

UNIVERSIDAD DE TARAPACÁ



## FACULTAD DE INGENIERÍA

Departamento de Ingeniería en Computación e Informática



### Informe Final RoboCleaner

Lego EV3 RoboCleaner

**Autor(es):** Joaquín Guarachi

Juan Yampara

Hernán Vazque

Benjamin Varas

Bairon Núñez

**Asignatura:** Proyecto I

**Profesor(es):** Leonel Alarcón

ARICA, 15 DE DICIEMBRE 2022.

# 1 Panorama general de proyecto

## 1.1 Modificaciones de documento

<b>Título</b>	<b>Informe Final de proyecto</b>
<b>Versión Final</b>	2.7

### 1.1.1. Historial de cambios: Informe de formulación

<b>Fecha</b>	<b>Versión</b>	<b>Descripción</b>	<b>Autor(es)</b>
01/09/2022	1.0	Versión preliminar de formato Informe	Benjamin. V
06/06/2022	1.1	Revisión y correcciones	T. Integrantes
20/09/2022	1.2	Cambios en secciones	T. Integrantes
01/10/2022	1.3	Modificación y arreglos	T. Integrantes
04/10/2022	1.4	Corrección sección: planificación de recursos	T. Integrantes
06/10/2022	1.5	Cambios sección: planificación de recursos	T. Integrantes

## 1.2. Historial de cambios: Informe de avance I

<b>Fecha</b>	<b>Versión</b>	<b>Descripción</b>	<b>Autor(es)</b>
11/10/2022	1.6	Avance en nuevas secciones de informe	Benjamin. V
18/10/2022	1.7	Avance y corrección en nuevas secciones de informe	Benjamin. V
20/10/2022	1.8	Corrección de secciones	Benjamin. V
21/10/2022	1.9	Ultimo avance de Informe I	Benjamin. V
22/10/2022	2.0	Corrección ortográfica y gramática Informe I	Bairon. N
25/10/2022	2.1	Cambios de descripción de secciones	Hernán. V

## 1.3. Historial de cambios: Informe Final

<b>Fecha</b>	<b>Versión</b>	<b>Descripción</b>	<b>Autor(es)</b>
22/11/2022	2.2	Avance en secciones de Informe Final	Benjamin. V
24/11/2022	2.3	Avance y corrección de secciones finales	Benjamin. V
06/12/2022	2.4	Avance en secciones	Benjamin. V
08/12/2022	2.5	Modificación de secciones	Benjamin. V
13/12/2022	2.6	Corrección	Benjamin. V
18/12/1011	2.7	Última modificación	T. Integrantes

## 1.2 Índice de contenidos

1	Panorama general de proyecto.....	2
1.1	Modificaciones de documento.....	2
1.2	Índice de contenidos.....	4
1.3	Introducción.....	5
1.4	Objetivo General.....	6
1.5	Objetivos específicos.....	6
1.6	Restricciones.....	6
1.7	Entregables.....	7
2	Organización de personal.....	8
2.1	Descripción de roles.....	8
2.2	Gestión de personal.....	8
2.3	Mecanismos de comunicación.....	8
3	Planificación de proyecto.....	9
3.1	Actividades.....	9
3.2	Asignación de tiempo.....	11
3.3	Gestión de riesgos.....	12
4	Planificación de recursos.....	13
4.1	Recursos de hardware y software.....	13
4.2	Estimación de costos.....	14
5	Análisis y diseño.....	15
5.1	Especificación de requerimiento.....	15
5.2	Arquitectura del robot.....	15
5.3	Interfaz y diseño.....	17
6	Implementación.....	18
6.1	Fundamentos de proyectiles.....	18
6.2	Descripción de programas.....	19
6.3	Diagramas.....	21
7	Pruebas.....	22
7.1	Descripción de pruebas realizadas.....	22
7.2	Resultados de pruebas.....	22
8	Resultados.....	23
8.1	Estado final del proyecto.....	23
8.2	Problemas encontrados y soluciones propuestas.....	23
9	Conclusiones.....	24
9.1	Conclusiones generales.....	24
9.2	Trabajo futuro.....	24
10	Referencias.....	25
11	Anexos.....	26

## 1.3 Introducción

El siguiente informe tratara sobre la elaboración del robot LEGO Robo Cleaner usando el Kit LEGO Mindstorms Education, cuya funcionalidad estará determinada en que gracias a los sensores el robot sea capaz de determinar una ruta y que el mismo sea capaz de derribar obstáculos que se le puedan establecer en su trayecto.

El proyecto constara de varias etapas de desarrollo, las cuales se regirán por la tabla de actividades y roles asignados, las actividades estarán seguidas por documentación en bitácora. Finalmente, entre los objetivos planteados para la elaboración del proyecto, principalmente se considera la adecuada aplicación de los conceptos básicos de la matemática, física e ingeniería informática para concluir y desarrollar las funcionalidades del robot, además de cumplir con los estándares solicitados para el proyecto.

## 1.4 Objetivo General

Aplicar conceptos de la ingeniería, física y matemática para implementar correctamente las funcionalidades del robot.

## 1.5 Objetivos específicos

Entre los objetivos específicos planteados en la elaboración del proyecto, se consideran los siguientes:

1. Indagar la documentación de las herramientas utilizadas para la eficiencia del proyecto.
2. Diseñar el funcionamiento interno y externo del robot para la articulación del mismo.
3. Implementar mantenimiento del robot para su funcionamiento.
4. Realizar prueba de la implementación del diseño para determinar su buen funcionamiento.
5. Comunicar avances con el cliente con fin de desarrollar adecuadamente el producto.

## 1.6 Restricciones

Las delimitaciones que se presentan o presentarían en la elaboración están dadas por:

- Estructura del kit Lego Mindstorms
- Limitación tecnológica en los sensores proporcionados del kit
- Tiempo en la dedicación al proyecto
- Problemas de salud que involucre a un integrante

## 1.7 Entregables

Los productos a entregar durante y al término de la elaboración del proyecto son los siguientes.

### **Primera Entrega Formulación de proyecto:**

1. Informe de formulación.
2. Bitácora (seguimiento de proyecto).
3. Presentación de formulación del proyecto.
4. Armado de robot.

### **Segunda Entrega Avance I de proyecto**

1. Implementación interna de movimientos del robot (código).
2. Programación de interfaz.
3. Informe de avance I.
4. Presentación de avance I.
5. Bitácora (seguimiento del proyecto)

### **Tercera Entrega Terminó de proyecto**

1. Implementación del lanzamiento del robot (código).
2. Integración y robot funcional.
3. Informe final.
4. Presentación final.
5. Terminó de bitácora de seguimiento.

## 2 Organización de personal

### 2.1 Descripción de roles

- **Project Manager:** Encargado de la administración y gestión del proyecto.
- **Analista de programación:** Encargado de la programación y análisis de solución informática.
- **Social Media Manager:** Encargado en la gestión de actividades en redes sociales.
- **Administración:** Encargado en la redacción de actividades de avance.

### 2.2 Gestión de personal

El grupo está conformado y destituido en los siguientes roles:

- **Hernán Vazque** (jefe, Project Manager)
- **Joaquín Guarachi** (rol: Social Media Manager)
- **Juan Yampara** (rol: Social Media Manager)
- **Benjamín Varas** (rol: Administración)
- **Bairon Núñez** (rol: Analista de programación)

### 2.3 Mecanismos de comunicación

Entre los mecanismos de comunicación que se establecen se consideran las siguientes plataformas.

- **Telegram:** Retroalimentar sobre avances de cliente a personal
- **WhatsApp:** Comunicación y organización del personal respecto a las actividades del proyecto.
- **Gmail:** Entrega de avances del proyecto
- **Discord:** Reunión del personal con fin de informar y avanzar actividades.



### 3 Planificación de proyecto

#### 3.1 Actividades

A continuación se mostrará el programa de actividades del proyecto. (Primera tabla)

Actividades durante el proyecto				
Semana	Nombre	Descripción	Responsables	Producto
1	Formulación del proyecto	Reconocimiento de KIT LEGO EV3	Todo el grupo	Avance en formulación del proyecto
2	Avance I en la formulación del proyecto	Construcción del robot, elaboración de bitacora		Avance en la construcción del robot
3	Avance II en la formulación del proyecto	Avance en armado, estudio librería EV3		Avance en armado
4	Termino de armado de robot, elaboración de informe	Termino de armado, elaboración informe, presentación	Bitacora: Joaquin. G Armado: Juan. Y, Hernan Informe: Benjamin. V	Termino de armado y comienzo de informe de formulación
5	Termino de informe y presentación	Termino de elaboración de informe y presentación	Documentación: Joaquin Programador: Juan. Y, Hernan. V. Informe: Benjamin V	Termino de informe y presentación
6	Desarrollo del robot	Pruebas de programación de robot	Todo el grupo	Avance en el funcionamiento del robot
7	Programación de movimientos del robot	Prueba de movimientos del robot	Programador: Hernan. V, Bairon. N. Documentación: Joaquin. G, Juan. Y. Informe: Benjamin. V	Avance en funcionamiento interno del robot
8	Interfaz, informe y presentación	Elaboración de informe de avance I y presentación		Avance en robot, informe y presentación

Ilustración 1 Programa de actividades I

## Actividades

Segunda tabla de actividades del proyecto.

Actividades durante el proyecto				
7	Programación de movimientos del robot	Prueba de movimientos del robot	Programador: Hernan. V, Bairon. N. Documentación: Joaquin. G, Juan. Y. Informe: Benjamin. V	Avance en funcionamiento interno del robot
8	Interfaz, informe y presentación	Elaboración de informe de avance I y presentación		Avance en robot, informe y presentación
9	Termino de informe y presentación de avance I	Termino de presentación e informe	Todo el grupo	Terminos de informe y presentación de avance I
10	Implementación de comunicación remota	Avance de robot e elaboración de informe final	Programador: Hernan. V, Bairon. N. Documentación: Joaquin. G, Juan. Y. Informe: Benjamin. V	Avance de robot e informe final
11	Diseño de arquitectura	Avance en arquitecturas del robot e informe final		
12	Pruebas e integración	Pruebas de funcionalidad	Todo el grupo	Pruebas del producto
13	Pruebas y elaboración de presentación final	Pruebas de robot y presentación final		Avance de presentación y robot
14	Termino de actividades del proyecto	Termino de presentación final, informe y pruebas		Termino del producto
15	Presentación del producto	Presentación de producto funcional		Presentación de producto
16	Exhibición de resultados del proyecto	Fin del proyecto		Exhibición de resultados

Ilustración 2 Programa de actividades II

### 3.2 Asignación de tiempo

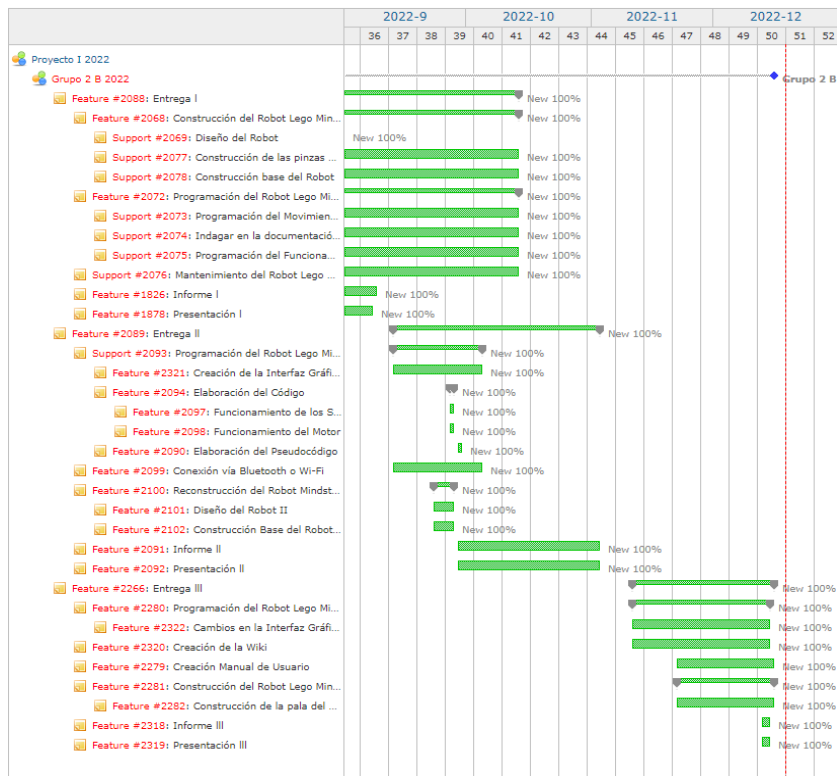


Ilustración 3 Asignación de tiempo

#### Entrega I:

- **Construcción del robot LEGO RoboCleaner EV3:** Procedimiento de armado del robot.
- **Programación del robot LEGO RoboCleaner EV3** Procedimiento de codificación del robot.
- **Mantenimiento del robot LEGO RoboCleaner EV3:** Pruebas de funcionamiento.

#### Entrega II:

- **Conexión vía Wi-Fi:** Pruebas de conexión de robot a equipo
- **Reconstrucción de robot LEGO RoboCleaner EV3:** Cambios en diseño de robot y nuevo funcionamiento
- **Programación de robot LEGO RoboCleaner EV3:** Cambios en elaboración de código fuente del robot.

#### Entrega III:

- **Termino de funcionalidades del robot:** Se termina todas las funcionalidades del robot.
- **Pruebas del código del robot:** Pruebas de que el código no contenga errores.
- **Cambios en interfaz:** Cambios y añadidos a la interfaz de usuario

### 3.3 Gestión de riesgos

Para determinar los accionares en caso de posibles riesgos se idea la siguiente tabla, la cual estará dividida en dichas categorías: 1. Catastrófico, 2. Crítico, 3. Marginal y 4. Despreciable cuya categoría está dividida según su grado de impacto y según su probabilidad de ocurrencia se determinará qué acción remedial corresponde.

Tratamiento de riesgos			
Riesgos	Probabilidad de ocurrencia	Nivel de impacto	Acción remedial
1. Problemas tecnologicos	70%	3	Investigación de personal en busca de solución
2. Dificultades de organización	60%	4	Reunión y cambio de roles del personal
3. Cambio de requisitos	60%	3	Revisión inmediata del proyecto e implementación de nuevos requisitos
4. Dificultades de tiempo del personal	50%	3	Reunión con fin de establecer cambio de actividades
5. Estimación del proyecto erronea	50%	2	Reformulación del proyecto
6. Problemas con entrega del proyecto	50%	4	Corrección y reunión del personal
7. Problemas de salud del personal	40%	3	Ajuste de roles del personal
8. Problemas de hardware o software	30%	2	Solicitud al departamento de equipos

*Ilustración 4 Gestión de riesgos*

## 4 Planificación de recursos

### 4.1 Recursos de hardware y software

#### Recursos de hardware y software empleados para elaboración del proyecto

Recursos de hardware y software		
Recurso	Utilidad	Manera de uso
1. Computadores	Trabajo en funcionalidades y entregables	Entregables y elaboración interna del robot
2. Kit LEGO Mindstorms	Armado del robot	
3. Tarjeta MicroSD	Almacenamiento de las implementaciones del robot	
4. Editor Visual Studio Code	Creación de código (funcionalidades)	
5. Lenguaje Python	Interprete de código	
6. Máquina Virtual LEGO	Interprete de funcionalidades del robot	
7. Plataforma Redmine UTA	Registro de actividades a lo largo del proyecto	Registro y Visualización de actividades
8. Gmail	Entrega de avances	Medios de comunicación
9. Discord	Medio de reunión del personal	
10. WhatsApp	Comunicación y organización del personal	
11. Telegram	Retroalimentar avances de personal a cliente	

Ilustración 5 Recursos de hardware y software

## 4.2 Estimación de costos

### Estimación de gastos en personal y recursos informáticos.

Ilustración 6 Costos de hardware y software

Hardware y Software				
Producto	Descripción	Cantidad	Estimado	
Kit Lego EV3 Education	KIT PIEZAS ROBOT LEGO	1	1.262.251	
Tarjeta MicroSD	TARJETA ALMACENAMIENTO	1	5.000	
Lenguaje Python	LENGUAJE DE PROGRAMACION	1	-	
Editor Visual Studio Code	EDITOR DE TEXTO	1	-	
Maquina Virtual LEGO EV3	SOFTWARE LEGO EV3	1	-	
Equipos (Notebook)	EQUIPOS DE TRABAJO	5	4.000.000	
TOTAL			5.267.251	

Ilustración 7 Costos de personal

Gestion de personal [estimado mensual]				
Función	Persona(s)	N* de Personas	Estimado	
Project Manager	Hernan.V	1	1.400.000	
Analista de programación	Bairon. N	1	1.000.000	
Administración	Benjamin. V	1		
Social Media Manager	Juan. Y, Joaquin G	2	470.000	
Estimado/mes			5.023.330	
TOTAL			20.093.320	

## 5 Análisis y diseño

### 5.1 Especificación de requerimiento

RoboCleaner es un robot que su función principalmente estará determinada en establecer una ruta y derribar a través de un lanzamiento un obstáculo. Para ello este funcionalmente será capaz de:

- **Uso de sensor ultrasonido:** el cual permitirá la correcta movilidad del motor, al este servir como un sensor de proximidad calculado la distancia entre el sonido producido y la superficie próxima.
- **Motor:** estará enlazado con la funcionalidad del sensor ultrasonido, y permite la dirección de su movilidad.
- **Sensor de color:** indica si este llega a un extremo de la ruta establecida,

### 5.2 Arquitectura del robot

#### 1 HARDWARE:

**Sensor de proximidad (Sensor Ultrasonido):** Dicha función estará determinada en los sensores del motor y ultrasonido, específicamente las bibliotecas EV3 UltrasonicSensor, el sensor ultrasónico es capaz de detectar ciertas frecuencias bajas del sonido y su velocidad sirviendo como sensor de proximidad del robot mejorando su movilidad.

**Sensor de color del robot:** Establece si el robot llega a un extremo a través de su recorrido, según el caso este será de un color u otro, el sensor que determinará esta funcionalidad estará delimitada por EV3 Color Sensor.

**Motor mediano:** El motor mediano o motor de rotación permite al robot girar dicha parte, esto se usa para la funcionalidad de la pala y hélice.

**Motor Largo:** Permite el movimiento del robot hacia cualquier destino.

#### 2 SOFTWARE:

**Máquina virtual LEGO EV3:** Software que permite ejecutar el código hecho en lenguaje Python al dispositivo LEGO EV3Brick.

**Biblioteca LEGO Mindstorms:** Biblioteca que faculta el importar funcionalidades para lo que requiere el robot.

## 5.3 Arquitectura del robot

### 3 COMUNICACIONES INVOLUCRADAS

Entre las conexiones realizadas, para conectar un equipo al robot es necesario primeramente que el equipo y el dispositivo LEGO EV3Brick compartan una misma conexión de red WI-FI, tras eso el equipo debe establecer al LEGO EV3Brick una dirección IP (ipv4) y un puerto del robot, tras ello habrá concluido la conexión del robot, en caso contrario indicara el error y el fallo de conexión del robot.



### 5.3 Interfaz y diseño

La siguiente imagen presenta la interfaz de usuario del robot, a través de cada punto señalado en la imagen se referenciara que funcionalidad cumple.

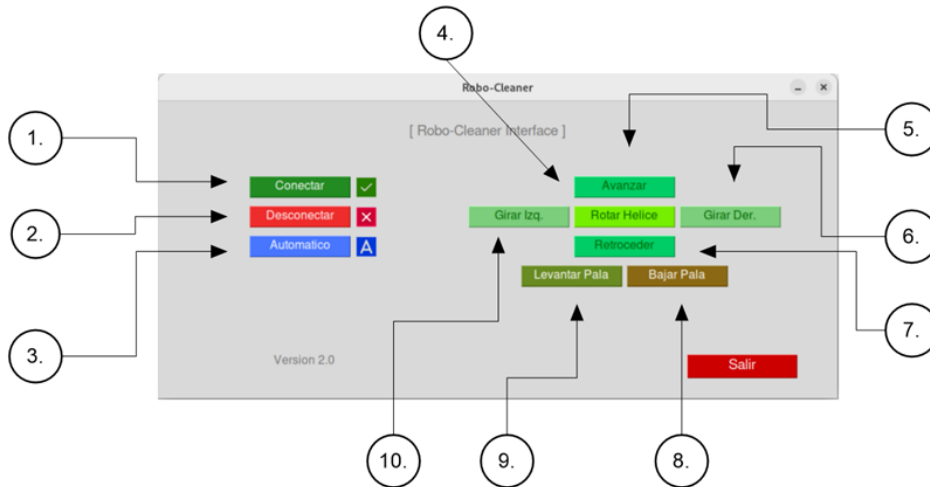


Ilustración 8 Interfaz de usuario RoboCleaner

#### FUNCIONALIDADES DE CONEXIÓN:

- 1. Conectar: Conectar dispositivo LEGO EV3 Brick a equipo a disponibilidad
- 2. Desconectar: Desconectar vinculo de equipo a dispositivo LEGO EV3 Brick

#### FUNCIONALIDADES DE MOVIMIENTO:

- 3. Automático: Movimiento delantero automatizado del Robot
- 4. Rotar hélice: Rota la hélice del Robot para derribar un objeto
- 5. Adelante: Movimiento delantero
- 6. Girar derecha: Movimiento hacia lado derecho del robot
- 7. Retroceder: Movimiento hacia atrás
- 8. Bajar Pala: Baja la pala en torno a 90 grados negativos
- 9. Levantar Pala: Sube la pala en torno de 90 grados
- 10. Girar izquierda: Movimiento hacia lado izquierdo del robot

## 6 Implementación

### 6.1 Fundamentos de proyectiles

#### DERRIBAMIENTO DE OBSTACULOS

El robot inicialmente para derribar un obstáculo este requiere de usar una determinada rotación y fuerza estimada, para determinar dicha fuerza es necesario obtener la masa del objeto, por lo tanto el fenómeno que se contempla para este procedimiento estará establecido en el fenómeno de movimiento circular uniforme y fuerza centrípeta que son los que permitirán dicha funcionalidad, aplicándose estos en sus respectivos ejes, adecuándose así a los conceptos de física e ingeniería establecidos para el funcionamiento del robot.

#### LIMPIEZA DE OBJETOS

Respecto a la funcionalidad de limpieza de objetos, para esta ser funcional en el motor ligado al motor de la pala del robot se aplica el fenómeno denominado torque o momento de una fuerza al momento de subir y bajar la pala, si el robot levante un objeto dicha fuerza también afectara al objeto en cuestión; para esta funcionalidad también influye la velocidad en la que se realiza el movimiento de la pala.

## 6.2 Descripción de programas

Respecto a la interfaz de usuario mostrada en la sección “Interfaz y diseño” [pág. 16], se describirá más detalladamente las funcionalidades de dichas opciones disponibles.

### FUNCIONALIDADES DE CONEXIÓN:

**[1] Conexión:** Para dar inicio con todas las funcionalidades del robot, primeramente se debe realizar la conexión del robot al equipo al cual se intenta conectar, para ello antes es necesario estar conectado a internet y a una misma red ambos dispositivos (Robot RoboCleaner y equipo). Luego directamente con las opciones que dispone la interfaz RoboCleaner realizar la conexión con el botón “Conectar”, ingresar la IP de red a la cual se conectara y finalmente la conexión habrá concluido, en caso contrario informara mediante la propia interfaz el error al intentar la conexión.

**[2] Desconexión:** Para realizar la desconexión del robot al equipo, únicamente es necesario interactuar con la opción “Desconectar” directamente desde la interfaz y tras un breve lapso de tiempo la desconexión del Lego EV3 Brick y el dispositivo se habrá realizado.

### 1 FUNCIONALIDADES DE MOVIMIENTO:

**[3] Automático:** La funcionalidad de “Movimiento automático”, tal como menciona, provoca que el robot se mueva automáticamente según la posición y dirección en la que esta, dicho movimiento solo se detendrá solo si el robot es capaz de encontrar un obstáculo y por consecuente lo derribara, en caso contrario solo continuara su movimiento.

**[4] Rotar hélice:** La siguiente funcionalidad es la encargada de derribar los obstáculos que estén frente al robot, esta acción al contrario de “Movimiento automático” es necesario ejecutarlo de manera manual según se vea conveniente y ejecutara dicha acción según el tiempo que el botón este accionado.

**[5] Avanzar:** Permite el movimiento hacia la dirección a la cual esta apuntado el robot, y para efectuarlo es necesario hacerlo manualmente desde la interfaz de usuario y mantendrá su movimiento si el botón sigue accionado.

**[6] Girar derecha:** Opción que provoca que el robot se incline y mueva a la derecha según la dirección que apunte el robot, su giro dependerá únicamente si el botón sigue apretado o no.

**[7] Retroceder:** Permite que el robot se mueva en dirección contraria a la que apunta.

## Descripción de programas

### 1.1 FUNCIONALIDADES DE MOVIMIENTO

**[8] Bajar pala:** El robot respecto al movimiento de su motor, baja la pala en torno a  $-90$  grados respecto a la posición de la pala y seguirá bajando si está presionado el botón.

**[9] Subir pala:** Al igual que su contraparte permite levantar la pala con la diferencia que lo hace a  $90$  grados.

**[10] Girar izquierda:** El robot se inclina y mueve hacia su lado izquierdo respecto hacia donde apunte este.

### 6.3 Diagramas

Los siguientes diagramas mostraran la serie de pasos por la cual el robot se comunica con el dispositivo o equipo al que realiza la conexión, tanto interno como externo.

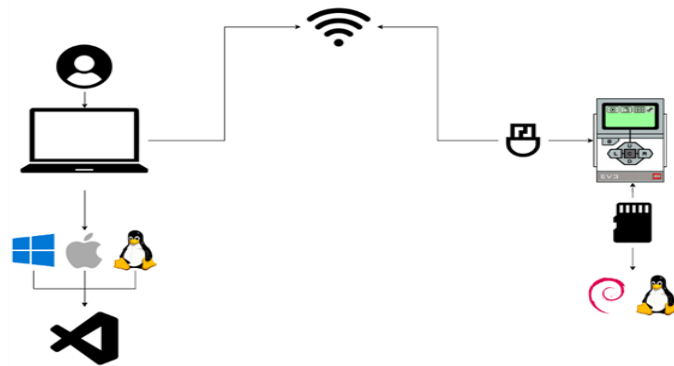


Ilustración 9 Diagrama de comunicación

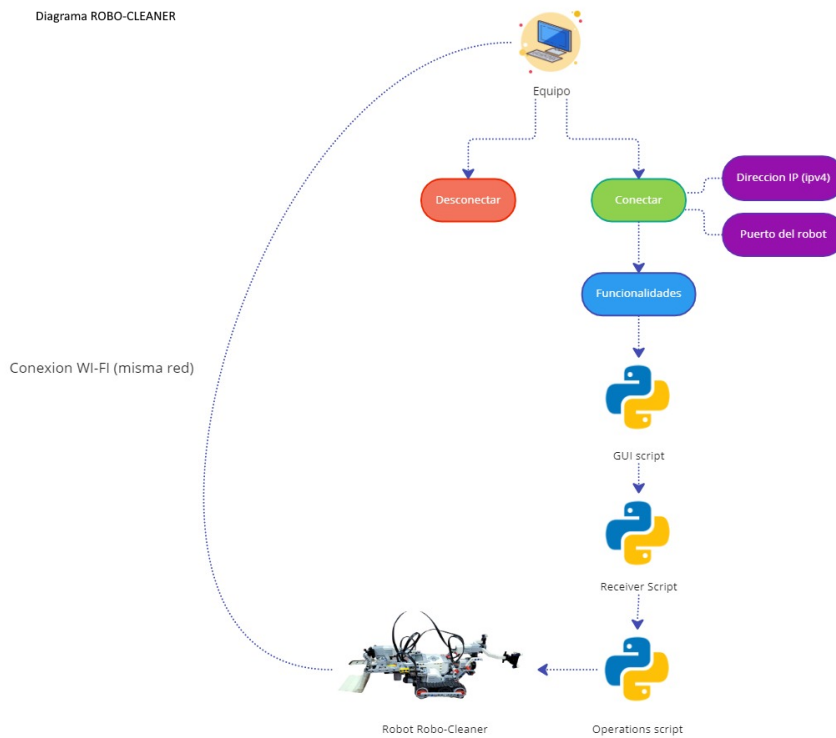


Ilustración 10 Diagrama de implementación

## 7 Pruebas

### 7.1 Descripción de pruebas realizadas

- **Pruebas de conexión Wi-Fi de robot:** Se corrige y comprueba si el robot (Lego EV3Brick) es capaz de conectarse a un determinado equipo.
- **Pruebas de conexión de motor:** Se realizan pruebas si el motor está conectado al dispositivo LEGO EV3BRICK de forma correcta.
- **Pruebas de sensores:** Se prueba que los sensores funcionen correctamente y que se adecuen a las funcionalidades para la implementación de la pala (código).
- **Pruebas de movilidad del robot:** Pruebas de las direcciones de movimiento del robot (código).
- **Pruebas de modo automático:** Pruebas de movimiento automático delantero del robot
- **Pruebas de la pala [funcionalidad]:** Se comprueba la efectividad del material de la pala.

### 7.2 Resultados de pruebas

- **Resultado P. conexión Wi-Fi de robot:** Se establece que la conexión entre el robot y el equipo a ocupar será únicamente por Wi-Fi.
- **Resultado conexión de motor:** Se logra la conexión de motor a robot
- **Resultado sensores:** Se implementa la nueva funcionalidad de la pala
- **Resultado movilidad del robot:** Se corrige partes deficientes del código y se logra que se ejecuten de manera correcta los movimientos del robot
- **Resultado modo automático:** Implementación del movimiento delantero automático del robot
- **Resultado P. pala:** Se cambia el material de la pala por uno mejor.

## 8 Resultados

### 8.1 Estado final del proyecto

El estado final del producto se encuentra en completa implementación de todas las funcionalidades propuestas del robot (derribar y limpiar objetos), cada funcionalidad con sus pruebas pertinentes lográndose cumplir el objetivo general destino para el proyecto.

### 8.2 Problemas encontrados y soluciones propuestas

Entre los problemas encontrados en la elaboración del proyecto.

- Problemas de tiempo del personal
- **Solución:** Distribución de tareas y relegación de actividades.
  
- Problemas de cambio de actividades de la Universidad
- **Solución:** Reunión fuera de la jornada.
  
- Problemas y errores en el levantamiento de la pala del robot (funcionalidad)
- **Solución:** Modificación de forma de la pala
  
- Problemas de falta de piezas del robot
- **Solución:** Solicitud de nuevas piezas al departamento.
  
- Problemas de conexión de motor
- **Solución:** Cambios en la implementación de código.
  
- Problemas de estabilidad rígida de motor
- **Solución:** Ajustes en la forma en que está construido el robot.
  
- Problema en la pala, no toca superficie
- **Solución:** Modificación de la forma de la pala

## 9 Conclusiones

### 9.1 Conclusiones generales

Tal como se había planteado los objetivos en inicios de la elaboración del proyecto, principalmente se contemplaba el idear una forma de emplear conceptos de la ingeniería y adecuarlos a la implementación del robot, actualmente este se encuentra terminado con todas las funcionalidades propuestas de parte de personal y requisitos del cliente, considerándose solo posibles añadidos del robot en un futuro, lográndose cumplir el objetivo establecido. Por otro lado a través del desarrollo del proyecto, este ha permitido al resto del personal desarrollar y comprender las fases e implementaciones de lo que requiere desarrollar un proyecto.

### 9.2 Trabajo futuro

Entre las posibles funcionalidades del robot que se podrían implementar en un futuro.

- Mejoras de funcionalidades, es decir añadir o cambiar las funcionalidades ya presentes para hacerlas más efectivas
- Permitir distintos tipos de conexión al robot, ya que la única conexión disponible de equipo a robot actualmente es únicamente por W-Fi, y con el cambio pertinente podría ser vía Wi-Fi y Bluetooth.



## 10 Referencias

- [1] LEGO, «LEGO MINDSTORMS EDUCATION,» LEGO, 2019. [En línea]. Available: <https://pybricks.com/ev3-micropython/>.
- [2] Indeed, «Salary Manager Chile,» 2022. [En línea]. Available: <https://cl.indeed.com/career/administración/salaries>.
- [3] Talent, «Salary Programmer Analyst Chile,» 2022. [En línea]. Available: <https://cl.talent.com/salary?job=analista+analista+programador>.
- [4] Talent, «Salary Project Manager Chile,» 2022. [En línea]. Available: <https://cl.talent.com/salary?job=project+manager>.
- [5] Indeed, «Salary Social Media Manager Chile,» 2022. [En línea]. Available: <https://cl.indeed.com/career/social-media-manager/salaries>.
- [6] L. MINDSTORMS, «Getting started with EV3 MicroPython,» 26 03 2019. [En línea]. Available: <https://le-www-live-s.legocdn.com/sc/media/files/ev3-micropython/ev3micropythonv100-71d3f28c59a1e766e92a59ff8500818e.pdf>.

## 11 Anexos

### Anexo [A]: HARDWARE

La siguiente imagen señala los componentes que usa el robot para su funcionalidad.

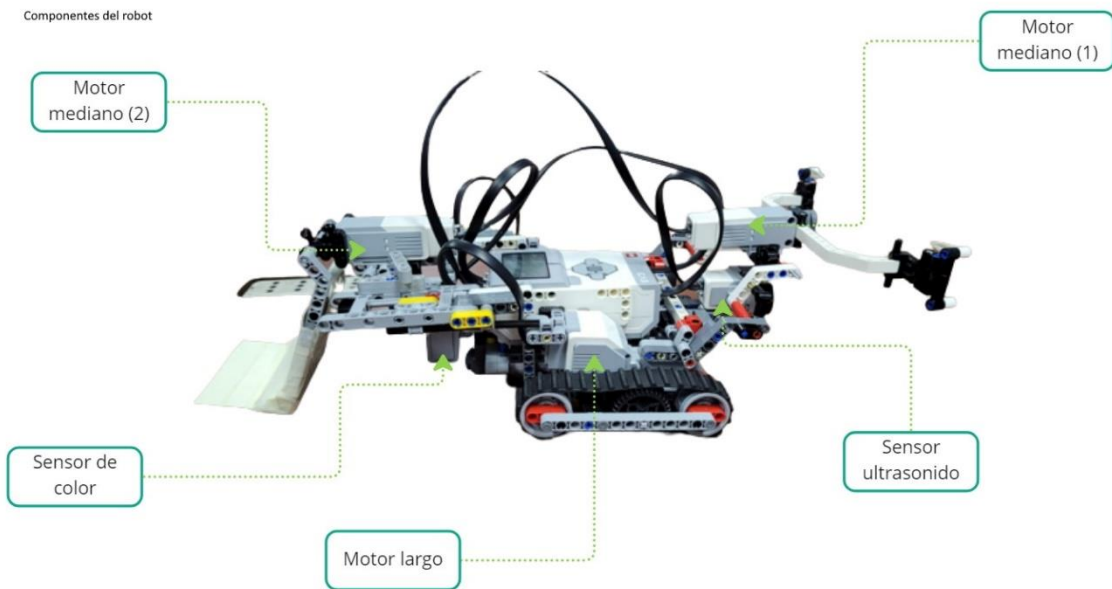


Ilustración 11 Componentes del robot

## Anexo [B]: SOFTWARE

El robot se comunica en base a 3 programas (scripts), "GUI", que contiene mayormente aspectos de implementación de la interfaz de usuario, luego "receiver", el cual recibe los comandos o acciones que el usuario realice y las envía a otro programa y finalmente "operations" que es quien recibe y ejecuta dichas acciones del robot.

### 1 INTERFAZ DE USUARIO [GUI]:

El programa GUI además de contener aspectos y opciones de la interfaz gráfica, esta también dispone de subrutinas que cumplen con las funciones del robot o están ligadas a otros scripts.

```

29 def conection(addr):
30     port = 12345
31     try:
32         client.connect((addr, port))
33         messagebox.showinfo("Mensaje Recibido",
34             "El host se conectó al robot: \nDirección IP: {0}\nPuerto: {1}".format(addr,port))
35     except socket.error:
36         messagebox.showwarning("Error de conexión",
37             "No se logró realizar la conexión. \nRevise la dirección IP e intentelo nuevamente. \nDirección IP: "
38             + addr)
39         get_IP()

```

Ilustración 12 conection

La función "conection" inicialmente recibe como parámetro una dirección IP (ipv4), para luego a través de un manejo de excepción se intenta conectar al socket (cliente) dado la dirección y un puerto, si la conexión al robot es exitosa entonces mostrara un mensaje al usuario de su estado, en caso contrario a través del manejo de excepción mostrara el error.

```

90 def disconnection():
91     try:
92         client.send(bytes([ord('q')]))
93         messagebox.showinfo("Información", "El robot se desconectó correctamente")
94     except BrokenPipeError:
95         messagebox.showwarning("Advertencia", "No hay un robot conectado a la máquina")

```

Ilustración 13 disconnection

La siguiente subrutina "disconnection" permite desconectar el robot del dispositivo al cual se haya realizado la conexión anteriormente, a través de un manejo de excepción se manda al cliente en formato Unicode representado en bytes la orden de realizar la desconexión, si el robot aún no ha sido conectado a ningún dispositivo, entonces se desplegara la excepción.

## Anexo [B]: SOFTWARE

### 1.1 INTERFAZ DE USUARIO [GUI]:

Entre las subrutinas que están ligadas a las funcionalidades principales del robot se encuentran.

```

72 def LeverUp():
73     try:
74         client.send(bytes([ord('y')]))
75     except BrokenPipeError:
76         messagebox.showerror("Error", "El robot no está conectado")
77
78 def LeverDown():
79     try:
80         client.send(bytes([ord('h')]))
81     except BrokenPipeError:
82         messagebox.showerror("Error", "El robot no está conectado")
83
84 def Automatic():
85     try:
86         client.send(bytes([ord('a')]))
87     except BrokenPipeError:
88         messagebox.showerror("Error", "El robot no está conectado")
89
90 def Halt(event):
91     try:
92         client.send(bytes([ord('w')]))
93     except BrokenPipeError:
94         messagebox.showerror("Error", "El robot no está conectado")

```

Ilustración 15 funciones gui

```

42 def Upward(event):
43     try:
44         client.send(bytes([ord('i')]))
45     except BrokenPipeError:
46         print()
47
48 def Backwards(event):
49     try:
50         client.send(bytes([ord('k')]))
51     except BrokenPipeError:
52         print()
53
54 def LeftTurn(event):
55     try:
56         client.send(bytes([ord('j')]))
57     except BrokenPipeError:
58         print()
59
60 def RightTurn(event):
61     try:
62         client.send(bytes([ord('l')]))
63     except BrokenPipeError:
64         print()
65
66 def Rotate(event):
67     try:
68         client.send(bytes([ord('p')]))
69     except BrokenPipeError:
70         print()
71

```

Ilustración 14 funciones gui

Las anteriores subrutinas realizan cada una de las funcionalidades del robot (automático, rotar hélice, subir pala, etc.), a través de bytes codificados en formato Unicode envía la información al script ligado el cual pertenece a “receiver” que será quien reciba dicha información, cada una de las subrutinas están enlazados con un manejo de excepción en caso que el usuario aun no haya realizado la conexión con el robot con anterioridad.

## Anexo [B]: SOFTWARE

### 2 Programa receiver [script]:

El script “receiver” es quien se encarga de recibir las órdenes de la interfaz GUI, tras recibirla codifica y lo transforma a formato Unicode (utf-8), tras eso, a través de condicionales compara el dato y según su tipo lo deriva al script “operations” y dado que aún está presente el ciclo while el robot aun estará a disponibilidad de realizar acciones; caso contrario si se realiza la desconexión del robot ya que inmediatamente se finalizara el ciclo while y se desconectara el dispositivo.

```

1  #!/usr/bin/env python3
2  import socket
3  from operations import *
4  conector = socket.socket()
5  puerto = 12345
6  conector.bind(('', puerto))
7  conector.listen(5)
8  conexion, addr = conector.accept()

```

Ilustración 18 receiver

```

9  while True:
10     dato_puro = conexion.recv(1)
11     key = dato_puro.decode("utf-8")
12     if (key == 'i'):
13         Upward()
14     if (key == 'k'):
15         Backwards()
16     if (key == 'j'):
17         LeftTurn()
18     if (key == 'l'):
19         RightTurn()
20     if (key == 'p'):
21         Rotate()

```

Ilustración 17 comp. Receiver I

```

22     if (key == 'y'):
23         LeverUp()
24     if (key == 'h'):
25         LeverDown()
26     if (key == 'a'):
27         Automatic()
28     if (key == 'q'):
29         break
30     if (key == 'w'):
31         Halt()

```

Ilustración 16 comp. Receiver II

### 3 Programa operations [script]:

El script “operations” ejecuta las acciones que son mandadas por el script “receiver”, para ello también en dicho script se importa los inputs del robot, sensores y motores a emplear, para el movimiento y movimiento automático del robot se considera la velocidad del motor, sensor de color y ultrasónico, mientras que para la subida y bajada de la pala del robot, se considera la velocidad del motor y el grado o ángulo.

```

1  #!/usr/bin/env python3
2  from ev3dev2.motor import SpeedPercent, MoveTank, LargeMotor,
3  MediumMotor, OUTPUT_A, OUTPUT_B, OUTPUT_C ,OUTPUT_D
4  from ev3dev2.sensor import INPUT_1, INPUT_4
5  from ev3dev2.sensor.lego import UltrasonicSensor, ColorSensor
6  tank = MoveTank(OUTPUT_A, OUTPUT_B)
7  mot_rot= MediumMotor(OUTPUT_D)
8  mot_lev = MediumMotor(OUTPUT_C)
9  prox = UltrasonicSensor(INPUT_1)
10 col = ColorSensor(INPUT_4)

```

Ilustración 22 imports operations

```

23 def Upward():
24     tank.on(left_speed= 90, right_speed= 90)
25 def Backwards():
26     tank.on(left_speed= -90, right_speed= -90)
27 def LeftTurn():
28     tank.on(left_speed= 90, right_speed= 0)
29 def RightTurn():
30     tank.on(left_speed= 0, right_speed= 90)
31 def Rotate():
32     mot_rot.on(speed = 75, brake=True)

```

Ilustración 20 funciones operations

```

11 def Automatic():
12     while (col.color != 6):
13         while (prox.distance_centimeters_continuous > 5):
14             tank.on(left_speed= 90, right_speed= 90)
15             if (col.color == 6):
16                 tank.off(brake=True)
17                 break
18         else:
19             tank.off(brake=True)
20             mot_rot.on_for_seconds(speed = 75, seconds = 1, brake=True)
21     tank.off(brake=True)

```

Ilustración 21 automatic

```

34 def LeverUp():
35     mot_lev.on_for_degrees(speed= 35, degrees= 90, brake= True)
36 def LeverDown():
37     mot_lev.on_for_degrees(speed= 35, degrees= -90, brake= True)
38 def Halt():
39     tank.off(brake=True)
40     mot_rot.off(brake=True)

```

Ilustración 19 funciones operations

## Anexo [C]: Comunicaciones

El robot inicialmente se conecta a través de una IP el cual hace de puente entre el dispositivo donde se ejecuta el código y el LEGO EV3Brick, además es necesario estar en una única red W-Fi ambos dispositivos, de esta manera es posible hacer uso de motores y sensores del robot. Mientras que para realizar la conexión del socket, el equipo (computador o laptop) tiene que conectarse al “socket server”, y el robot al “socket client”, dicha conexión no es necesaria realizarla mediante interfaz, ya que es necesaria para la funcionalidad del robot.