Ilustración 9: Diagrama 2 1](#_heading=h.idwidqfdba2z)8

8[. Ilustración 10: Diagrama 3](#_heading=h.94uj06z5u5l2) 18

# 

# 1. Panel General

## 1.1. Introducción

Durante el transcurso del semestre, se ha planteado la tarea de realizar, en equipo, el proyecto "Blitz" que consiste en el diseño, construcción y programación de un robot. Utilizando el Lego Mindstorm Ev3 Core Set, se espera que este sea capaz de seguir instrucciones que le permitan desplazarse, así como recoger y soltar una pelota de ping pong.

## 1.2. Objetivos

### 1.2.1. Objetivo General

Desarrollar un robot capaz de seguir instrucciones a través de una interfaz gráfica codificada en Python, con la finalidad de desplazarse, recoger y dejar una pelota de ping pong.

### 1.2.2. Objetivos Específicos

* Investigar las herramientas y funcionalidades del Lego Mindstorm EV3.
* Ensamblar las piezas siguiendo un diseño que permita al robot desplazarse y realizar sus funciones de manera estable.
* Estudiar las librerías necesarias para la programación del robot y la interfaz gráfica.
* Desarrollar e implementar el software que permita dar instrucciones al robot.
* Evaluar el funcionamiento del robot.

## 1.3. Restricciones

* El lenguaje de programación a utilizar debe ser Python.
* Se debe usar el sistema operativo linux.
* El registro de los avances del proyecto, se debe realizar en Redmine.
* Para ensamblar el robot, sólo se pueden usar piezas disponibles.
* El tiempo destinado para el desarrollo del proyecto es limitado.
* La disponibilidad del robot está restringida.
* El equipo debe estar conformado por un máximo de 5 personas.

## 1.4. Entregables

* Bitácoras: Contendrán información respecto al avance semanal del proyecto, los problemas por resolver y las tareas a realizar en la próxima semana.
* Informe: Incluirá el progreso detallado del proyecto, su fundamentación y planificación.
* Presentación: Mostrará información respecto al avance y formulación del proyecto.
* Carta Gantt: Incluirá la organización temporal del proyecto.
* Redmine: Contendrá información sobre la organización, fundamentación y planificación del proyecto.

# 2. Organización del Personal

## 2.1. Descripción de los Roles

* Jefe de Proyecto: Es el representante del equipo, responsable de la supervisión y organización del proyecto.
* Ensamblador: Está a cargo del diseño y construcción del robot.
* Programador: Se encarga de la codificación e implementación del software necesario para el funcionamiento del robot.
* Documentador: Es responsable de la redacción de bitácoras, informes, presentaciones y documentación en la wiki.
* Diseñador: Se encarga del diseño visual y gráfico del proyecto.

## 2.2. Personal que Cumplirá los Roles

| **Rol** | **Responsable** | **Involucrados** |
| --- | --- | --- |
| Jefe de proyecto | Cristóbal Hernández | Cristóbal Hernández |
| Ensamblador | Fernando Garrido | Felipe Lira |
| Programador | Jhon Alarcón | Cristóbal Hernández |
| Documentador | Ana Gutiérrez | Jhon Alarcón |
| Diseñador | Ana Gutiérrez | Cristóbal Hernández |

###### Tabla 2: Roles.

## 2.3. Mecanismos de comunicación

Se utilizan dos medios de comunicación: Whatsapp y Discord. WhatsApp permite la mensajería instantánea y formación de grupos, lo que facilita la comunicación rápida. Por otro lado, Discord se emplea para realizar llamadas y mantener discusiones en tiempo real mediante sus canales de voz y texto, siendo ideal para reuniones virtuales.

# 3. Planificación del Proyecto

## 3.1. Actividades (Carta Gantt)

###### Ilustración 1: Carta Gantt.

## 3.2. Asignación de tiempo

La planificación del tiempo para el proyecto es la siguiente:

* Planificación del proyecto: 2 a 3 semanas.
* Ejecución del proyecto: 5 a 6 semanas.
* Finalización del proyecto: 16 semanas.

## 3.3. Gestión de Riesgos

La siguiente tabla indica los riesgos que se pueden presentar durante el proyecto. Estos serán descritos, se les asignará una probabilidad de ocurrencia, se definirá una acción remedial y se establecerá un nivel de impacto, siguiendo los siguientes niveles de impacto:

1. Daño catastrófico: Requiere resolución inmediata, ya que puede hacer que el proyecto se detenga durante un tiempo indeterminado.
2. Daño crítico: Debe solucionarse de inmediato, puesto que puede retrasar el proyecto varias etapas.
3. Daño circunstancial: Es necesario resolver el riesgo a la brevedad, ya que puede causar un retraso en una etapa del proyecto.
4. Daño irrelevante: Puede resolverse en cualquier momento..

| **Riesgo** | **Probabilidad de Ocurrencia** | **Nivel de Impacto** | **Acción Remedial** |
| --- | --- | --- | --- |
| Ausencia del personal | 50% | 3 | Reorganizar al equipo para avanzar en las tareas del personal ausente. |
| Reconstrucción total del robot por no cumplir lo requerido. | 20% | 1 | Buscar nuevas ideas que cumplan con lo pedido y llevarlas a cabo. |
| Incumplimiento de tareas. | 40% | 2 | Reorganización priorizando las tareas atrasadas |
| Retrasos en la entrega  de materiales | 30% | 2 | Anticiparse a las necesidades y asegurar un margen de tiempo suficiente para la llegada de materiales. |
| Errores en el código | 50% | 3 | Implementar pruebas y revisiones de código para detectar y corregir errores. |
| Sobrecarga de trabajo en los miembros del equipo | 35% | 3 | Redistribuir las tareas y priorizar las más críticas para aliviar la carga de trabajo de los miembros. |
| Escasez de piezas | 40% | 3 | Realizar pedidos de piezas al personal encargado |
| Daño en la tarjeta SD | 10% | 1 | Cambiar y configurar la tarjeta SD |
| Equipo defectuoso | 10% | 1 | Solicitar otro equipo al personal encargado |
| Baja batería del EV3 | 10% | 1 | Recargar el EV3 |

###### Tabla 3: Riesgos.

# 4. Planificación de los Recursos

## 4.1. Hardware

* Kit LEGO MINDSTORMS EV3.
* Notebook con sistema operativo Linux.
* Wi-Fi Dongle USB.
* Micro SD.
* Router.

## 4.2. Software

* Licencia de Microsoft Office: Utilizada para la documentación.
* Visual Studio Code: Empleado para el desarrollo en Python.
* Discord: Herramienta utilizada para la comunicación del equipo.
* Canva: Utilizado para diseño gráfico y presentaciones.

## 4.3. Estimación de Costos

A continuación, se presentan los costos asociados al hardware, software y la gestión del personal:

## 4.3.1Costos de Hardware

| **Productos** | **Cantidad** | **Valor unitario** | **Valor total** |
| --- | --- | --- | --- |
| Kit Lego MINDSTORMS (EV3) | 1 | $1.229.990 | $1.229.990 |
| Dongle USB Wifi | 1 | $8.990 | $8.990 |
| Micro SD | 1 | $5.000 | $5.000 |
| Notebook Ubuntu | 1 | $899.990 | $899.990 |

###### Tabla 4: Costos Hardware.

## 4.3.2Costos de Software

| **Productos** | **Cantidad** | **Valor mensual** | **Valor total (4 meses)** |
| --- | --- | --- | --- |
| Visual Studio Code | 3 | Gratuito | Gratuito |
| Discord | 5 | Gratuito | Gratuito |
| Canva | 5 | Gratuito | Gratuito |

###### Tabla 5: Costos Software.

## 4.3.4 Costos de Gestión

*Sueldos:*

| **Rol** | **Horas trabajadas** | **Horas extra trabajadas** | **Valor por hora trabajada** |
| --- | --- | --- | --- |
| Jefe de proyecto | 66 horas | 14 horas | $29.500 |
| Programador | 66 horas | 16 horas | $25.000 |
| Ensamblador | 66 horas | 14 horas | $22.500 |
| Diseñador | 66 horas | 11 horas | $21.000 |
| Documentador | 66 horas | 13 horas | $22.000 |
| Total : | - | - | $9.565.000 |

###### Tabla 6: Costos Gestión.

## 4.3.5 Costos Totales

*Total de Costo:*

| Costo Hardware | $2.143.970 |
| --- | --- |
| Costo Software | $0 |
| Costo Empleados | $9.565.000 |
| Total : | $11.708.970 |

###### Tabla 7: Costos totales.

# 5. Análisis y diseño

## 5.1. Especificación de requerimientos

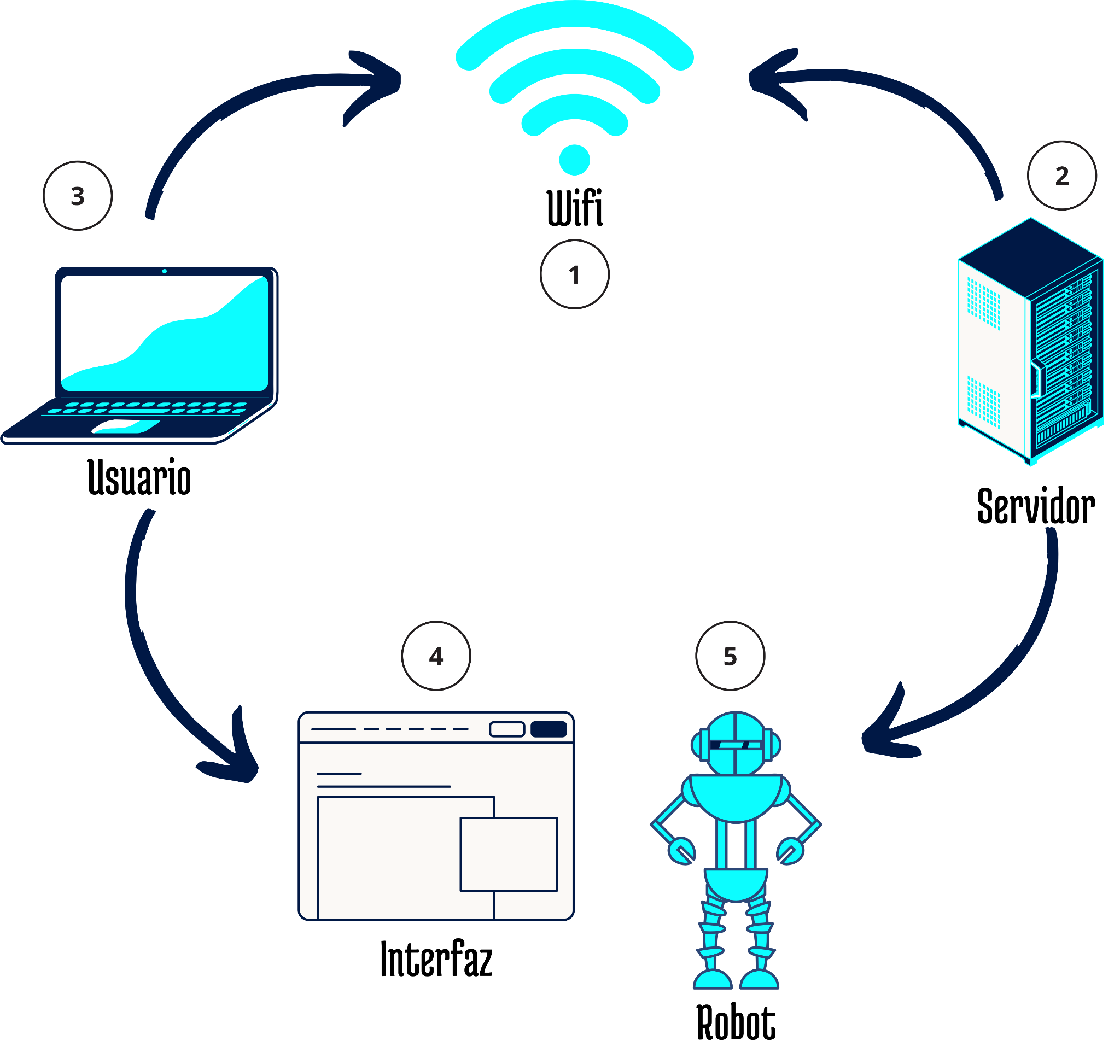
### 5.1.1. Requerimientos funcionales

* El robot debe poder moverse en todas las direcciones.
* El robot debe ser capaz de recoger una pelota de ping pong y dejarla.
* El robot debe poder ser controlado por el usuario a través de una interfaz.

### 5.1.2. Requerimientos no funcionales

* La interfaz gráfica debe ser fácil de usar y responder en menos de 2 segundos.
* El manual de usuario debe ser detallado y claro.
* El robot debe funcionar por al menos 3 horas sin necesidad de recargar.

## 5.2 Arquitectura



###### Ilustración 2: Arquitectura.

1. El notebook y el robot deben estar conectados a la misma red de WiFi.
2. Este se conecta al robot, se comunica con él y recibe los comandos enviados por el usuario a través de la interfaz.
3. El usuario se conecta al robot mediante el servidor y envía instrucciones a través de la interfaz.
4. Se utiliza para el control del robot.
5. Ejecutará acciones según las instrucciones del usuario.

## 5.3. Interfaz gráfica

Al presionar el botón de encendido en la ventana principal, se abre la siguiente ventana, que permite establecer la conexión con el robot mediante su dirección IP. Es posible conectar utilizando una IP previamente guardada, haciendo clic en el botón 'Ingresar IP guardada' y luego en 'Conectar'. Alternativamente, también se puede ingresar manualmente la IP deseada y presionar 'Conectar' para establecer la conexión.



###### Ilustración 3: Interfaz conexión.

La ventana principal cuenta con un botón de encendido y apagado, que, al presionarlo, permite la conexión del robot mediante la IP, como se mencionó anteriormente, y da acceso a las funciones de control del robot. Existen tres métodos para que el robot genere las acciones correspondientes:

* Botones:
  + Las flechas en la sección “Robot” permiten mover al robot hacia adelante, atrás, izquierda y derecha.
  + Las flechas en la sección “Garra” permiten que el brazo de la garra suba y baje, mientras que los botones de la misma sección permiten abrir y cerrar la garra.



###### Ilustración 4: Interfaz botones.

* Teclado:
  + Las teclas en la sección “Robot” permiten mover al robot hacia adelante, atrás, izquierda y derecha.
  + Las teclas L y K , en la sección “Garra”, permiten que el brazo de la garra suba y baje, mientras que las teclas X y C de la misma sección permiten abrir y cerrar la garra.



###### Ilustración 5: Interfaz teclado.

* Joystick:
  + La palanca analógica permite mover el robot en las direcciones adelante, atrás, izquierda y derecha, y el movimiento realizado se refleja en la sección "Robot" de la interfaz.
  + Los botones L2 y R2 controlan el movimiento vertical del brazo de la garra, permitiendo que suba y baje, mientras que los botones A y B permiten abrir y cerrar la garra.

###### 

###### Ilustración 6: Interfaz joystick.

# 6. Implementación

## 6.1. Fundamentación

### Torque máximo requerido

El torque depende del grado de orientación del brazo "θ":

**T = Ftotal · r · sin(θ)**

Donde **Ftotal = (mgarra + mcarga) · g**:

**T = (mgarra + mcarga) · g · r · sin(θ)**

Sustituyendo para el torque máximo en 90°:

**Tmax = 2.242 N · 0.212 m · 1 = 0.475 Nm**

Todo, incluyendo el peso de la garra, tendría un impacto significativo en el torque necesario: de **0.0056 Nm**, que corresponde solo a la carga, a **0.475 Nm** con la carga y la garra incluidas.

### Velocidad promedio

Fórmula de velocidad:

**v = d / t**

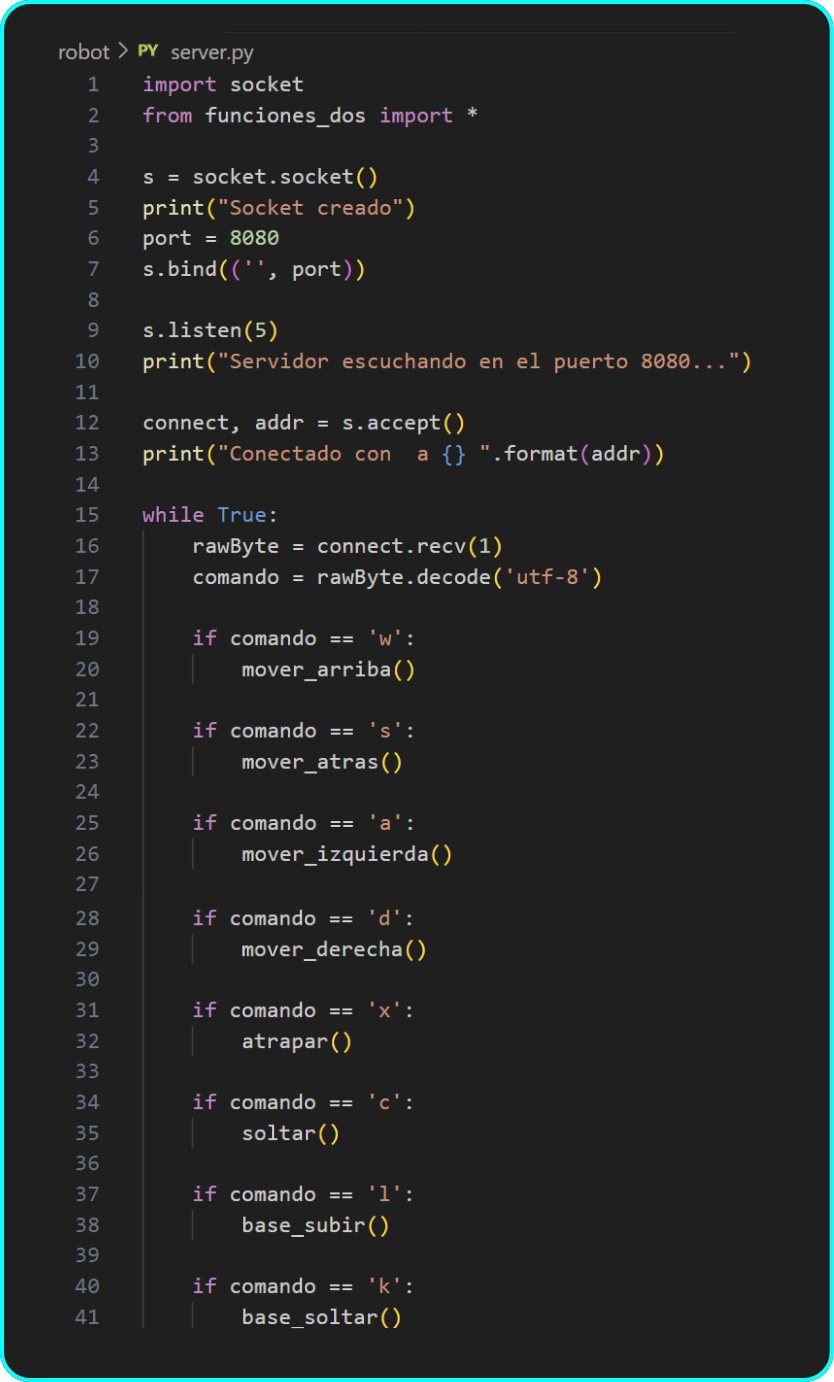
Reemplazando:

**v = 1 m / 3.5 s = 0.286 m/s**

## 6.2. Descripción de los programas

### 6.2.1. Servidor (server.py)

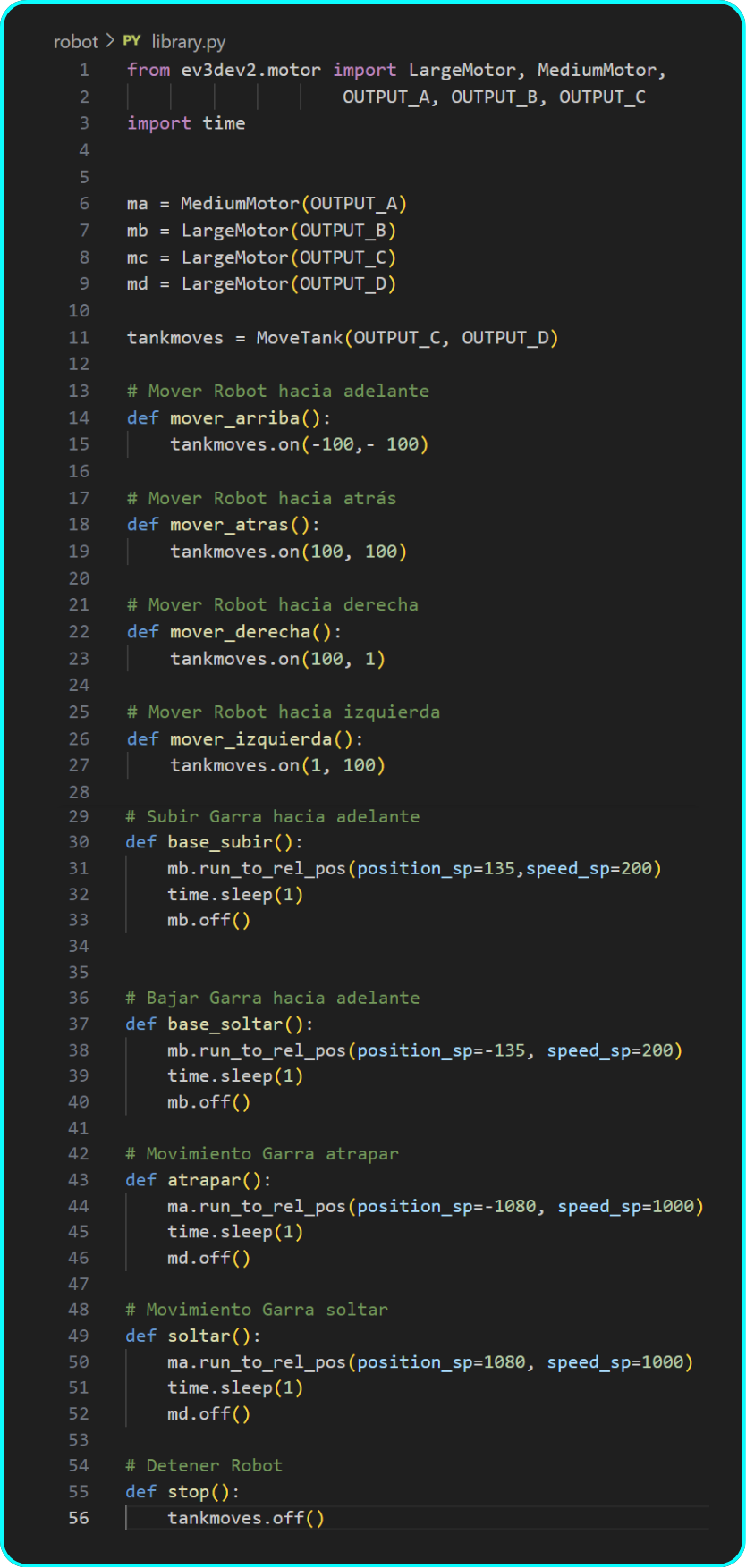
Es el programa encargado de conectar el robot con la interfaz que permite al usuario controlarlo, mediante un servidor creado con la librería socket de Python.



###### Ilustración 7: server.py.

### 6.2.2. Librería de funciones (library.py)

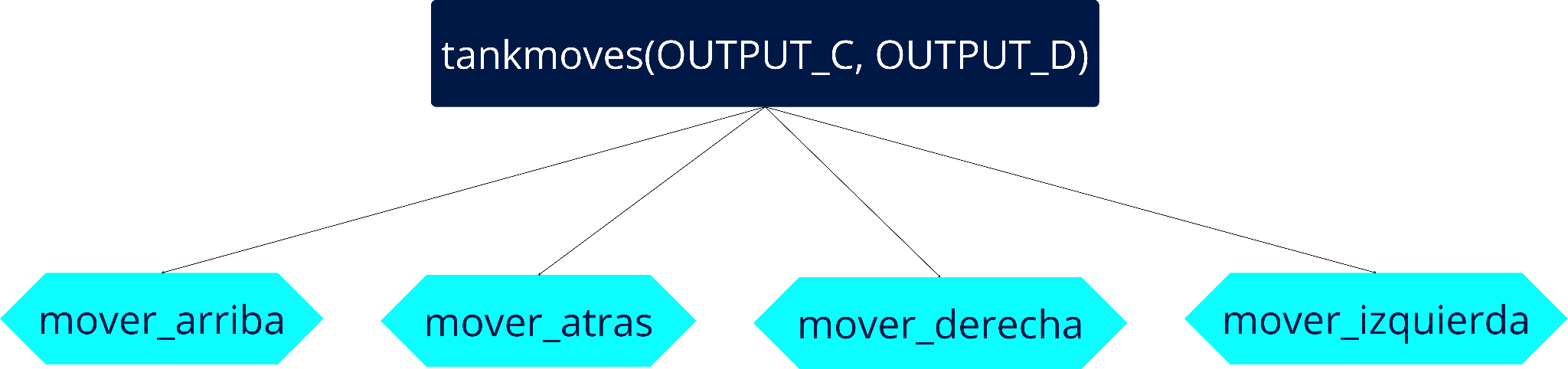
Contiene las funciones que definen las acciones que el robot puede ejecutar.



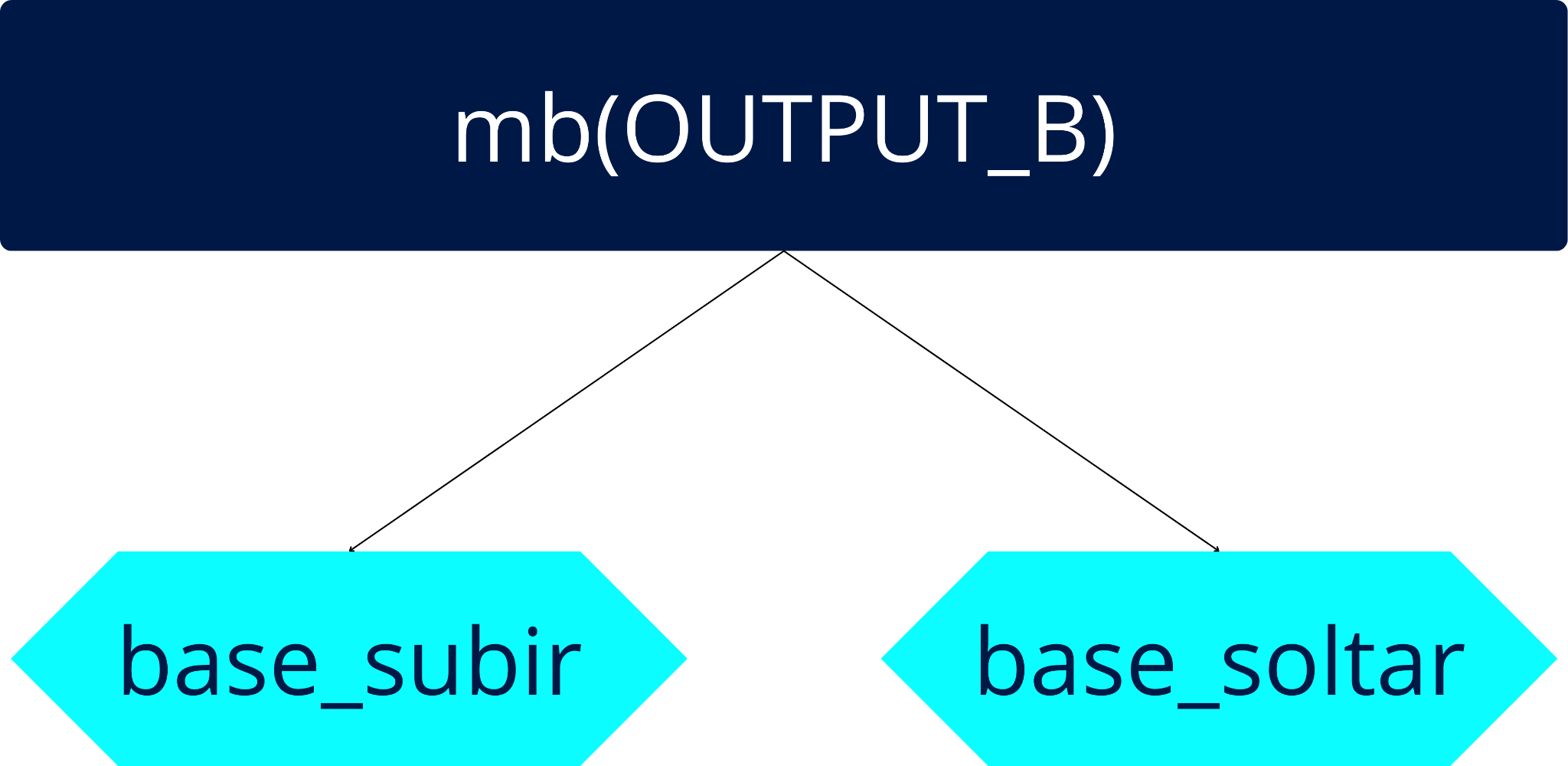
###### Ilustración 8: library.py.

## 6.3. Diagramas

Los siguientes diagramas describen los movimientos del robot.



###### Ilustración 9: Diagrama 1.



###### Ilustración 10: Diagrama 2.



###### Ilustración 11: Diagrama 3.

# 7. Resultados

## 7.1. Estado actual del proyecto

* Ensamblaje del robot: La construcción del robot está completa.
* Funciones de movimiento del robot: El robot puede desplazarse en todas las direcciones, así como recoger y soltar una pelota de ping pong.
* Interfaz: Desarrollada con tkinter y completamente finalizada, permitiendo la interacción del usuario con el robot.
* Servidor: Desarrollado usando la librería “socket” de Python e implementado en el entorno EV3.
* Conexión remota funcional.
* Documentación del proyecto: Wiki del proyecto actualizada.
* Carta Gantt: Planificación del proyecto actualizada.
* Bitácoras, informes y presentaciones: Generados para la documentación y presentación del proyecto.

## 7.2. Problemas encontrados y soluciones propuestas

| Problemas encontrados | Soluciones propuestas |
| --- | --- |
| Falta de materiales | Se contactó al responsable de la distribución de piezas para solicitar las piezas faltantes. |
| Diseño del robot inestable | Se reconstruyó el robot para cumplir con los requisitos de estabilidad, reforzando los puntos débiles. |
| Movimientos del robot | Se corrigieron las incongruencias entre el código y el diseño de la garra del robot. |
| Planificación errónea | Se reajustaron los tiempos para las actividades en la Carta Gantt. |

###### Tabla 8: Problemas y soluciones.

# 

# 5. Conclusión

En esta segunda etapa, el proyecto planteó desafíos técnicos y organizativos que impulsaron el crecimiento del equipo. El diseño del robot, logrado tras repetidos ajustes, y la programación, especialmente la integración del servidor y la interfaz, requirieron investigación.

El uso de herramientas como Redmine para documentar y planificar fue clave para mantener el control del proyecto, fomentando responsabilidad y accesibilidad a la información del proyecto. La comunicación efectiva, la asignación de roles y el cumplimiento de plazos resultaron esenciales para avanzar de manera organizada.

Este proceso no solo fortaleció las habilidades técnicas del equipo, sino también las capacidades de trabajo en equipo, gestión y resolución de problemas, preparándolo mejor para futuros retos.

# 6. Referencias

[1] LEGO Mindstorms EV3. (2023). LEGO.com. <https://www.lego.com/en-us/themes/mindstorms>

[2] Python Programming for EV3. (n.d.). EV3Lessons.com. <https://ev3lessons.com/en/lessons.html#python>

[3] LEGO Mindstorms EV3 Support. (n.d.). LEGO Education. <https://education.lego.com/en-us/support/mindstorms-ev3>

[4] Setting Up Wi-Fi on EV3. (2022). Dexter Industries. <https://www.dexterindustries.com/howto/setup-wifi-on-lego-mindstorms-ev3/>

[5] Programming LEGO EV3 with Python. (n.d.). STEM Robotics. <https://stemrobotics.cs.pdx.edu/node/5383>