

Instituto Tecnológico de Costa Rica

Carrera de Ingeniería Mecatrónica



Rediseño del robot móvil Atta-Bot para actividades de educación en colegios públicos.

**Informe de Proyecto de Graduación para optar por el título de Ingeniero en
Mecatrónica con el grado académico de Licenciatura**

Leisel Carolina Villalobos Quesada

Cartago, junio 2024

Licencia creative commons



Esta obra está bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional.](#)

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, junio de 2024

Firma del autor

Leisel Carolina Villalobos Quesada

Céd: 117890779

INSTITUTO TECNOLÓGICO DE COSTA RICA
PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA
PROYECTO FINAL DE GRADUACIÓN

ACTA DE APROBACIÓN

El profesor asesor del presente trabajo final de graduación, indica que el documento presentado por el estudiante cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica del Instituto Tecnológico de Costa Rica para ser defendido ante el jurado evaluador, como requisito final para aprobar el curso Proyecto Final de Graduación y optar así por el título de Ingeniero(a) en Mecatrónica, con el grado académico de Licenciatura.

Estudiante: Leisel Carolina Villalobos Quesada

Proyecto: Rediseño del robot móvil Atta-Bot para actividades de educación en colegios públicos.

Ms. Marta Eugenia Vílchez Monge

Asesor

Cartago, junio 2024

INSTITUTO TECNOLÓGICO DE COSTA RICA

PROGRAMA DE LICENCIATURA EN INGENIERÍA MECATRÓNICA

PROYECTO FINAL DE GRADUACIÓN

ACTA DE APROBACIÓN

Proyecto final de graduación defendido ante el presente jurado evaluador como requisito para optar por el título de Ingeniero(a) en Mecatrónica con el grado académico de Licenciatura, según lo establecido por el programa de Licenciatura en Ingeniería Mecatrónica, del Instituto Tecnológico de Costa Rica.

Estudiante: Leisel Carolina Villalobos Quesada

Proyecto: Rediseño del robot móvil Atta-Bot para actividades de educación en colegios públicos.

Miembros del Jurado

Ing. Juan Luis Crespo Mariño

Jurado

Ing. Eduardo José Asenjo Castillo

Jurado

Los miembros de este jurado dan fe de que el presente proyecto final de graduación ha sido aprobado y cumple con las normas establecidas por el programa de Licenciatura en Ingeniería Mecatrónica.

Cartago, junio 2024

Resumen

Este proyecto busca tomar el diseño original de un robot llamado Atta-Bot creado dentro del proyecto PROE para la definición de rutas óptimas usando la robótica de enjambres y transformarlo en un robot educativo que pueda ser usado en las aulas de los colegios públicos de Costa Rica.

Para ello se realizará un diagnóstico del diseño actual para definir sus ventajas y desventajas. Después de esto se aplicará la metodología de Ulrich y Eppinger para determinar las necesidades y especificaciones del rediseño y, a partir de sus resultados, definir y seleccionar el concepto del diseño.

De este concepto se diseñará su arquitectura electrónica y su código principal. Por último, se construye un prototipo para la validación del diseño, con el cual se encontró que el robot cumple con los parámetros de precisión, autonomía, facilidad de carga y detección de obstáculos.

Palabras clave: Robótica, Robótica Educativa, Robótica de Enjambres.

Abstract

This project seeks to take the original design of a robot called Atta-Bot created within the PROE project for the definition of optimal routes using swarm robotics and transform it into an educational robot that can be used in the classrooms of public schools in Costa Rica. .

To do this, a diagnosis of the current design will be carried out to define its advantages and disadvantages. After this, the Ulrich and Eppinger methodology will be applied to determine the needs and specifications of the redesign and based on its results, define, and select the design concept.

From this concept its electronic architecture and main code will be designed. Finally, a prototype is built to validate the design, with which it was found that the robot meets the parameters of precision, autonomy, ease of loading and obstacle detection.

Keywords: Robotics, Educational Robotics, Swarm Robotics.

Dedicatoria

Para mi familia, que siempre ilumina mi camino y sostiene mi mano.

Agradecimiento

Mis agradecimientos más grandes a todos los miembros de mi familia que me han apoyado en cada paso de mi vida. A mi papá con su inteligencia y humor, a mi mamá con su carisma y bondad y a mi hermana con su espíritu guerrero. Ustedes son mis inspiración.

Tabla de contenidos

Lista de Figuras	iii
Lista de Tablas	vi
Lista de Abreviaciones.....	vii
1.0 Introducción	1
1.1 Antecedentes del proyecto.....	1
1.2 Descripción del problema a resolver.....	1
1.3 Síntesis del problema.....	2
1.4 Objetivos.....	2
1.5 Estructura del documento	3
2.0 Marco teórico	4
2.1 Robótica.....	4
2.1.1 Definición.....	4
2.1.2 Leyes de la Robótica	5
2.1.3 Tipos de Robots.....	6
2.1.4 Subsistemas de un Robot	10
2.2 Robótica Educativa	16
2.2.1 Definición.....	16
2.2.2 Importancia de la robótica educativa	17
2.3. Estado del arte	18
2.3.1 Robot Sphero Mini	19
2.3.2 Robot Zowi	19
2.3.3 Robot Mindstorms de Lego	20
2.3.4 Robot Bee-Bot	22
2.3.5 Robot Thymio	22
2.3.6 ELEGOO Smart Robot Car	23
2.3.7 Comparación de los robots educativos	23
2.3.8 Robótica educativa en Costa Rica	24
3.0 Metodología	27

3.1 Diagnóstico del Atta-Bot y definición del problema	27
3.2 Definición de las necesidades	36
3.3 Establecimiento de las especificaciones	43
3.4 Conceptualización del Proceso de Diseño.....	58
3.5 Proceso de Selección y Evaluación de Conceptos	87
4.0 Propuesta de Diseño	94
 4.1 Diseño Electrónico	94
 4.2 Diseño del código	106
5.0 Resultados y Análisis	112
 5.1 Pruebas de validación.....	112
 5.2 Análisis económico.....	121
6. 0 Conclusiones y Recomendaciones	124
 6.1 Conclusiones.....	124
 6.2 Recomendaciones.....	124
Referencias bibliográficas.....	125
Anexos.....	Error! Bookmark not defined.

Lista de Figuras

Figura 1. Comparación directa de un brazo robótico con la anatomía humana. [4].....	7
Figura 2. Robot AGV industrial. [1].....	8
Figura 3. Robot AMR de la Universidad Estatal de Ohio. [1].....	8
Figura 4. Robot PUMA con señalización del subsistema de movimiento. [1].....	12
Figura 5. Transmisión de banda síncrona para robot. [1]	13
Figura 6. Transmisión de cadena de rodillos para un robot. [1]	13
Figura 7. Tipos de engranajes de transmisión para robots. [1]	14
Figura 8. Mecanismo de transmisión de eslabones en un robot industrial. [1]	14
Figura 9. Esquema de control típico de un robot. [1]	16
Figura 10. Robot Sphero Mini. [12]	19
Figura 11. Robot Zowi. [14]	20
Figura 12. Robot humanoide construido con Lego Mindstorms. [15]	21
Figura 13. Robot móvil construido con Lego Mindstorms. [15].....	21
Figura 14. El robot Bee-Bot. [16].....	22
Figura 15. Robot Thymio. [17].....	23
Figura 16. Diagrama de alto nivel de los dispositivos electrónicos usados en el robot actual. [21]	31
Figura 17. Primer escenario de validación de los Atta-Bot. [21]	32
Figura 18. Segundo escenario de validación de los Atta-Bot. [21]	33
Figura 19. Matriz Necesidades-Métricas.....	45
Figura 20. Descomposición Funcional en la Etapa de Caja Negra.	59
Figura 21. Descomposición Funcional Final.....	59
Figura 22. Diagrama de la patente CN210189807. [29].....	64
Figura 23. Diagrama de la patente CN209986949. [30].....	64
Figura 24. Diagrama de la patente CN210361316. [31].....	65
Figura 25. Diagrama de la patente US20110129804. [32]	66
Figura 26. Árbol de clasificación de conceptos de la fuente de energía.	78
Figura 27. Árbol de clasificación de conceptos de los sensores de movimiento.....	79
Figura 28. Árbol de clasificación de conceptos de los protocolos de comunicación.	79
Figura 29. Árbol de clasificación de conceptos de los componentes educativos.	80

Figura 30. Árbol de clasificación de conceptos de placas de microcontroladores.....	80
Figura 31. Diagrama del primer candidato a la solución.....	81
Figura 32. Diagrama del segundo candidato a la solución.....	82
Figura 33. Diagrama del tercer candidato a la solución.	83
Figura 34. Diagrama del cuarto candidato a la solución.	84
Figura 35. Diagrama del quinto candidato a la solución.	85
Figura 36. Diagrama del sexto candidato a la solución.	86
Figura 37. Placa IdeaBoard elegida para el diseño. [33].....	95
Figura 38. Motores CD elegidos para el diseño. [33].....	95
Figura 39. Batería de litio de 2500mAh elegida para el diseño. [33].....	96
Figura 40. PowerBoost elegido para el diseño. [33]	97
Figura 41. Ball caster seleccionado para el diseño. [33]	97
Figura 42. Teclado de botones elegido para el diseño. [33]	98
Figura 43. Módulo de sensor infrarrojo seleccionado para el diseño. [33]	98
Figura 44. Módulo de sensor infrarrojo inferior seleccionado. [33]	99
Figura 45. Sensor de luz y color elegido para el diseño. [33]	100
Figura 46. Servomotor elegido para el diseño. [33]	100
Figura 47. Pantalla LCD a color elegida para el diseño. [33]	101
Figura 48. Interruptor elegido para el diseño. [33].....	101
Figura 49. Cables elegidos para el diseño. [33].....	102
Figura 50. Kit de construcción de jumpers seleccionado para el diseño. [33]	102
Figura 51. Cable Qwiic macho-macho elegido para el diseño. [33]	103
Figura 52. Cable Qwiic macho a 4 pines elegido para el diseño. [33]	103
Figura 53. Kit de tornillos, separadores y tuercas de nylon elegido para el diseño. [33]... <td>104</td>	104
Figura 54. Diagrama electrónico del Atta-Bot.....	105
Figura 55. Diagrama de la distribución física de los componentes del Atta-Bot.	105
Figura 56. Diagrama de estados del Atta-Bot.....	107
Figura 57. Imagen del proceso de validación de las distancias recorridas por el Atta-Bot.	113
Figura 58. Diagrama de Caja de Bigotes para el nivel de 100mm	114
Figura 59. Diagrama de Caja de Bigotes para el nivel de 200mm	114
Figura 60. Diagrama de Caja de Bigotes para el nivel de 300mm	114

Figura 61. Diagrama de Caja de Bigotes para el nivel de 900mm	115
Figura 62. Imagen del proceso de validación de los giros realizados por el Atta-Bot.	116
Figura 63. Diagrama de Caja de Bigotes para el nivel de 20°.....	117
Figura 64. Diagrama de Caja de Bigotes para el nivel de 45°.....	117
Figura 65. Diagrama de Caja de Bigotes para el nivel de 90°.....	117
Figura 66. Diagrama de Caja de Bigotes para el nivel de 120°.....	118
Figura 67. Diagrama de Caja de Bigotes para el nivel de 180°.....	118
Figura 68. Diagrama de Caja de Bigotes para el nivel de 240°.....	118

Lista de Tablas

Tabla 1. Comparación de los robots educativos.	24
Tabla 2. Características Generales de los Robots Atta-Bot. [21].....	28
Tabla 3. Componentes eléctricos del robot Atta-Bot actual.	29
Tabla 4. Jerarquización de necesidades primarias y secundarias.	39
Tabla 5. Jerarquización de las necesidades según su prioridad.	41
Tabla 6. Comparación de las Métricas de los Robots Educativos más Cercanos al Atta-Bot.	
.....	47
Tabla 7. Comparación de las Métricas de los Robots Educativos más Diferentes al Atta-Bot.	
.....	50
Tabla 8. Métricas del Atta-Bot Actual.	52
Tabla 9. Definición de Valores Meta y Valores Marginales para cada una de las métricas del Rediseño del Robot.....	55
Tabla 10. Resultados de la búsqueda externa desde el punto de vista de usuarios líderes..	61
Tabla 11. Listas de los componentes disponibles para el diseño del robot. [33].....	66
Tabla 12. Análisis de contradicciones, parámetros y principios de inventiva.	76
Tabla 13. Desarrollo de propuesta de inventiva y solución.	76
Tabla 14. Tabla de combinación para el primer candidato a la solución.	81
Tabla 15. Tabla de combinación para el segundo candidato a la solución.	82
Tabla 16. Tabla de combinación para el tercer candidato a la solución.	83
Tabla 17. Tabla de combinación para el cuarto candidato a la solución.	84
Tabla 18. Tabla de combinación para el quinto candidato a la solución.	85
Tabla 19. Tabla de combinación para el sexto candidato a la solución.	86
Tabla 20. Etapa de filtrado de la selección de conceptos.	88
Tabla 21. Etapa de evaluación de la selección de conceptos.....	90
Tabla 22. Resultados de la validación de las distancias recorridas por el Atta-Bot.	113
Tabla 23. Resultados de la validación de los giros realizados por el Atta-Bot.....	116
Tabla 24. Resultados de la validación de la detención de obstáculos del Atta-Bot.....	119
Tabla 25. Resultados de la validación de la facilidad de carga del Atta-Bot.....	120
Tabla 26. Análisis económico del proyecto.	121

Lista de Abreviaciones

ADC = Analogic Digital Convertor

AGV = Vehículos Guiados Automáticamente

AMR = Robots móviles Autónomos.

BOM = Bill of Materials

CPU = Central Processing Unit

12C = Inter-Integrated Circuit

IMU = Unidad de Medida Inercial

ISO = Internaciona Organization for Standardization

LCD = Liquid Crystal Display

LED = Light Emitting Diode

RGB = Red Green Blue

RIA = Instituto de Robótica de América

PID = Proporcional Integral Derivativo

PROE = Planificación de Rutas Óptimas por medio de Enjambres

PUMA = Programmable Universal Machine for Assembly

STEM = Science Technology Engineering Math

TEC = Instituto Tecnológico de Costa Rica

TRIZ = Teoría de Resolución de Problemas de Innovación

USB = Universal Serial Bus

USB-C = Universal Serial Bus de tipo C

USB-mini = Universal Serial Bus de tipo mini

Capítulo 1.0 Introducción

Durante este capítulo se busca introducir al lector al problema que el proyecto pretende resolver y la forma en la que se estructura el diseño de su solución. En primer lugar, se citan los antecedentes del proyecto para entenderlo mejor. Después, se describe a fondo el problema que el proyecto pretende resolver, así como una síntesis de éste con el objetivo de aclarar la razón detrás del rediseño. Se presentan también los objetivos, tanto el general como los específicos, ya que estos definen el alcance del proyecto y, por último, se define la estructura que tomará el documento para su desarrollo.

1.1 Antecedentes del proyecto

El proyecto PROE es un proyecto de investigación del Instituto Tecnológico de Costa Rica en el que colaboran la Escuela de Matemática y el Área Académica Ingeniería Mecatrónica, iniciado en 2019 con el objetivo de diseñar un sistema multirobot físico para la exploración y digitalización de escenarios y determinación de rutas óptimas.

Dentro de este proyecto se diseñó un robot llamado Atta-Bot, un enjambre está compuesto de múltiples de estos robots. En la actualidad, con base en el proyecto PROE, se desarrolló un nuevo proyecto llamado Atta-Bot STEM: Impulsando la educación pública mediante una plataforma robótica desarrollada por el TEC.

La implementación de la robótica educativa en las instituciones públicas es importante porque este tipo de programas permiten que los estudiantes de todas edades se familiaricen con los principios de robótica y programación. Además, las actividades que se llevan a cabo con robots educativos también desarrollan habilidades cognitivas, el pensamiento lógico matemático e implementan nuevas dinámicas de aprendizaje en las aulas. Esta es la razón por la que se busca tomar el simple y barato diseño del Atta-Bot y transformarlo en un robot educativo accesible al público costarricense.

1.2 Descripción del problema a resolver

El robot de enjambre desarrollado en el proyecto PROE, llamado Atta-Bot, fue diseñado para funcionar dentro de los parámetros de robótica de enjambres. Por esta razón, el Atta-Bot actual posee una combinación de componentes que desarrollan tareas ligadas a este tipo de robótica. Por ende, al introducir al robot a un ambiente educativo, se debe crear un nuevo

diseño que implemente el bajo costo, la precisión y ensamblaje local del robot actual con una nueva combinación de sensores, actuadores y controladores orientados específicamente a cumplir funciones de robótica educativa con software y hardware abierto y público. También se busca implementar tareas de construcción del robot en las aulas.

Además, se espera que este nuevo diseño use como base el código creado para la lógica de enjambres, incluyendo nuevas funciones educativas, así como que el robot utilice un protocolo de comunicación que lo conecte a la aplicación diseñada para su programación de forma inalámbrica.

1.3 Síntesis del problema

Los Atta-Bots actuales fueron creados para la robótica de enjambre, lo que significa que sus funciones principales pueden ser usadas también en un robot educativo si se implementan cambios enfocados en la simplificación de la motora y en la expansión de los actuadores y sensores del robot. Por esta razón es necesario su rediseño para crear un robot educativo de bajo costo y de software y hardware abierto y producido localmente.

1.4 Objetivos

Objetivo general:

Rediseñar el robot llamado “Atta-Bot” dentro del proyecto de investigación PROE para que su resultado final corresponda a un robot educativo que se utilice en aulas de colegios.

Objetivos específicos:

1. Diagnosticar la condición actual del robot Atta-Bot con respecto a los parámetros buscados en un robot educativo.
2. Diseñar la arquitectura del nuevo Atta-Bot con un enfoque en la robótica educativa.
3. Diseñar el código que utiliza el controlador del robot para cumplir con las instrucciones programadas, crear una conexión con la aplicación, controlar la locomoción y utilizar los actuadores y sensores del robot.
4. Validar el rediseño del robot, el código interno y el protocolo de comunicación.

1.5 Estructura del documento

El proyecto fue desarrollado siguiendo la metodología Ulrich y Eppinger que fue documentada y ejemplificada en su libro “Diseño y desarrollo de productos”. Según esta metodología, se tiene una serie de etapas que se siguen para el desarrollo del producto final.

En primer lugar, se desarrolla un marco teórico que busca aclarar los conceptos más importantes que forman la base del proyecto. Después de esto se debe de realizar un diagnóstico del Atta-Bot actual. Este diagnóstico estará enfocado en determinar que partes del robot deben ser reemplazadas para crear una versión de éste que sea funcional para la robótica educativa. Además, debido a que el robot fue diseñado y construido algunos años atrás, se identificarán cuáles de sus partes tienen tecnología desactualizada o han incrementado su precio por un aumento en su demanda.

A continuación, se realiza una entrevista inicial con dos de los investigadores del proyecto PROE, uno por cada área de investigación, para determinar una serie de necesidades que definen el enfoque de la solución. Además, en esta etapa se busca eliminar cualquier tipo de sesgos para un mejor resultado final. Eventualmente se realiza una clasificación de las necesidades según su prioridad.

Una vez que se tienen las necesidades finales y su orden de prioridad, se definen valores que deben de ser alcanzados según cada necesidad; estos valores corresponden a las especificaciones del proyecto y son definidos mediante información externa e interna y lo que se busca según el criterio de los investigadores del proyecto.

Con esta información se generan una serie de conceptos que combinan diferentes soluciones para diferentes subproblemas. Después, se realizan diferentes procesos para definir cuál de los conceptos es el mejor, para lo que se tomará en cuenta el precio de la solución, la facilidad de la compra de componentes, la facilidad de control del robot, la importancia de los componentes en robótica educativa y la compatibilidad de la solución de los subproblemas entre sí. El concepto considerado ganador será utilizado para crear un prototipo.

Por último, el prototipo se utilizará para validar la funcionalidad de los robots dentro de las especificaciones definidas inicialmente. Especialmente se busca validar la precisión de sus movimientos y giros, su tiempo de autonomía y su capacidad de detección de obstáculos.

Capítulo 2.0 Marco teórico

Este capítulo busca explicar a fondo todos los conceptos teóricos necesarios para el entendimiento del proyecto. Se presenta información especialmente sobre la robótica en todas sus facetas relevantes con respecto al proyecto, como la robótica de enjambres y la robótica educativa. Además, existe una sección del estado del arte que presenta un resumen del diseño de algunos de los robots educativos actuales para un mejor entendimiento del punto de inicio del proyecto.

2.1 Robótica

En esta sección se busca establecer los conceptos teóricos ligados a la robótica, los cuales son necesarios para la comprensión del proyecto desarrollado. Estos comprenden su definición, tipos de robots y otros temas de interés.

2.1.1 Definición

En primer lugar, para establecer una definición de robótica, se debe conocer qué es un robot. Definir un robot es una tarea complicada, debido a que esta definición cambia en orden de la época, el enfoque y el ambiente en que se desarrolle. Un robot, según el Instituto de Robótica de América (RIA), puede ser definido como “una máquina programable, de uso práctico, que se puede programar para llevar a cabo una serie de tareas automáticamente, con o sin supervisión humana”. [1]

Por otro lado, el robot también ha sido definido por la Organización Internacional para la Estandarización (ISO), como “un manipulador multifuncional reprogramable, capaz de mover materiales, piezas, herramientas o dispositivos especiales, a través de movimientos variables programados, para el desempeño de tareas diversas.” [1]

Como se puede observar, ambas definiciones son muy similares, y para este caso en concreto se define al robot como una máquina que puede cumplir una serie de instrucciones de diferentes áreas de forma automatizada, por medio de la reprogramación de su software. Este corresponde a la definición ideal en el ambiente del proyecto.

Una vez que se tiene la definición de robot, se define que es la robótica. La definición más reconocida de la robótica según Legendre (1988) es “conjunto de métodos y medios derivados de la informática cuyo objeto de estudio concierne la concepción, la programación

y la puesta en práctica de mecanismos automáticos que pueden sustituir al ser humano para efectuar operaciones reguladoras de orden intelectual, motor y sensorial.”. [2]

Sin embargo, la definición de robótica ha evolucionado por lo que, actualmente y en general, se considera que es una ciencia que agrupa múltiples disciplinas o ramas de la tecnología, con la finalidad de crear robots. Esta es la definición más relevante para esta investigación.

2.1.2 Leyes de la Robótica

El término robot surge de la ciencia ficción, pero estos escritores también fueron los responsables de establecer la guía moral del desarrollo de la robótica en el futuro. Entre estos trabajos se destaca el desarrollo de las llamadas leyes de la robótica propuestas por Isaac Asimov en la revista Galaxy Science Fiction (1945), que consisten en reglas básicas que debe de seguir un robot para ser considerado seguro para humanos.

La primera ley estipula que un robot no puede dañar o perjudicar a un ser humano, esto incluye que no lo puede dañar con su inacción. Lo que tiene como significado que un robot no puede ser programado para dañar a seres humanos y además debe de ser capaz de actuar para evitar daños a los seres humanos. La segunda ley indica que un robot tiene que obedecer las órdenes recibidas, y siempre que estas órdenes no entren en conflicto con la primera ley. Es decir, toda tarea que sea ordenada al robot por un ser humano debe ser ejecutada, siempre y cuando esta no dañe a seres humanos. Por último, para la tercera ley se tiene que un robot debe de proteger su propia existencia, siempre y cuando esto no interceda con las leyes anteriores. De esta regla se analiza que el robot debe de cuidar y respaldar por su propia existencia mientras que esta acción no dañe a seres humanos y no le impida llevar a cabo el cumplimiento de las instrucciones recibidas. [3]

Como se puede observar, todas estas leyes dependen del diseño y construcción del hardware y software del robot. Por lo que se dice que estas se utilizan como una importante guía a la hora de diseñar un robot. Su seguimiento permite la creación de un robot que será útil para el ser humano, sin crear ningún tipo de peligro o daño. Además, al autor se le ha reconocido su contribución a la creación del término robótica y con la divulgación y difusión de la robótica, por lo que sus aportes son considerados relevantes a la hora de la creación de parámetros en este ámbito.

Más tarde se introdujo una cuarta ley, Fuller (1999) estableció que un robot puede tomar el trabajo o la labor que realiza un ser humano, pero no debe dejar a esa persona sin empleo. Esta ley busca establecer una guía moral acerca de la utilización de robots a las zonas industriales u otros ámbitos. De esta forma, los robots pueden ser usados para tareas difíciles y peligrosas o para la mejora de los procesos de manufactura, sin desemplear a los humanos. Si no, se busca que el empleo de los seres humanos evolucione a otras áreas que no pueden ser controladas por los robots ya que dependen de la naturaleza humana para su desarrollo.

[1]

Es importante reconocer que, a pesar de existir un método para apegarse a estas leyes, no hay un modo automático para su implementación. Existen múltiples robots que son diseñados en completa contradicción a las leyes mencionadas, como los robots militares. Por esto es por lo que estas reglas están creadas para ser seguidas por los diseñadores y los robots no son capaces de aplicarlas por sí solos.

2.1.3 Tipos de Robots

Debido a que el término robot abarca una gran cantidad de máquinas en una variedad de ámbitos, se han creado diferencias en los tipos de robots para los cuales se usan sus características básicas para su clasificación. Durante esta sección se exploran algunos de los tipos de robots relevantes para el proyecto.

2.1.3.1 Tipos de robots según movilidad

Uno de los tipos de clasificación de robots se basa en su movilidad. Esto quiere decir que se basa en su capacidad y desempeño de movimiento y la toma de decisiones relacionada al movimiento. En esta clasificación se tienen robots articulados o brazos robóticos, vehículos de guiado automático, robots móviles autónomos y robots humanoides.

Los robots articulados, más comúnmente llamados brazos robóticos, corresponden a los típicos robots industriales. Estos, en el área de movilidad, se caracterizan por tener una capacidad muy reducida, por lo que son principalmente usados para mover productos, empaquetar o manipular herramientas desde un punto fijo. Este tipo de robots son referidos como la base de la robótica y el origen de su estructura y funciones está basado en las semejanzas a un brazo humano. Por esta razón los brazos robóticos poseen una articulación

de hombro que corresponde a la articulación que sostiene la mayor cantidad de peso; una articulación de codo que busca aportar extensión y retracción para el posicionamiento de la muñeca y una articulación de muñeca que cumple la función de posicionar de forma precisa el efector final de un brazo robótico [4]. La comparación directa de la anatomía humana con la del brazo robótico se encuentra en la Figura 1.

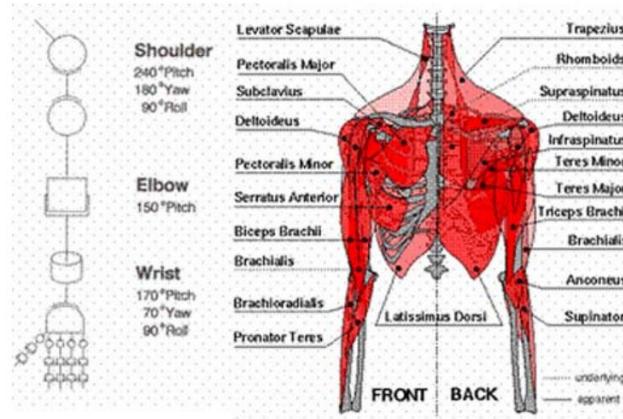


Figura 1. Comparación directa de un brazo robótico con la anatomía humana. [4]

Los vehículos guiados automáticamente, los cuales son referidos en la industria por sus siglas en inglés AGV, se pueden describir como sistemas robóticos móviles, cuyo principal uso corresponde al manejo de materiales en fábricas. En la mayoría de los casos los robots AGV están configurados para seguir una trayectoria señalada en el piso, de aquí viene su nombre, ya que son guiados automáticamente por la trayectoria marcada. En este tipo de robots se expande la capacidad de movilidad de los robots articulados y por esto sus aplicaciones cambian ampliamente [1]. En la Figura 2 se encuentra un ejemplo de un robot AGV.



Figura 2. Robot AGV industrial. [1]

Los robots móviles autónomos, también conocidos como AMR por sus siglas en inglés, son capaces de moverse y tomar decisiones por si solos. Esto significa que estos robots son utilizados para aplicaciones en los que se busca que puedan navegar en un ambiente impredecible. Los AMR tienen un nivel de movilidad libre ya que pueden crear un modelo del ambiente y ubicarse en éste, mediante la medida de parámetros creados por el ambiente, gracias a los sensores incorporados en su diseño [5]. En la Figura 3 se muestra un ejemplo de un AMR.

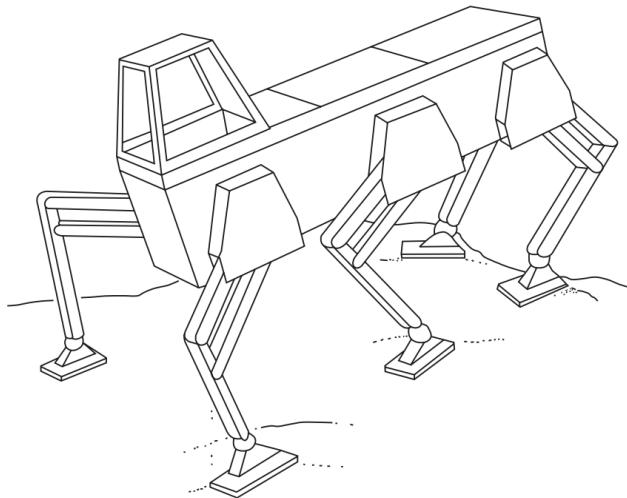


Figura 3. Robot AMR de la Universidad Estatal de Ohio. [1]

Los robots humanoides en particular corresponde a un tipo de AMR con forma humana y con funciones que se asemejan a las realizadas por los seres humanos. También son llamados

robots caminantes debido a que su forma de desplazarse se basa en la forma de caminar de los humanos. [1]

2.1.3.2 Tipos de robots por función o sector

Este tipo de clasificación ha tomado cada vez más relevancia debido a que los robots han sido implementados en un creciente número de ámbitos de la vida humana. En concreto se describirán los robots industriales, domésticos, educativos, militares y médicos.

La definición concreta de los robots industriales ha sido un punto de discusión, debido a las diferencias entre el mercado japonés, estadounidense y europeo. En primer lugar, el mercado japonés reconoce como un robot industrial a cualquier máquina con articulaciones que esté creada para manipulación, por otro lado, el mercado occidental prefiere extender la definición a algo más complejo, incluyendo la incorporación de un sistema de control [3]. Pero en general, según su función este tipo de robots se utilizan en la industria para manipular diferentes objetos en los procesos de producción. Son los robots con más historia de desarrollo y los más documentados en la robótica.

Los robots domésticos, como indica su nombre, son diseñados para realizar diferentes tareas dentro del hogar. También pueden ser definidos como robots de servicio y su propósito es aliviar la carga de trabajo dentro de las casas. Un robot doméstico debe de realizar las tareas de las casas con total autonomía, buscando eliminar al máximo la intervención humana. Usualmente los seres humanos solo deberán realizar una programación inicial para que el robot conozca las preferencias del usuario y el hogar en el que ubica. [6]

Los robots militares son aquellos encargados de actuar como un apoyo en ciertas operaciones militares de los ejércitos. Entre las actividades que realizan estos robots se encuentra el transporte material y la ayuda de detección de explosivos, entre otros. Como se puede observar, las funciones que son llevadas a cabo por estos robots son actividades peligrosas para el ser humano, por esta razón se busca incorporar la tecnología robótica para reducir el riesgo de la pérdida humana.

Los robots médicos consisten en el resultado de la aplicación de la técnica robótica en los procedimientos médicos, con el fin de generar nuevas oportunidades para la ingeniería sanitaria y biomédica. Por ende, estos robots se incorporan en el desarrollo de procedimientos

médicos, después de pasar por pruebas que los validen como herramientas seguras. Entre las tecnologías incorporadas en esta área se encuentra la robótica quirúrgica y la robótica para rehabilitación y asistencia. [7]

Los robots educativos son máquinas desarrolladas a partir del concepto de robótica pedagógica. Estos robots permiten que los estudiantes inicien desde muy jóvenes en el estudio de las ciencias y la tecnología y su robótica está basada en teoremas de epistemología y psicología. [2]

2.1.3.3 Robots de enjambre

Los robots de enjambre pueden ser descritos como aquellos robots diseñados con el propósito de formar un equipo de colaboración conocido como un enjambre. Los enjambres de robots sirven como una aproximación para la colaboración de sistemas formados por robots que actúan como agentes autónomos y que interactúan con ellos mismos y con su entorno. Los enjambres de este tipo se caracterizan por tener muchos robots, pero el comportamiento de cada individuo es simple. [8]

El objetivo principal de los enjambres de robots consiste en determinar, a partir de sus conductas y sus interacciones con otros agentes y con su entorno, patrones de comportamiento colectivo predeterminados de forma distribuida y autoorganizada. Los procesos autoorganizados ocurren comúnmente en la naturaleza, como en la distribución de los tallos de las plantas, las colonias de abejas o los hormigueros. Estos procesos toman un problema complejo y lo resuelven de forma simple mediante los patrones de colaboración de múltiples individuos. [8]

Al desarrollar un enjambre de robots comúnmente se busca que sean capaces de realizar una tarea compleja en un menor tiempo y de forma más sencilla ya que trabajan en conjunto. Por esta razón los robots pueden ser más simples porque su fortaleza recae en la colaboración programada en su software. [8]

2.1.4 Subsistemas de un Robot

En general, un sistema robótico puede ser dividido en tres subsistemas generales y, dependiendo de la complejidad de un robot, estos subsistemas después pueden ser divididos

también. Los principales subsistemas corresponden al subsistema de movimiento, el subsistema de reconocimiento y el subsistema de control.

En términos generales, el subsistema de movimiento corresponde a la estructura física del sistema robótico que realiza un movimiento deseado. Dependiendo del tipo de robot, éste puede ser un movimiento similar al que realiza un brazo humano o puede ser el desplazamiento del robot; algunos robots pueden realizar ambos movimientos.

Por otro lado, el subsistema de reconocimiento es descrito como aquel encargado de recolectar información por medio de sensores para comprender las condiciones del propio robot, los objetos que serán manipulados y el ambiente de trabajo.

Por último, el subsistema de control cumple la función de regular el movimiento que realiza el robot con el fin de completar una tarea. Este se apoya de la información obtenida por el subsistema de reconocimiento. [1]

Es importante destacar que, para un funcionamiento correcto de un robot, todos los subsistemas deben de funcionar al unísono. Todos los subsistemas dependen el uno del otro para completar la tarea del robot de la mejor forma posible. Esto es uno de los factores fundamentales en el proceso del diseño de un robot.

2.1.4.1 Subsistema de movimiento

Dentro de este subsistema se pueden encontrar algunos elementos de forma consistente. Estos corresponden a los manipuladores, los efectores finales, los actuadores y la transmisión. El trabajo de estos elementos en conjunto permite el movimiento de un robot.

Un manipulador consiste en la parte o estructura física que se está moviendo e incluye a los llamados eslabones o cuerpos y a las articulaciones o pares cinemáticos. Estas partes están normalmente conectadas en serie. Los eslabones son estructuras fijas que crean la estructura del robot, por otro lado, las articulaciones son las partes que permiten el movimiento el cual puede ser rotatorio o de traslación [1]. Estas partes se pueden reconocer en la Figura 4, la cual ejemplifica un robot PUMA.

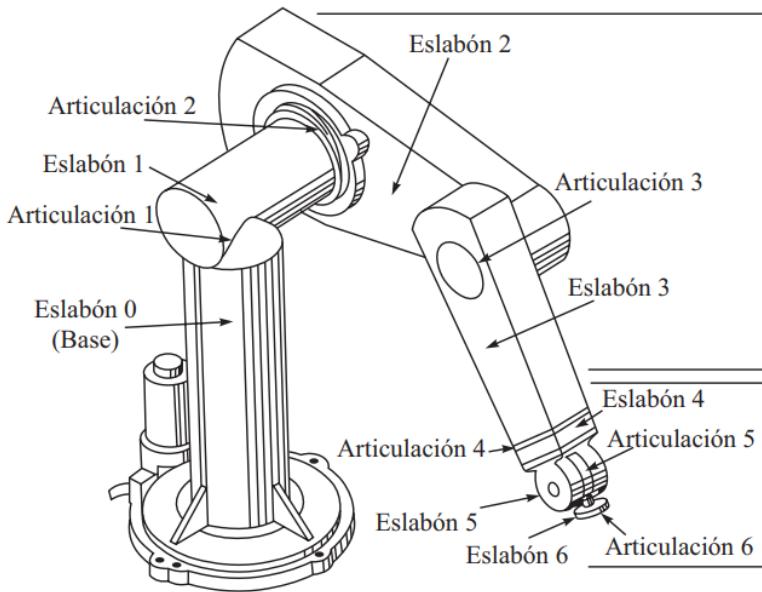


Figura 4. Robot PUMA con señalización del subsistema de movimiento. [1]

El efecto final corresponde a la parte del robot que se encuentra instalada al final del manipulador. Esta parte puede llevar a cabo una serie de tareas, como sostener un objeto, moverlo, soldar, quemar piezas, enfriar piezas, entre otros. Como su función varía, también varía la forma en la que están construidos, por lo que existen actuadores finales mecánicos, eléctricos, neumáticos e hidráulicos. Además, también varía su complejidad y la cantidad de acciones que pueden cumplir. [1]

Los actuadores del subsistema de movimiento son aquellas partes del robot que proporcionan el movimiento tanto para el manipulador como para los efectores finales. Pueden ser calificados como eléctricos, hidráulicos o neumáticos dependiendo de su forma de funcionamiento. La selección del tipo de actuador que será usado está ligada a la acción que debe cumplir el robot, el presupuesto y la naturaleza del ambiente en el que éste será usado. [1]

Por último, la transmisión se define como los elementos encargados de transmitir el movimiento proporcionado por motores y actuadores a los eslabones del manipulador. Existen diferentes tipos de transmisores, su escogencia se ve ligada al tipo de actuador, el tipo de robot y la acción que será realizada por él. La transmisión por banda y cadena se usa especialmente como banda síncrona, como se muestra en la Figura 5. Debido a que dependen

de la tensión de la banda para crear un agarre a través de la polea, su vida útil es breve. Este mismo tipo de transmisión también se puede utilizar por medio de cadenas, como se muestra en Figura 6. Estas son más económicas y poseen una mayor capacidad de carga y una mayor vida útil, aunque en general es menor a la vida útil de los engranajes. [1]

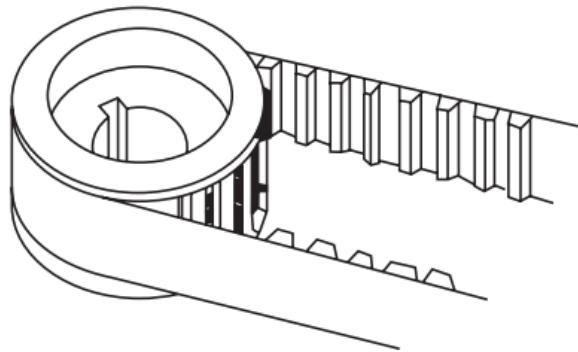


Figura 5. Transmisión de banda síncrona para robot. [1]

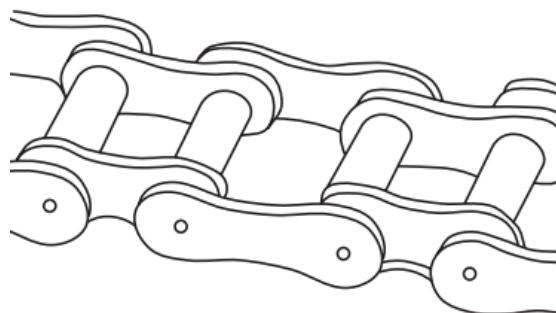


Figura 6. Transmisión de cadena de rodillos para un robot. [1]

El otro tipo de transmisión ampliamente usada corresponde a los engranajes. Existen muchos tipos de engranajes como se muestra en la Figura 7. Esta lista de engranajes posee los tipos más duraderos y confiables. En la fase de diseño, se debe de tomar en cuenta el juego de dientes cuidadosamente para que la transmisión sea optima tomando en cuenta la velocidad, los eslabones y las articulaciones. [1]

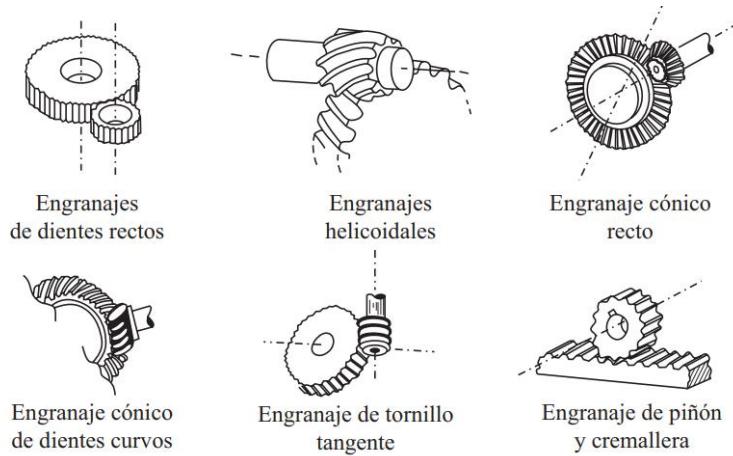


Figura 7. Tipos de engranajes de transmisión para robots. [1]

El último tipo de transmisión corresponde a los mecanismos de eslabones. Este método nace de la necesidad reducir el peso de los elementos mencionados anteriormente. Este tipo de mecanismos comprometen la alta flexibilidad de los engranes, las cadenas y la banda síncrona por los que son usados para una menor cantidad de robots [1]. Un ejemplo se encuentra en la Figura 8.

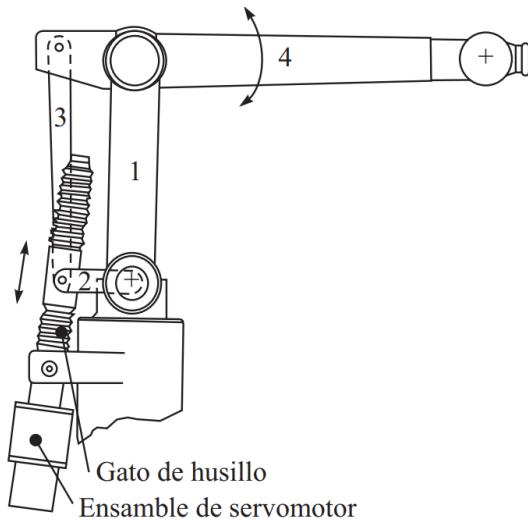


Figura 8. Mecanismo de transmisión de eslabones en un robot industrial. [1]

2.1.4.2 Subsistema de reconocimiento

El subsistema de reconocimiento está basado en torno a los sensores, estos pueden ser descritos como los elementos de un robot que le permiten tener los sentidos de la vista, tacto

u olor. La implementación de los sensores en un robot es la clave para que éste sea una máquina inteligente, ya que, con este elemento, pueden detectar señales análogas. Al implementar un convertidor analógico digital estas señales pueden ser procesadas por el robot y así ser usadas en su control y su acción. Los principales elementos de este sistema corresponden a los sensores y el convertidor analógico digital. [1]

Los sensores son, en su mayoría, esencialmente transductores que convierten la forma de una señal en otra. En la mayor parte de los casos los sensores toman una señal del ambiente y la transforman en una señal de electrónica analógica [1]. Existen sensores en múltiples áreas y para múltiples tareas y, en el área de la robótica es importante reconocer que los sensores son los elementos que permiten que el robot interactúe con su ambiente y tome decisiones por medio de la información recolectada. Por esta razón, el tipo y la cantidad de sensores que posee un robot depende del uso que se le esté dando y de la tarea para la que fue diseñado cumplir.

El convertidor analógico digital, que también se le llama ADC por sus siglas en inglés, consiste en un dispositivo encargado de comunicar los sensores con el controlador del robot. Físicamente es una tarjeta de interfaz de computadora dentro del CPU. Éste transforma la señal analógica creada por el sensor en una señal digital, es decir, una señal de 1 o 0, con el objetivo de procesar la información para que esta pueda ser utilizada por el robot. [1]

2.1.4.3 Subsistema de control

En un subsistema de control se encuentran dos dispositivos principales, el controlador y el convertidor digital analógico. El controlador digital corresponde a un elemento electrónico que va a tener una unidad de procesamiento central, una memoria y, de ser necesario, algún tipo de almacenamiento como un disco duro. Este almacenamiento se utiliza en la mayoría de los casos para almacenar los datos procesados y la información recolectada una vez que fue utilizada. El objetivo del controlador es, como dice su nombre, controlar los movimientos del manipulador y del efecto final. Esto se logra ya que procesa los comandos dados por el usuario y transmite señales a los actuadores para generar un movimiento apropiado. En general, estos controladores corresponden a computadoras y los lenguajes de programación son los mismos que se usan en la computación como C, C++, BASIC, Python entre otros. [1]

El convertidor digital analógico convierte la señal digital creada por el controlador en una señal analógica que puede ser enviada a los actuadores. El esquema general de control de un robot se encuentra en la figura 9. Además de estos elementos y en la mayoría de los casos, se implementa en el subsistema de control un amplificador que se encarga de tomar las señales análogas generadas por el convertidor y amplificarlas ya que estas señales tienden a ser débiles. Sin el amplificador las señales análogas no serían capaces de accionar los motores o actuadores del manipulador y el efecto final del robot. [1]

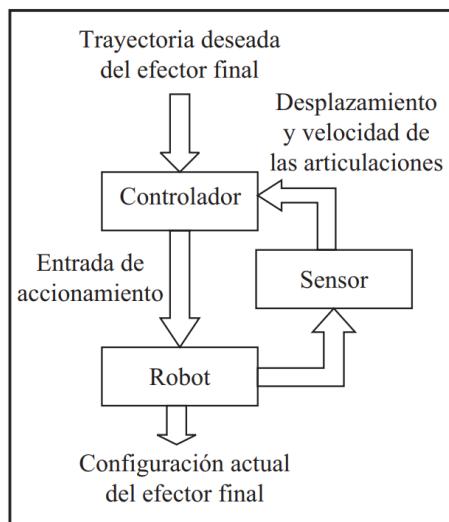


Figura 9. Esquema de control típico de un robot. [1]

2.2 Robótica Educativa

Durante esta sección se busca explorar la robótica educativa con el objetivo de generar una imagen clara de lo que se busca de un robot educativo. Además, se explora su importancia y su historia para comprender el proyecto de forma completa.

2.2.1 Definición

Cuando inició el desarrollo de la robótica, los robots creados estaban destinados a ser únicamente herramientas que llevaban a cabo diferentes funciones dentro de un proceso automatizado y, por ello, no se consideraba que los robots fueran inteligentes. En la actualidad, se ha trabajado en el desarrollo de robots inteligentes que efectúan funciones usando como indicadores la información que pueden procesar del medio en el que se encuentran. [2]

La robótica educativa o robótica pedagógica es el resultado de la expansión de la robótica para englobar una gama aún más amplia de disciplinas, lo cual fue posible al generar robots inteligentes. Esto llevó al proceso en el que se busca utilizar la robótica en el campo de la educación, lo que corresponde a la robótica educativa [2]. Por ende, para el desarrollo de esta investigación se define a la robótica educativa como la aplicación de la ciencia de la robótica en el área de la educación.

La primera utilización de la robótica educativa aparece en 1975, ya que se buscaba desarrollar un sistema automatizado de control que se encargara de administrar las experiencias de laboratorio en el campo de la psicología. Gracias a esto se desarrolló el primer encargado-robot que corresponde a un robot que permite que los estudiantes configuren una gran cantidad de experiencias al cambiar los parámetros por medio de una computadora. Además, una vez que los estudiantes inician la experimentación, el robot es el encargado de controlar el desarrollo de la experiencia, adquirir los datos y presentar los resultados en tablas y figuras. Este proceso le permite al estudiante desarrollar una progresión de forma óptima dentro de la exploración del dominio del conocimiento de estudio. [2]

Este primer ejemplo de la robótica educativa muestra su implementación en un ambiente de educación superior, pero en el presente la robótica educativa es implementada a estudiantes de todas las edades. Debido a la naturaleza reprogramable de los robots, esos pueden ser implementados en la enseñanza, variando el nivel de dificultad; además, diferentes robots pueden ser diseñados para el respaldo de diferentes áreas de enseñanza.

2.2.2 Importancia de la robótica educativa

Se puede pensar en muchas ocasiones que la robótica educativa sirve como un ejemplo del uso de la tecnología como un recurso didáctico, pero este no es su único objetivo. Utilizar la tecnología como un medio para promover los modelos de educación tradicionales es una práctica que ignora la verdadera innovación que trae la robótica educativa. Por esta razón, durante la implementación de robots en la educación, especialmente en escuelas y colegios, se busca poner un foco en los procesos de aprendizaje más que en los contenidos didácticos. [9]

Por esta razón, cuando se implementa la robótica educativa al asignar el proyecto de construcción o programación de un robot para la ejemplificación de un tema estudiado, el

proceso de desarrollo del proyecto va a ser de mayor ayuda para demostrar el tema por aprender que la presentación del resultado final. Esto se debe al proceso cognitivo por el que pasan los estudiantes al construir y programar un robot, que es el resultado de retar su potencial intelectual, afectivo y manual. [9]

El fundamento de esta perspectiva está basado en la teoría de aprendizaje llamada construcciónismo, la que plantea que los seres humanos crean estructuras de conocimiento de forma independiente de las circunstancias de su aprendizaje. Además, estas estructuras son especialmente construidas bajo la condición de que el individuo está conscientemente construyendo una entidad pública, la cual puede ser representada como una maqueta, un castillo de arena o un robot. [9]

Además, se ha demostrado que la implementación de la robótica educativa puede impactar otras áreas del aprendizaje y desarrollo personal de forma positiva como las habilidades sociales, el pensamiento creativo, la toma de decisiones, la resolución de problemas, la comunicación, la distribución de tareas y el trabajo en equipo. Todas estas habilidades son vitales para el futuro de los estudiantes en su vida personal y de trabajo, sin importar el área en el que se vayan a desarrollar. [10]

Por ende, se puede decir que la importancia de la robótica educativa se centra en dos pilares: El primero de ellos es la incorporación de la tecnología en los ambientes de estudio desde una temprana edad, lo que facilita el entendimiento de la tecnología actual y mejora la habilidad para adaptarse a su constante evolución. El segundo pilar es el énfasis que la robótica educativa le da a el proceso de aprendizaje, que no solo facilita la comprensión de un tema en particular, sino que beneficia al estudiante mediante la construcción de estructuras de conocimiento.

2.3. Estado del arte

Durante esta sección se busca establecer un entendimiento completo de los robots educativos más relevantes para el proyecto por cuanto su diseño es similar al Atta-Bot original; porque se encuentran disponibles para ser comprados en Costa Rica o porque su diseño de hardware abierto al público puede ser construidos en el país.

2.3.1 Robot Sphero Mini

El robot Sphero mini corresponde a un pequeño robot caracterizado por su forma esférica que posee diferentes elementos de la robótica que pueden ser controlados por medio de su software. Entre sus componentes se encuentran luces LED, servomotores, giroscopio y acelerómetro. Su software puede ser programado por medio de bloques o JavaScript. Además de esto también posee una aplicación llamada Sphero Edu por la cual se puede llevar a cabo su programación, disponible en dispositivos móviles y en computadoras, usando una extensión de Google Chrome. [11]

Sphero describe a este robot como una forma imaginativa y divertida para aprender las bases de la programación, perfecto para principiantes. Además de la aplicación mencionada anteriormente que se conecta por medio de bluetooth, también posee una biblioteca extensa de actividades diseñadas para el estudio de la computación pero que además refuerza temas de otras materias estándar como las matemáticas, la ciencia y la historia. Esta es una gran ventaja ya que permite que los educadores incorporen fácilmente el robot a sus lecciones. En la Figura 10 se encuentra el robot Sphero Mini. [12]



Figura 10. Robot Sphero Mini. [12]

2.3.2 Robot Zowi

El robot Zowi fue creado para el uso educacional y se caracteriza por tener altas capacidades de expansión. Es controlado a partir de una aplicación que se conecta con tecnología bluetooth, solo disponible en Android. Además, también puede ser controlado de forma elemental por medio de botones físicos que son reprogramables. [13]

La razón por la que se dice que tiene una alta capacidad de expansión es porque el robot está diseñado para tolerar cambios a sus partes con impresoras 3D, por lo que su exterior puede ser fácilmente modificado. Por otro lado, se puede abrir y manipular las conexiones y los componentes internos como la placa controladora, los sensores y los actuadores. También es importante ver que el robot no posee ruedas, en su lugar posee dos patas. En la Figura 11 se muestra un robot Zowi. [14]



Figura 11. Robot Zowi. [14]

2.3.3 Robot Mindstorms de Lego

En este caso en particular, el producto vendido por Lego no corresponde a un robot, sino a un equipo de montaje en el que se encuentran múltiples piezas y unidades programables de control que permiten la construcción de múltiples tipos de robots. Hay diferentes equipos disponibles pero el Ev3 corresponde al modelo creado específicamente para la robótica educativa. [10]

Entre las partes que trae cada equipo se tienen engranajes, bandas, ruedas, eslabones y conectores. En el área de sectores se busca enfocarse en aquellos que le provean al robot información de la luz, el tacto y el sonido, así como sensores infrarrojos para la evasión de obstáculos. Además, se proveen múltiples tipos de motores para que estos se incorporen en diferentes partes de los robots. Pero el componente más importante corresponde al ladrillo Ev3, que corresponde a una pequeña computadora que controla los motores y sensores del robot. [10]

Debido a que estos equipos no tienen una forma cerrada o correcta de crear un robot, son especialmente ideales para fomentar la creatividad de los estudiantes y la capacidad de solucionar problemas. Su variabilidad es su mejor característica ya que le permite ser un robot educativo con gran valor en todos los niveles de educación y en todos los tipos de clases tradicionalmente impartidas [10]. En la Figura 12 y Figura 13 se encuentran ejemplos de los robots que pueden ser construidos usando Lego Mindstorms.



Figura 12. Robot humanoide construido con Lego Mindstorms. [15]



Figura 13. Robot móvil construido con Lego Mindstorms. [15]

2.3.4 Robot Bee-Bot

Este robot educativo se caracteriza por su sencillez debido a que no cuenta con ningún sensor. La única información que recibe el robot del medio ambiente es mediante el uso de botones. Cada uno de estos botones representa un bloque de programación o una instrucción y pueden ser presionados secuencialmente y el robot guardara esta información hasta que se presione el botón de inicio, a partir de este momento realizara las instrucciones dadas. Estos botones tienen instrucciones fijas y no están creados para ser reprogramados por el usuario. [16]

Este robot en particular está diseñado para el uso en aulas de educación primaria y esa es la razón por la que la programación está simplificada, no hay necesidad de una aplicación para controlar al robot ni hay introducción directa de lenguajes de programación, que es una de las características más deseadas y vitales de la robótica educativa. Para llenar este vacío se utilizan tarjetas impresas de comandos que se pueden acomodar secuencialmente para simular la construcción de un código. En la figura 14 se encuentra el robot Bee-Bot.



Figura 14. El robot Bee-Bot. [16]

2.3.5 Robot Thymio

El robot Thymio es descrito como un robot móvil de hardware abierto que fue diseñado con los objetivos de mantener un bajo precio, ser atractivo para personas de distintas edades y géneros, tener un diseño mecánico que promueva la creatividad, una combinación de sensores y actuadores ideal para el aprendizaje, una serie de programas que permitan inspirar a los docentes y una comunidad abierta de información y conocimiento. [17]

Uno de los objetivos más importantes al diseñar el robot Thymio corresponde a la facilidad de aprendizaje mediante el uso del robot. Por ejemplo, se añade una luz LED por cada elemento, los cuales se encargan de mostrarle al usuario el momento preciso en el que un

censor lee una señal o un actuador se activa. También estas luces pueden indicar mensajes o señalar el inicio o el final de una tarea. A nivel de software, el robot también se enfoca en el aprendizaje creando múltiples ambientes de programación para que principiantes sean capaces de conocer poco a poco los aspectos básicos de la programación [17]. En la figura 15 se encuentra el robot Thymio.



Figura 15. Robot Thymio. [17]

2.3.6 ELEGOO Smart Robot Car

Este robot en particular es vendido en forma de un kit educativo. Este está diseñado para principiantes con el objetivo de generar prácticas sobre la programación, el ensamblaje de electrónica y el conocimiento de robótica. El kit contiene 24 tipos de piezas del módulo que incluye evitar obstáculos, módulos de trazo de línea y control remoto infrarrojo. También puede ser conectado a aplicaciones y es de ensamblaje modular. Además, cuenta con un manual de ensamblaje que incluye programaciones básicas para su uso.

2.3.7 Comparación de los robots educativos

Los robots educativos desarrollados anteriormente poseen diferentes ventajas y desventajas que definen cuál es su uso ideal. Para un mejor entendimiento de cómo estos robots se comparan entre sí, se muestra a continuación la Tabla 1.

Tabla 1. Comparación de los robots educativos.

Características	Robot Sphero Mini	Robot Zowi	Robot Mindstorms de Lego	Robot Bee-Bot	Robot Thymio
Dimensión	4 x 4 x 4 cm [12]	14 x 9.3 x 15 cm. [14]	Versátil dependiendo de la construcción. [17]	10 x 10 x 10 cm. [16]	5 x 11 x 11 cm. [17]
Movilidad	Ruedas (robot esférico) [12]	Piernas. [14]	Versátil (ruedas, piernas, estático) [17]	Ruedas. [16]	Ruedas. [17]
Precio	\$50 [12]	\$108	\$370 [17]	\$98 [16]	\$189 [17]
Sensores	Giroscopio, encoders y acelerómetro. [12]	Sensor de sonido, sensor ultrasónico, botones. [14]	Giroscopio, sensor ultrasónico, sensor de color y sensores táctiles. [17]	Botones. [16]	Micrófono, sensores de proximidad, acelerómetro, seguidores de líneas, sensor de temperatura y botones. [17]
Actuadores	Motor y LED. [12]	Motores y altavoz. [14]	Servomotores medianos y grandes. [17]	Motores. [16]	Motores, altavoz y LED. [17]
Complejidad	Media [12]	Media [14]	Alta. [17]	Simple [16]	Alta [17]
Forma de programación	Tres lenguajes: Texto, dibujo y bloques. Conexión Bluetooth y aplicación móvil. [12]	Dos lenguajes: bloques y control remoto. Conexión Bluetooth y aplicación móvil. [14]	Lenguaje de programación de bloques. Conexión Bluetooth y aplicación móvil. [17]	Programación mediante botones. [16]	Compatible con múltiples lenguajes de programación. Conexión mediante Dongle con interfaz. [17]

2.3.8 Robótica educativa en Costa Rica

En los últimos años se ha dado un cambio en el modelo de producción del país, con las exportaciones de alta tecnología y desarrollo de software superando las exportaciones de

productos tradicionales. Por esta razón, se ha puesto un énfasis en el impulso de estos sectores, lo que implica la enseñanza de las ciencias y la tecnología como un imperativo estratégico. [18]

Lastimosamente, si se analiza la educación primaria y secundaria, aun se considera insuficiente la promoción de una cultura científica. El mayor esfuerzo de parte del MEP se basa en la creación de los Colegios Científicos Costarricenses, los cuales fueron fundados a finales de los ochenta con el objetivo de preparar a los jóvenes de 15 a 18 años para estudios universitarios con fundamentos de matemáticas, física, química, biología e informática. A pesar del éxito de estos colegios, su población estudiantil solo representa un 1% de la población total de estudiantes en el país, por lo que es necesario la implementación de nuevas estrategias en las áreas de la ciencia, tecnología, ingeniería y matemáticas STEM por sus siglas en inglés, para escuelas y colegios públicos. [18]

Otro de los mecanismos que ha implementado el sistema de educación pública consiste en la implementación de ferias científicas en todas las escuelas y colegios del país, cuyo objetivo es introducir el método científico a estudiantes de todas las edades para desarrollar las vocaciones científicas y tecnológicas. Además, se busca implementar educación por medio de los procesos de investigación. [18]

Aparte de estos dos grandes pilares, Costa Rica incorpora programas de robótica educativa en algunos colegios y escuelas. En este sector es importante reconocer el esfuerzo realizado en el país a la hora de incorporar en la educación tecnologías digitales. En el año 1988, Costa Rica se convirtió en el primer país de América Latina en implementar un programa nacional de informática educativa, lo cual implicó proveer la infraestructura tecnológica adecuada y un equipo de docentes entrenados en el uso de tecnologías en una gran parte de la escuelas y colegios públicos, lo que fue realizado con el apoyo de la Fundación Omar Dengo a través del Centro de Innovación Educativa. [18]

En la actualidad existe una amplia oferta de procesos educativos para impulsar la robótica educativa promovida por otras organizaciones privadas con o sin fines de lucro como la Universidad de Costa Rica, el Instituto Tecnológico de Costa Rica, la Universidad Nacional y otros centro educativos privados. Sin embargo, uno de los puntos débiles de estos

programas es su inaccesibilidad para las poblaciones ubicadas fuera de Gran Área Metropolitana. [18]

Los proyectos de robótica educativa impartidos en el país han mostrado resultados positivos, mediante la implementación de un marco basado en la creación de ambientes pedagógicos que apoyen la construcción mental. En los primeros años de estos proyectos se buscaba que los estudiantes determinaran las temáticas específicas de mayor interés y a partir de esto se apoyaba la exploración del tema, lo que dio como resultado el desarrollo de robots individuales, creados para una acción particular y que no interactuaban fuera de ella. Debido a esta tendencia se desarrolló un programa en el que se reorienta la elección y definición de los temas escogidos para crear un proceso de aprendizaje que profundice no solo las áreas de robótica sino también los temas de enseñanza tradicional. [19]

Con esta estrategia se obtuvieron resultados en los cuales los estudiantes presentan mayores niveles de compresión de los temas seleccionados a partir de los proyectos de robótica. Además, este tema se ve mejor representado en sus creaciones debido a una forma más adecuada de guiar el proceso creativo de parte de los docentes. Por último, el desarrollo de estos proyectos lleva al fortalecimiento de diferentes habilidades que son solo conseguidas a través de la práctica, como puede ser la capacidad de investigación, las habilidades motoras o el trabajo en equipo. Todas estas áreas han demostrado ser una parte vital de los programas establecidos en Costa Rica y han sido destacados por los docentes como los factores en los que más se distingue el avance en los estudiantes. [19]

En conclusión, en los casos en los que se han podido impartir diferentes talleres de robótica educativa en colegios y escuelas del país, se ha visto un verdadero impacto en la población estudiantil. En algunos casos este corresponde a un elevado interés en el área de tecnología, pero también puede ser un mejor desarrollo de habilidades motoras, capacidad de investigación y procesamiento de información. Esto sirve de justificación para ampliar estos programas para cubrir más áreas del país y lograr un mayor impacto en la población estudiantil costarricense. [19]

Capítulo 3.0 Metodología

Durante este capítulo se busca exponer la metodología que fue utilizada para desarrollar la propuesta de diseño del proyecto. Para esto se aplica la metodología de Ulrich y Eppinger, siguiendo cada uno de sus pasos hasta llegar a un concepto ganador.

3.1 Diagnóstico del Atta-Bot y definición del problema

El primer paso para resolver de forma satisfactoria un problema es poder definirlo. Esta etapa es vital por múltiples razones, en primer lugar, corresponde al primer filtro de sesgos en la metodología de Ulrich y Eppinger ya que el establecimiento del problema debe de seguir ciertas reglas, eliminando posibles sesgos del cliente y del individuo al aplicar la metodología. Los dos tipos de sesgos que deben evitarse en esta etapa son denominados el sesgo del “problema percibido” y el sesgo de la confusión entre problema y técnica de solución. El primero de estos corresponde a identificar a uno de los síntomas del problema como la causa, el segundo representa el tipo de sesgo en el que la declaración del problema está tomando en cuenta el uso de un elemento en particular para su solución. [20]

Por esta razón, se decidió realizar una lista de ventajas y desventajas del diseño actual del Atta-Bot desde el punto de vista de que este sea usado para una aplicación de robótica educativa. Esto no solo permite diagnosticar el Atta-Bot actual, sino que también ayuda a reconocer el verdadero problema que se busca resolver, el cual corresponde al primer paso de la metodología. Además, esta lista ayuda a cumplir el primer objetivo planteado del diagnóstico del robot. Para poder realizar este análisis se debe de comprender su diseño actual por lo que se estudió la documentación de éste, así como su código y sus componentes.

En primer lugar, se buscó definir exactamente el concepto detrás del robot actual, resultando que el Atta-Bot corresponde a un prototipo de robot terrestre que fue construido con el objetivo de ser aplicado dentro de la robótica de enjambres en grupos, en el que resuelven tareas cooperativas. El robot se caracteriza por su bajo costo, construcción simple y documentación abierta de su software y hardware. La construcción del Atta-Bot utiliza solo componentes electrónicos preensamblados que pueden ser adquiridos fácilmente por medio de proveedores locales, además que su sistema mecánico está construido con técnicas de prototipado. En términos de estructura eléctrica, el robot actual no es modular, sino que posee

una única placa de circuito impreso que conecta las diferentes partes entre sí y al cerebro del robot.

Algunas de las características generales del robot pueden ser encontradas en la Tabla 2. [21]

Tabla 2. Características Generales de los Robots Atta-Bot. [21]

Característica	Valor
Dimensiones	155 x 178 x 105 mm
Autonomía	>45 min
Masa	455.3 g
Velocidad	108 mm/s
Costo de partes	≈ \$216.18

El costo de las partes presentado en la tabla es diferente del que se expresa en la documentación del robot, debido a que este nuevo costo refleja los cambios de precio que han ocurrido en el mercado desde la producción de los prototipos originales del Atta-Bot. Por esta razón, para realizar un análisis apropiado del robot se realizó un nuevo cálculo del costo, tomando en cuenta la lista inicial de partes del robot y sus actuales precios conforme a la información existente en los enlaces a los sitios web de los proveedores encontrados en la documentación del proyecto PROE. También es importante destacar que, aunque se puede obtener un precio aproximado de los componentes del robot, este no necesariamente refleja su precio total ya que no se considera el costo de producción, ensamblaje o diseño. La lista de componentes, la cantidad de cada uno y su precio se presentan en la Tabla 3.

Tabla 3. Componentes eléctricos del robot Atta-Bot actual.

Cantidad	Componente	Precio
1	Batería de litio de 3.7 V y 2500mAh	\$17.95
1	PowerBoost de 5V y 500mA	\$10.95
1	Driver de puente H de 1A Dual	\$6.95
2	Motor DC con enconder y rueda N20	\$14.95
2	Suporte de motor	\$0.75
1	Microcontrolador Adafruit Feather M0 RFM69 Packet Radio	\$24.95
1	Ball Caster	\$2.75
1	Módulo de magnetómetro de triple eje	\$9.95
1	Microcontrolador Blue Pill STM32	\$16.95
1	Micro Servomotor MG90S	\$6.95
1	Módulo de sensor infrarrojo para evadir obstáculos	\$2.95
2	Módulo de sensor de seguimiento	\$1.95
1	Sensor de temperatura	\$1.95
1	Sensor de medición de distancia	\$16.95
1	Set de pines	\$1.95
1	Interruptor general	\$1.25
1	Interruptor pequeño	\$1.25
3	LED de color	\$0.15
1	Módulo de triple eje de giroscopio y acelerómetro	\$6.95
1	Memoria EEPROM de doble cable	\$1.40
3	Capacitor de poliéster metalizado	\$0.33
3	Header hembra 2.54mm de 40 pines 1 fila	\$0.70
2	Conecotor de motor hembra JST-ZH 1.5mm 4 pines	\$1.50
1	Conecotor de la batería macho JST-PH 2mm 2 pines	\$1.50
1	Base DIP de 8 pines	\$0.99
1	Programador Blue Pill	\$9.95
1	Cable USB microB de 80 cm	\$4.95
1	Kit de espaciadores y tornillos	\$15.95
1	Kit de resistencias	\$8.95
Total		\$216.18

A continuación, se busca describir el diseño del Atta-Bot para comprender la lógica detrás del prototipo y crear un diagnóstico de su compatibilidad con actividades educativas.

En términos del diseño electrónico, los diseñadores eligieron una arquitectura que usa como base dos microcontroladores diferentes, el controlador Adafruit Feather M0 RFM69 se encarga del control de movimientos, comunicación y comportamientos de alto nivel para el enjambre y el controlador STM32 “Blue Pill”, que maneja los sensores externos y la detección de obstáculos. El uso de ambos microcontroladores ayuda a ampliar la capacidad del robot. El primero de los microcontroladores fue seleccionado por su capacidad de comunicación por medio de radio FM y el segundo fue elegido debido a ser un controlador comúnmente usado y de fácil acceso, además de económico. [21]

Debido a que se manejan múltiples tensiones por la variedad de componentes eléctricos usados en el robot se eligió una batería de Litio con una tensión de 3.7V y se complementó con un “PowerBoost” que corresponde a un componente que ayuda a amplificar el voltaje a 5V para los componentes cuya alimentación debe de ser mayor. [21]

En cuanto a los sensores, se cuenta con dos sensores infrarrojos en la parte inferior para la detección de bordes o agujeros en el piso. Se tiene un tercer sensor infrarrojo al frente del robot para la detección de obstáculos. Se implementó un sensor SHARP GP2Y0A21YK para la detección de obstáculos periféricos, ya que tiene un rango mayor de detección que los sensores infrarrojos: este sensor se montó en un servomotor que le permite apuntar a cualquier dirección en un rango de 180°. También se incluye en el diseño un sensor de temperatura para la medición de variables ambientales. [21]

Se tienen además una serie de sensores que se califican como internos ya que miden las variables del robot. Cada uno de los motores posee un codificador magnético conocido como encoder que permiten el control de la dirección de giro y la velocidad de los motores para que estén en sincronia. Un magnetómetro funciona como una brújula digital para medir la magnitud de los giros realizados por el motor y un IMU o medición inercial realiza las mediciones del giroscopio. Por último, se tiene una memoria EEPROM en la que se almacenan los datos de calibración de los sensores. En la figura 16 se muestra el diagrama de alto nivel de las dispositivos electrónicos usados en el robot actual. [21]

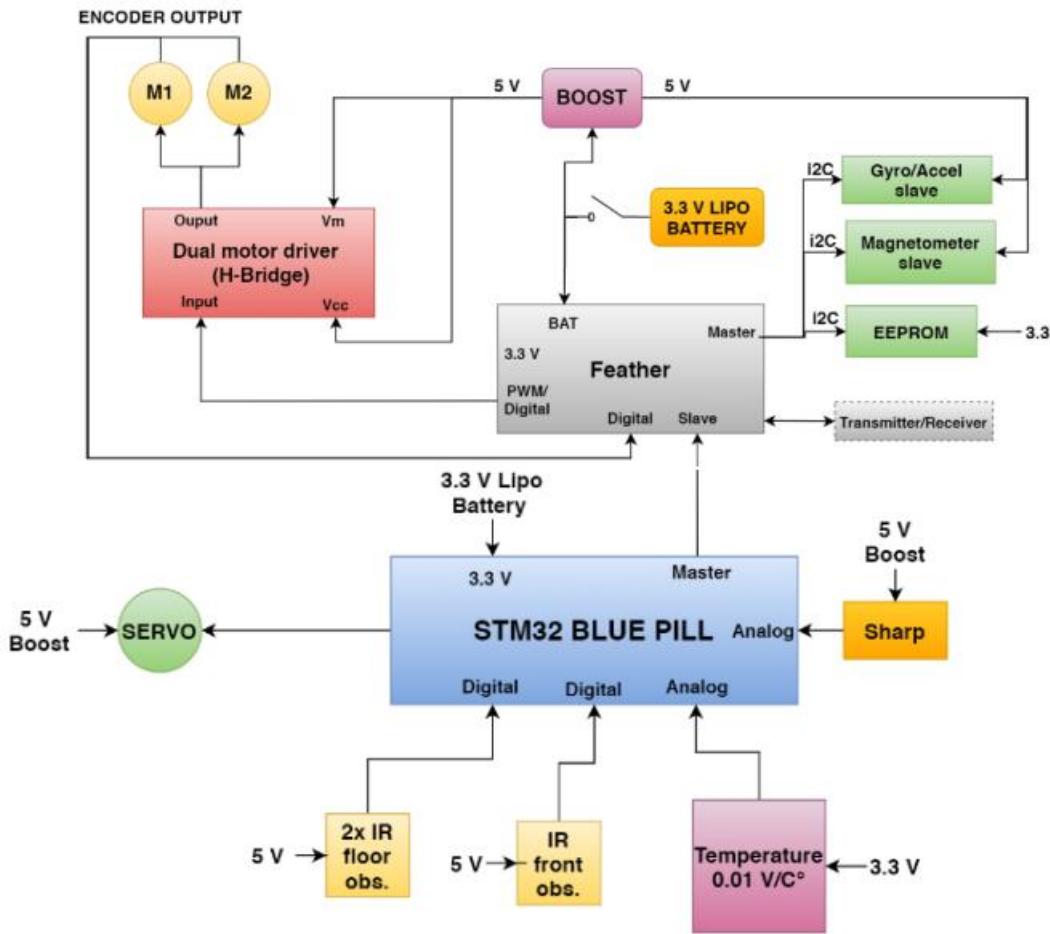


Figura 16. Diagrama de alto nivel de las dispositivos electrónicos usados en el robot actual.

[21]

Además, como se mencionó anteriormente, la estructura mecánica fue realizada con técnicas de prototipado, lo cual involucró la selección de los materiales. Por esta razón la carcasa fue creada con un acrílico de 3mm de grosor del cual se obtuvieron las partes con la técnica de corte láser. El ensamblado se complementó con un kit de separadores y tornillos. Los Attabot son robots diferenciales, lo que quiere decir que poseen dos actuadores rotatorios y un tercer punto de apoyo, que en este caso corresponde a un “BallCaster”. [21]

En términos del código creado para el movimiento de los robots, se generaron dos lazos de control que actúan sobre cada motor. El primer lazo de control es usado para la posición en cada rueda que es vital para el control de los giros del robot, este se realimenta con los valores recolectados por el magnetómetro, el giroscopio y los encoders. Por otro lado, el segundo

lazo genera un control de la velocidad usando los datos de ésta recolectados por los encoders y los usa para mantenerla en ambos motores lo más constante posible para obtener un movimiento rectilíneo de parte del motor. Ambos controladores son de tipo PID y sus constantes fueron calculadas experimentalmente. [21]

La comunicación del robot está basada en un protocolo de Radio Frecuencia. Esto puede ser generado gracias al módulo de esta comunicación que posee el controlador Feather M0 RFM69. Este protocolo fue elegido debido a que los robots están creados para ser usados en ambientes de experimentación poco controlados, por lo que tecnologías como Wi-Fi y Bluetooth no son compatibles. La tecnología de Radio Frecuencia permite la generación de topologías variadas y rangos de alcance de señal grandes lo que minimiza la dependencia de infraestructura externa a los robots. [21]

La última etapa antes de realizar el diagnóstico del robot corresponde a describir las acciones que éste realiza y el comportamiento obtenido a través del código.

El robot está creado para generar rutas óptimas al ser usado en grupos con otros robots idénticos, esto se valida al colocar un grupo de Atta-Bots en un laberinto estático, entonces ellos analizan el laberinto y se comunican entre sí y, una vez que han recolectado suficiente información mientras evitan obstáculos que incluyen paredes y otros robots, pueden identificar la ruta óptima.

Para que el robot siga un comportamiento de exploración se implementó el algoritmo de caminado aleatorio, el cual fue probado y validado inicialmente en simulaciones. Algunos ejemplos de los escenarios de validación usados para los robots se encuentran en la Figura 17 y la Figura 18. [21]

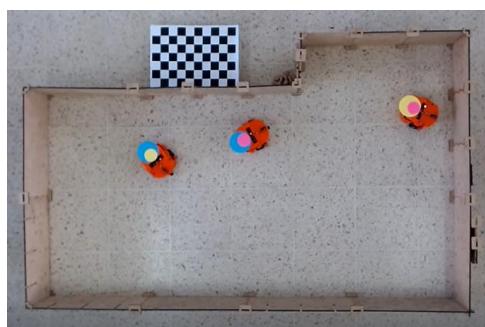


Figura 17. Primer escenario de validación de los Atta-Bot. [21]

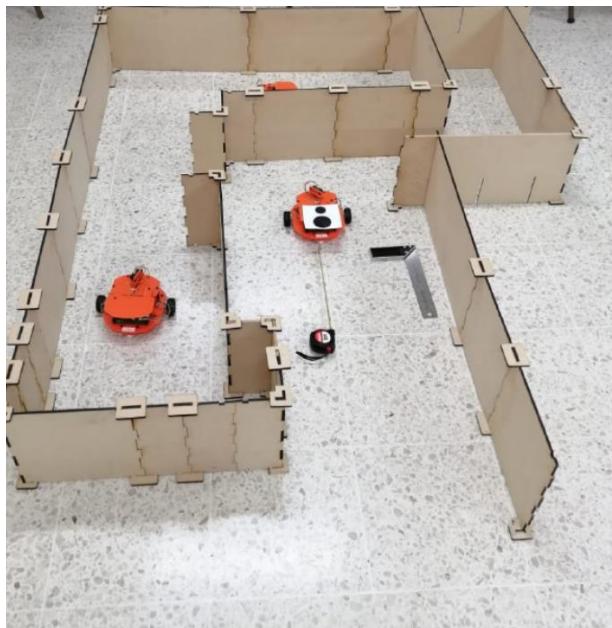


Figura 18. Segundo escenario de validación de los Atta-Bot. [21]

Al analizar el código del robot, cuyo lenguaje corresponde a Arduino IDE, se encontró que su base corresponde a una máquina de estado que se actualiza cada vez que se cumple el tiempo de muestreo. Esta ayuda a definir la acción que realiza el robot, ya sea avanzar, girar, detenerse, enviar información, mover la posición del sensor SHARP o activar la acción realizada en caso de la detección de un obstáculo, entre otras opciones. Además de esto, en términos del movimiento del robot, se encontró que para ambos lazos de control PID, aunque se estableció anteriormente que se utilizan el magnetómetro, giroscopio y los encoders, el sensor de principal impacto en ambos corresponde al set de encoders debido a que la función de fusión de los datos de estos sensores elimina o minimiza los valores de los otros dos sensores debido a errores causados por ellos en la validación.

Con este análisis del código se concluye la recolección de información, por lo que se realizó la construcción de la lista de ventajas y desventajas porque, como se mencionó anteriormente, esta será enfocada en un análisis de la capacidad de robótica educativa del prototipo actual, así como su simplificación general, lo que significa que los aspectos que serán considerados como desventajas también incluyen aquellas partes del diseño en el cual el robot esté sobredimensionado para la nueva aplicación. A continuación, se presenta la lista de ventajas del Atta-Bot original.

- El sistema de movimiento del robot es preciso, realiza giros y avanza a distancias dentro del margen ideal.
- La autonomía del robot es ideal al coincidir con el tiempo de una lección en el sistema de educación secundaria que corresponde a 45 minutos.
- El precio del robot, incluso una vez es actualizado con los precios actuales, es bajo.
- El robot es diseñado utilizando componentes accesibles por medio de distribuidores locales, por lo que estos son de fácil acceso y, además, esto representa un menor tiempo de espera para su producción.
- La carcasa y otros niveles de hardware utilizan métodos de construcción de prototipado, por lo que son simples de crear y sus partes son fáciles de reemplazar.
- Los detalles del software y hardware están abiertos al público para ser recreados.
- El código está estructurado mediante una máquina de estados, lo que facilita el manejo de la información y el desarrollo de los casos necesarios.
- La detección de obstáculos, el movimiento de avance, retroceso y giro y la incorporación de un servomotor, son todas funciones ideales para la robótica educativa.
- El código del robot está programado en un lenguaje altamente accesible, por lo que esto puede ser utilizado también en aplicaciones educativas.

Por otro lado, también se tiene la lista de desventajas que se comparte a continuación.

- El tamaño del robot es mayor al deseado en términos de robótica educativa, ya que un menor tamaño permite el uso sobre escritorios y aprovecha de mejor forma el espacio del piso de las aulas.
- El sistema de movimiento del robot, a pesar de ser preciso, se encuentra sobredimensionado para lo necesario en términos de robots educativos. Además, como se mencionó anteriormente, incluso en el diseño actual muchos de los componentes no representan un papel vital en el movimiento.
- El uso de dos microcontroladores también es uno de los aspectos que se encuentra sobredimensionado para un robot educativo, ya que este tipo de robótica no necesita la capacidad de procesamiento de datos actual.

- El protocolo de comunicación utilizado en la actualidad no es ideal para un robot educativo porque si se busca una conexión con el robot con una aplicación móvil, tanto la estabilidad de la conexión como la facilidad para ser realizada son afectadas con la elección de la tecnología de Radio Frecuencia. Las ventajas de este protocolo se pierden en aplicaciones educativas debido a que, en este caso, se usan los robots en un ambiente controlado.
- La construcción de la arquitectura electrónica está basada en soldar los componentes a una placa diseñada para las conexiones, el cual es un proceso complejo y no puede ser realizado sin un conocimiento previo en soldadura de componentes.
- El robot no es modular, lo que significa que si uno de los componentes es dañado este no será fácil de reemplazar.
- No se posee una conexión a una aplicación móvil que controle el robot o que cree una forma simplificada de su programación, lo que complica su uso dentro de las aulas.
- El sistema de reconocimiento de obstáculos esta sobredimensionado para el uso de aplicaciones educativas porque el uso del sensor SHARP es ideal en ambientes de exploración, pero los valores que genera no son necesarios para aplicaciones educativas.
- El robot no posee ningún actuador que pueda ser utilizado para tareas educativas aparte del sistema de movimiento de los motores CD.
- No se poseen sensores que estén enfocados para ser usados en aplicaciones educativas, debido a que todos los que tiene están enfocados en controlar el movimiento y reconocer el ambiente para la generación de rutas.

Una vez generadas estas listas, se puede definir el problema que se busca resolver mediante la aplicación de la metodología ya citada y que en este caso es que, el Atta-Bot actual no posee un diseño ideal para un robot educacional.

Después de este análisis, que incluyó añadir y descartar ventajas y desventajas, se concluye que el problema está planteado de forma correcta e ideal para este proceso y caso; además, la definición del problema no está siendo afectada por sesgos.

Se descartan también como sesgos, conforme a la lista de ventajas del Atta-Bot, las ideas de que el rediseño del robot no es una opción válida frente a la posibilidad de iniciar el diseño

de un robot educativo desde cero, así como que el Atta-Bot no debe ser rediseñado para un uso académico ante la existencia de algunas desventajas que muestran falta de capacidad del robot en este ambiente.

En conclusión, el problema puede ser definido solo después de la comprensión del prototipo y del ambiente en el que se desea incorporarlo. Los posibles sesgos deben ser identificados y eliminados durante este proceso por lo que se debe de descartar cualquier factor de consideración de la solución cuando se plantean los defectos del diseño actual y la problemática de este. Además, como en todos los pasos de la metodología, es necesario que este proceso sea revisitado múltiples veces hasta que sea perfeccionado mediante iteración.

3.2 Definición de las necesidades

Las necesidades representan la lista de deseos de un cliente que deben de ser tomadas en cuenta a la hora de desarrollar una solución para el problema formulado. De esta forma también pueden ser vistas como los estándares que deben ser cubiertos para que una solución, desde un punto de vista conceptual, sea válida. Establecer, comprender y jerarquizar estas necesidades es una parte vital para el desarrollo de una solución por lo que este paso fue desarrollado de forma rigurosa y evitando cualquier tipo de sesgo. [20]

Es importante que antes de investigar qué tipo de diseño busca el cliente, reconocer la descripción básica que se está buscando del producto. En este caso, se busca un robot educativo cuyo diseño esté basado en el robot Atta-Bot existente. Este robot debe de ser capaz de conectarse a una aplicación móvil desde la cual se programan diferentes instrucciones que después el Atta-Bot educativo ejecutara con precisión. Las primeras instrucciones corresponden al avance y retroceso en línea recta y son configuradas a una distancia exacta. Las siguientes instrucciones corresponden a los giros a la derecha y a la izquierda, este es un giro que realiza el robot sobre su propio eje y se puede configurar a un número de grados exactos.

Luego de esto, se tiene la detección de obstáculos que puede ser activada y desactivada dentro de las instrucciones de la aplicación. Si el robot detecta un obstáculo este se detiene, retrocede y reinicia la cadena de instrucciones, estos obstáculos pueden encontrarse frente al robot o ser una falta de superficie para que el robot siga su movimiento.

Por último, se busca que el robot ejecute ciclos de movimientos y, para ello, desde la aplicación se puede programar la cantidad de ciclos y los movimientos dentro de éstos. Además, se tienen una lista de componentes que serían ideales para aplicar en el futuro para las siguientes iteraciones del robot y se busca que esta primera iteración, en teoría, pueda alojarlos. Estos componentes corresponden a sensores de color, una garra y señalización por medio de pantallas.

Por otro lado, la carcasa será diseñada por otro de los investigadores del proyecto, por lo que las necesidades relacionadas con ella serán evaluadas desde el punto de vista de cómo ésta afecta el diseño electrónico o del código.

Al aplicar la metodología seleccionada se realizaron dos entrevistas para el establecimiento de las necesidades de la propuesta de diseño. La primera de ellas fue al ingeniero Juan Carlos Brenes Torres, investigador del proyecto PROE y asesor industrial del proyecto.

El objetivo de esta entrevista fue detectar sesgos de parte del cliente, así como recolectar información enfocada en el aspecto mecatrónico del diseño y conocer más a fondo las capacidades actuales del Atta-Bot desde la perspectiva eléctrica, computacional y mecánica.

La segunda entrevista fue realizada a las investigadoras del proyecto Cindy Calderón Arce y Rebecca Solís Ortega cuya formación está más inclinada al sector educativo. De esta entrevista se buscaba nuevamente cualquier tipo de sesgo de parte del cliente y la información necesaria para el establecimiento de las necesidades educacionales del robot, en base a sus futuras aplicaciones.

Una vez que se recolectó esta información, se establecieron una serie de necesidades iniciales. Estas fueron expuestas a una revisión de parte de los clientes para establecer su jerarquía de importancia, así como discutir a mayor profundidad aquellas necesidades que no fueron establecidas en el grupo inicial o posibles cambios en las que fueron presentadas. Este proceso fue iterativo, hasta encontrar la siguiente lista final de necesidades. Esta lista fue después jerarquizada en necesidades primarias y secundarias para un mejor entendimiento de su relación.

De esta forma se siguen los 5 pasos del proceso de diseño de necesidades establecidos en la metodología Ulrich y Eppinger. Estos pueden ser reconocidos siguiendo el siguiente orden:

recopilación con el cliente de los datos sin procesar, interpretación de los datos para la formación de necesidades, jerarquización de necesidades en primarias y secundarias, establecimiento del orden de importancia de la necesidades y, por último, la reflexión en el resultado final, así como el proceso en sí.[20]

A continuación, se encuentra la lista final de necesidades, antes de ser ordenada.

1. Se realiza un rediseño del robot actual llamado Atta-Bot.
2. El rediseño del robot simplifica el robot Atta-Bot actual.
3. El rediseño del robot está orientado a la generación de un robot educativo.
4. El precio del robot rediseñado es menor o igual al actual.
5. Se mantiene en gran parte el sistema de movimiento del Atta-Bot actual.
6. Se mantiene la calidad y precisión de los componentes actuales del Atta-Bot.
7. Se mantiene el tiempo de autonomía actual de Atta-Bot. (tiempo que dura la batería)
8. Se eliminan los componentes que no son relevantes para las aplicaciones educativas del robot.
9. El nuevo robot es capaz de vincularse a una aplicación móvil para ser controlado.
10. En el nuevo diseño se implementa un sensor de color.
11. En el nuevo diseño se implementa una garra.
12. Se mantiene un sistema de detección de obstáculos.
13. El nuevo robot realiza todas sus acciones de forma precisa para las aplicaciones educativas.
14. El nuevo robot es seguro, especialmente para su uso en escuelas y colegios.
15. El robot tiene una forma de mostrar que la acción está siendo realizada o finalizó (LEDs o pantalla)
16. En la carcasa se tiene una forma de señalizar el frente (una cara, una luz, etc)
17. La batería es fácil de cargar y se carga de forma intuitiva.
18. Se tienen botones para el inicio del programa y para cancelar el programa.
19. El nuevo diseño es modular (no está soldado en un solo lugar y los componentes pueden ser desconectados)
20. El nuevo robot es robusto

21. Se implementa un elemento de construcción a la hora de desarrollarse los procesos educativos.
22. El tamaño del nuevo robot es menor al diseño actual
23. El diseño del hardware y del software es de carácter abierto y accesible al público.

Esta lista de necesidades debe ser jerarquizada en aquellas primarias y secundarias, jerarquización que se muestra en la Tabla 4. Es importante destacar que lo que hace que una necesidad sea primaria es que ésta contiene criterios más generales, mientras que las secundarias son aquellos criterios más específicos del diseño. [20]

Tabla 4. Jerarquización de necesidades primarias y secundarias.

Necesidad Primaria	Necesidades Secundarias
Se realiza un rediseño del robot actual llamado Atta-Bot.	<ul style="list-style-type: none"> • El rediseño del robot simplifica el robot Atta-Bot actual. • El precio del robot rediseñado es menor o igual al actual. • Se mantiene en gran parte el sistema de movimiento del Atta-Bot actual. • Se mantiene la calidad y precisión de los componentes actuales del Atta-Bot. • Se mantiene el tiempo de autonomía actual de Atta-Bot. (tiempo que dura la batería) • Se mantiene un sistema de detección de obstáculos. • El tamaño del nuevo robot es menor al diseño actual
El rediseño del robot está orientado a la generación de un robot educativo.	<ul style="list-style-type: none"> • Se eliminan los componentes que no son relevantes para las aplicaciones educativas del robot.

	<ul style="list-style-type: none"> • En el nuevo diseño se implementa un sensor color. • En el nuevo diseño se implementa una garra. • El nuevo robot realiza todas sus acciones de forma precisa para las aplicaciones educativas. • Se implementa un elemento de construcción a la hora de desarrollarse los procesos educativos. • En la carcasa se tiene una forma de señalizar el frente (una cara, una luz, etc) • Se tienen botones para el inicio del programa y para cancelar el programa. • El nuevo robot es capaz de vincularse a una aplicación móvil para ser controlado.
El nuevo robot es seguro, especialmente para su uso en escuelas y colegios.	<ul style="list-style-type: none"> • El nuevo robot es robusto • El nuevo diseño es modular (no está soldado en un solo lugar y los componentes pueden ser desconectados) • La batería es fácil de cargar y se carga de forma intuitiva.
El diseño del hardware y del software es de carácter abierto y accesible al público.	

Como se mencionó anteriormente, se creó un orden de prioridad de las necesidades con base en la opinión de los clientes. A ellos se les pidió que clasificaran las necesidades formuladas del 1 al 5, con una instrucción detallada del significado de cada número. El 5 representa que la función o necesidad descrita es de importancia crítica y el cliente no consideraría un producto sin ésta. El 4 es usado para aquellas necesidades que son altamente deseables, pero el cliente consideraría una solución sin ellas. El 3 se elige si se considera que es una necesidad buena, pero no es necesaria. El 2 se usa para describir una función que no es necesaria pero no hay problema si se implementa. Por último, el 1 es usado para indicar que la función no es deseada. [20]

Debido a que la encuesta se hizo a dos personas y se desea darle el mismo peso a cada uno de los clientes, se decidió crear el orden de prioridad tomando las respuestas y creando un promedio. Debido a esto se obtuvieron prioridades calificadas con números con decimales, pero este método aún puede ser aplicado para el desarrollo de la metodología. Además, se debe agregar que durante la revisión de las necesidades de forma iterativa se implementó la necesidad correspondiente a "El diseño del hardware y del software es de carácter abierto y accesible al público.". Esta necesidad es importante debido a que la implementación de un software y hardware abierto es uno de los pilares del proyecto PROE, debido a que el software abierto permite la implementación del robot en las aulas a través del país.

En la Tabla 5 se presenta la jerarquía de prioridad de las necesidades.

Tabla 5. Jerarquización de las necesidades según su prioridad.

Clasificación	Necesidades
5	<ul style="list-style-type: none">• Se realiza un rediseño del robot actual llamado Atta-Bot.• El rediseño del robot está orientado a la generación de un robot educativo.• El nuevo robot es capaz de vincularse a una aplicación móvil para ser controlado.• El nuevo robot realiza todas sus acciones de forma precisa para las aplicaciones educativas.

	<ul style="list-style-type: none"> • El nuevo robot es seguro, especialmente para su uso en escuelas y colegios. • El diseño del hardware y del software es de carácter abierto y accesible al público.
4.5	<ul style="list-style-type: none"> • El rediseño del robot simplifica el robot Atta-Bot actual. • Se eliminan los componentes que no son relevantes para las aplicaciones educativas del robot. • El robot tiene una forma de mostrar que la acción está siendo realizada o finalizó (LEDs o pantalla) • La batería es fácil de cargar y se carga de forma intuitiva. • Se tienen botones para el inicio del programa y para cancelar el programa. • El nuevo diseño es modular (no está soldado en un solo lugar y los componentes pueden ser desconectados) • El nuevo robot es robusto
4	<ul style="list-style-type: none"> • Se mantiene la calidad y precisión de los componentes actuales del Atta-Bot. • Se mantiene un sistema de detección de obstáculos. • En la carcasa se tiene una forma de señalizar el frente (una cara, una luz, etc) • El tamaño del nuevo robot es menor al diseño actual
3.5	<ul style="list-style-type: none"> • Se mantiene el tiempo de autonomía actual de Atta-Bot. (tiempo que dura la batería) • En el nuevo diseño se implementa un sensor color. • En el nuevo diseño se implementa una garra. • Se implementa un elemento de construcción a la hora de desarrollarse los procesos educativos.
3	<ul style="list-style-type: none"> • El precio del robot rediseñado es menor o igual al actual. • Se mantiene en gran parte el sistema de movimiento del Atta-Bot actual.

En conclusión, según lo analizado mediante las entrevistas y diferentes conversaciones del cliente, así como los resultados de la encuesta y el diagnóstico anteriormente realizado de Atta-Bot, el cliente está buscando un rediseño del robot que mantenga sus características favorables, pero esté orientado a la robótica educativa.

A pesar de mencionar diferentes componentes y actividades en las que se puede incorporar el robot, el cliente en general mantiene una perspectiva abierta a la hora de su diseño, siempre y cuando se cumpla con los pilares del proyecto. Estos se basan en la apertura al público del software y hardware, el enfoque educativo, así como el desarrollo correcto y seguro en actividades académicas y la capacidad de conectar al robot a una aplicación móvil para su facilidad a la hora de ser incorporado en planes académicos en los colegios costarricenses.

3.3 Establecimiento de las especificaciones

Las especificaciones, según la metodología Ulrich y Eppinger, corresponden al método usado para transformar las necesidades del cliente en expresiones o datos que pueden ser medidos para establecer a qué nivel son cumplidas. Estas deben de continuar evitando cualquier tipo de sesgo que se relacione a la forma en la que se solucionará el problema. Además, las especificaciones poseen dos partes, la primera corresponde a la métrica que representa la sección cuantificable de la parte de un producto y la segunda parte, la cual es llamada valor objetivo, que toma la forma de los valores que debe alcanzar la métrica para que la necesidad sea considerada como cumplida. [20]

En primer lugar, se estableció la lista de métricas siguiendo la guía en la que se genera al menos una métrica por cada una de las necesidades establecidas anteriormente. Además, cada una de las métricas debe estar ligada de forma directa a una necesidad. Por esta razón se siguió el método de la matriz, el cual corresponde a una matriz que muestra la relación directa entre necesidades y especificaciones [20].

El resultado de esta matriz se encuentra en la Figura 19, además la lista completa de las métricas de las especificaciones se muestra a continuación.

1. Aplicaciones educativas en las que puede ser usado el robot.
2. Características del Atta-Bot actual mantenidas en el rediseño.
3. Precio del robot.

4. Precisión de los giros del robot.
5. Tiempo de autonomía.
6. Número de componentes no implementados en la aplicación móvil educativa actual.
7. Vinculación con la aplicación móvil.
8. Precisión del sensor de color.
9. Capacidad de manipulación de la garra.
10. Precisión de la detección de obstáculos.
11. Precisión de la distancias recorrida.
12. Elemento utilizado para la señalización del proceso de acciones del robot.
13. Señalización usada en la carcasa.
14. Tiempo para realizar la conexión de la batería al ser cargada.
15. Botones implementados.
16. Estructura modular del diseño.
17. Robustes eléctrica.
18. Elemento de construcción implementado.
19. Dimensiones del robot.
20. Nivel de accesibilidad del público al software y hardware.

	Aplicaciones educativas en las que puede ser usado	Características del Atta-Bot actualmente mantenidas en el rediseño	Precio del robot	Posición de los grados del robot	Tiempo de autonomía	Número de componentes no implementados en la aplicación móvil educativa actual	Vinculación con la aplicación móvil	Precisión del sensor de color trío	Capacidad de manipulación de la garras	Capacidad de manipulación de ligadas	Posición de la retroacción de obstáculos	Presión en distancia recordada	Complemento de la regulación en escuelas y colegios	Elemento utilizado para la señalización del proceso de acciones del robot	Selcción usada en la selección	Tiempo para realizar la conexión de la batería al ser cargada	Botoneras implementadas	Efectúa una modalidad de diseño	Nivel de resistencia a golpes/bolas de acero	Blanqueo de construcción implementado	Dimensiones del robot	Nivel de accesibilidad al público al software y hardware	
Se realiza un rediseño del robot actual llamado Atta-Bot.	x																						
El rediseño del robot simplifica el robot Atta-Bot actual.		x			x																		
El rediseño del robot está orientado a la generación de un robot educativo.		x																					
El precio del robot rediseñado es menor o igual al actual.			x																				
Se mantiene en gran parte el sistema de movimiento del Atta-bot actual.	x	x							x														
Se mantiene la calidad y precisión de los componentes actuales del Atta-Bot.		x			x			x	x														
Se mantiene el tiempo de autonomía actual de Atta-Bot (tiempo que dura la batería)			x			x														x			
Se eliminan los componentes que no son relevantes para las aplicaciones educativas del robot.							x																
El nuevo robot es capaz de vincularse a una aplicación móvil para ser controlado.					x				x														
En el nuevo diseño se implementa un sensor color.						x				x													
En el nuevo diseño se implementa una garra.							x			x													
Se mantiene un sistema de detección de obstáculos.								x		x	x												
El nuevo robot realiza todos sus acciones de forma precisa para las aplicaciones educativas.			x		x			x	x														
El nuevo robot es seguro, especialmente para su uso en escuelas y colegios.																							
El robot tiene una forma de mostrar que la acción está siendo realizada o finalizó (LEDs o pantalla)												x											
En la carcasa se tiene una forma de señalizar el frente (una cara, una luz, etc)												x											
La batería es fácil de cargar y se carga de forma intuitiva.												x											
Se tienen botones para el inicio del programa y para cancelar el programa.												x								x			
El nuevo diseño es modular (no está soldado en un solo lugar y los componentes pueden ser desconectados)												x								x			
El nuevo robot es robusto												x								x			
Se implementa un elemento de construcción a la hora de desarrollarse los procesos educativos.												x								x			
El tamaño del nuevo robot es menor al diseño actual												x								x			
El diseño del hardware y del software es de carácter abierto y accesible al público.												x								x			

Figura 19. Matriz Necesidades-Métricas.

Como se evidencia en la matriz creada, cada una de las necesidades del cliente posee al menos una especificación asociada. A demás de esto, múltiples de la necesidades se pueden asociar a más de una de las métricas para ser validadas. La creación de esta matriz permite verificar la correlación necesaria para que la metodología sea aplicada de forma correcta, por lo que su documentación es una parte vital del proceso de diseño. [20]

Una vez que se establecen las métricas, es necesario determinar los valores objetivos. Estos corresponden a los valores marginales y los valores ideales. El proceso para la determinación de estos valores está basado en una combinación entre un estudio del mercado actual y una comprensión de los deseos del cliente expresados a través de las necesidades.[20]

Durante el estudio del arte del presente proyecto se realizó una tabla comparativa de algunos de los robots educativos más importantes en el mercado actual, su resultados se presentan en la Tabla 1. Para la determinación de los valores se realiza una tabla comparativa de estos mismos robots educativos, enfocada en las métricas determinadas anteriormente. Los

resultados se encuentran en la por lo que se añadió tiempo para aquellos robots que deben manipularse para ser conectados, como cambiar su dirección para acceso del puerto.

Tabla 6 para los robots cuya naturaleza es más cercana al Atta-Bot original. En la

Tabla 7 se encuentra una tabla similar con los robots educativos que poseen menos de las métricas buscadas y además se consideran muy lejanos al Atta-Bot original.

Debido a que los robots existentes que están siendo evaluados no pudieron ser analizados en persona y no tienen ningún tipo de lazos al diseño original del Atta-Bot, algunas de las métricas no pueden ser definidas en esta comparación, por lo que fueron eliminadas de la tabla. Entre éstas se encuentran las métricas 2 y 6 las cuales están directamente relacionadas a los componentes internos del diseño actual del Atta-Bot, por lo que no existe una buena forma de definir los valores para los modelos de comparación.

Las métricas 4, 11, 12, 16 y 19 no son definidas en todos los casos por los fabricantes de los robots y tampoco se encontró información proveniente de los usuarios de los robots, por lo que en casos donde no se tiene suficiente información se crearon datos utilizando el proceso de estimación de Fermi, esto se señala en la tabla con un asterisco acompañando el dato conseguido. En caso de que una de las métricas no pueda ser aplicada a uno de los robots y sí al resto, se llenó el espacio correspondiente con un NA.

Para la precisión de movimiento y giros, los milímetros y grados mostrados en la tabla corresponden al posible fallo en movimientos que puede poseer el robot a la hora de realizar tareas.

Algunos de los robots son programados sin darse una distancia, sino que se dan segundos o solo se le indica que avance, por lo que el error aproximado es mayor; además, se toma en cuenta la naturaleza de la movilidad del robot y los motores usados para establecer este valor.

[12]

En términos de los sensores que perciben los obstáculos, se encontró la información para algunos robots, pero también se tuvo que estimar valores tomando en cuenta las características generales de sensores ultrasónicos.[22]

El tiempo de inicio para la carga del robot se calculó tomando en cuenta que el tiempo para realizar una conexión a puestos USB, USB mini y USB-C y es de aproximadamente 30 segundos a 1 minuto [23], por lo que se añadió tiempo para aquellos robots que deben manipularse para ser conectados, como cambiar su dirección para acceso del puerto.

Tabla 6. Comparación de las Métricas de los Robots Educativos más Cercanos al Atta-Bot.

# Métrica	# Necesidad	Métrica	Imp	Unidad	Robot Zowi	Robot Mindstorms de Lego	Robot Thymio
1	3	Cantidad de componentes enfocados a aplicaciones educativas.	5	NA	6 [14]	9 [15]	9 [17]
3	4	Precio del robot.	3	\$	108 [14]	370 [15]	189 [17]
4	5, 6, 14	Precisión de los giros del robot.	3, 4 y 5	grados	NA	10*	10*
5	7	Tiempo de autonomía.	3.5	minutos	De 300 a 480 [24]	De 300 a 600. [15]	600. [25]
7	9, 14	Vinculación con la aplicación móvil.	5	Lista	Bluetooth [14]	Bluetooth [15]	Dongle. [17]
8	6, 10	Capacidad del sensor de color.	4 y 3.5	Lista	NA	Sensor de color [15]	Seguidor de línea. [17]

9	11	Capacidad de manipulación de la garra.	3.5	Lista	NA	Puede ser construida con varios niveles de manipulación. [15]	NA
10	6, 13, 14	Distancia de detección de obstáculos.	4 y 5	cm	De 2 a 200*	De 70 a 200. [26]	De 5 a 10. [27]
11	5, 6, 14	Precisión de la distancia recorrida.	3, 4 y 5	mm	10 a 20*	5 a 10*	Depende del código.
12	17	Elemento utilizado para la señalización del proceso de acciones del robot.	4.5	Lista	LEDs configurables [14]	NA	Anillos de LEDs y LEDs en cada elemento. [17]
13	18	Señalización usada en la carcasa.	4	Lista	Cara del robot. [14]	Puede ser construida de forma personalizada. [15]	Curva indica el frente. [17]
14	19	Tiempo para realizar la conexión de la batería al ser cargada.	4.5	s	60 a 120*	30 a 60*	30 a 60*

15	20	Botones implementados.	4.5	Lista	2 botones de función múltiple. [14]	Botón de inicio y botones usados dependiendo de la programación. [15]	5 botones programables. [17]
16	21	Estructura modular del diseño.	4.5	Lista	Partes eléctricas y carcasa modular. [14]	Completa-mente modular. [15]	NA
17	3, 15, 22	Robustes eléctrica	5 y 4.5	Lista	Media	Alta	Alta
18	23	Elemento de construcción implementado.	3.5	Lista	Cambio de partes eléctricas y carcasa. [14]	Diseño basado en la construcción del robot. [15]	NA
19	24	Dimensiones del robot.	4	cm	14 x 9.3 x 15	Versátil dependiendo de la construcción [15]	5 x 11 x 11. [17]
20	25	Nivel de accesibilidad del público al software y hardware.	5	Lista	No hay acceso para el público fuera de su	No hay acceso para el público fuera de su	No hay acceso para el público fuera de

					programación. [14]	programación. [15]	su programación. [17]
--	--	--	--	--	-----------------------	--------------------	--------------------------

Tabla 7. Comparación de las Métricas de los Robots Educativos más Diferentes al Atta-Bot.

# Métrica	# Necesidad	Métrica	Imp	Unidad	Robot Sphero Mini	Robot Bee-Bot
1	3	Cantidad de componentes enfocados a aplicaciones educativas.	5	NA	6 [12]	3 [16]
3	4	Precio del robot.	3	\$	50 [12]	98 [28]
4	5, 6, 14	Precisión de los giros del robot.	3, 4 y 5	grados	NA	NA
5	7	Tiempo de autonomía.	3.5	minutos	De 180 a 300. [12]	De 120 a 480 [28]
7	9, 14	Vinculación con la aplicación móvil.	5	Lista	Bluetooth. [12]	NA
8	6, 10	Capacidad del sensor de color.	4 y 3.5	Lista	NA	NA
9	11	Capacidad de manipulación de la garra.	3.5	Lista	NA	NA

10	6, 13, 14	Distancia de detección de obstáculos.	4 y 5	cm	NA	NA
11	5, 6, 14	Precisión de la distancia recorrida.	3, 4 y 5	mm	10 a 50*	10 a 15*
12	17	Elemento utilizado para la señalización del proceso de acciones del robot.	4.5	Lista	NA	Sonido y luces LED en los ojos. [28]
13	18	Señalización usada en la carcasa.	4	Lista	Logo del robot. [12]	Cara de la abeja. [16]
14	19	Tiempo para realizar la conexión de la batería al ser cargada.	4.5	s	30 a 60*	60 a 120*
15	20	Botones implementados.	4.5	Lista	Botón de encendido. [12]	Botón de inicio, 4 de flecha, botón de cancelación y botón de limpieza. [16]
16	21	Estructura modular del diseño.	4.5	Lista	Cambio de carcasa [12]	NA
17	3, 15, 22	Robustes eléctrica.	5 y 4.5	Lista	Alta	Alta
18	23	Elemento de construcción implementado.	3.5	Lista	Cambio de carcasa [12]	NA
19	24	Dimensiones del robot.	4	cm	4 x 4 x 4. [12]	10 x10 x 10. [16]

20	25	Nivel de accesibilidad del público al software y hardware.	5	Lista	No hay acceso público fuera de programación. [12]	No hay acceso para el público. [16]
----	----	--	---	-------	--	--

Una vez se realiza esta tabla, se debe de tomar en cuenta el diseño inicial del Atta-Bot, debido a que las características de este robot son esenciales para definir los valores objetivo, especialmente cuando las necesidades buscan una mejora o que este valor se mantenga en el nuevo diseño. Por esta razón se creó la pero se concentra únicamente en los valores del Atta-Bot inicial. Los valores que no han sido establecidos en la documentación fueron estimados o probados en los Atta-Bots a disposición. Las estimaciones también son realizadas con la misma información disponible mencionada anteriormente y son indicadas nuevamente con un asterisco.

Tabla 8, la cual es similar a la por lo que se añadió tiempo para aquellos robots que deben manipularse para ser conectados, como cambiar su dirección para acceso del puerto.

Tabla 6 pero se concentra únicamente en los valores del Atta-Bot inicial. Los valores que no han sido establecidos en la documentación fueron estimados o probados en los Atta-Bots a disposición. Las estimaciones también son realizadas con la misma información disponible mencionada anteriormente y son indicadas nuevamente con un asterisco.

Tabla 8. Métricas del Atta-Bot Actual.

Número de Métrica	# Necesidad	Métrica	Imp	Unidad	Atta-Bot actual
1	3	Cantidad de componentes enfocados a aplicaciones educativas.	5	NA	6. [21]
3	4	Precio del robot.	3	\$	≈ 216.18

4	5, 6, 14	Precisión de los giros del robot.	3, 4 y 5	grados	10*
5	7	Tiempo de autonomía.	3.5	minutos	>45 [21]
7	9, 14	Vinculación con la aplicación móvil.	5	Lista	Radio Frecuencia. [21]
8	6, 10	Capacidad del sensor de color.	4 y 3.5	Lista	NA
9	11	Capacidad de manipulación de la garra.	3.5	Lista	NA
10	6, 13, 14	Distancia de detección de obstáculos.	4 y 5	cm	De 3 a 4. [21]
11	5, 6, 14	Precisión de la distancia recorrida.	3, 4 y 5	mm	10-20. [21]
12	17	Elemento utilizado para la señalización del proceso de acciones del robot.	4.5	Lista	NA
13	18	Señalización usada en la carcasa.	4	Lista	NA
14	19	Tiempo para realizar la conexión de la batería al ser cargada.	4.5	s	30 a 60*
15	20	Botones implementados.	4.5	Lista	NA
16	21	Estructura modular del diseño.	4.5	Lista	No es modular fuera de la carcasa. [21]
17	3, 15, 22	Robustes eléctrica.	5 y 4.5	Lista	Media

18	23	Elemento de construcción implementado.	3.5	Lista	Construcción de la carcasa y soldadura de los componentes. [21]
19	24	Dimensiones del robot.	4	cm	15.5 x 17.8 x 10.5. [21]
20	25	Nivel de accesibilidad del público al software y hardware.	5	Lista	Completamente abierto al público. [21]

Con esta información se puede realizar el proceso de establecimiento de los valores objetivo. Estos corresponden al primer acercamiento al grado de cumplimiento deseado. Por esta razón, estos deben de ser revisados para realimentación después del desarrollo de fases posteriores para tomar en cuenta factores imprevistos que se descubrirán en etapas más avanzadas del diseño. Estos valores se dividen en dos categorías diferentes. En primer lugar, se encuentran los valores meta que pueden ser definidos como aquellos que deberían de ser alcanzados, en cambio los valores marginales son caracterizados por imponer una menor restricción y pueden asumirse sin realimentación. Son los mínimos valores que se deben alcanzar para cumplir con una necesidad. Ningún valor dentro de estas categorías debe ser expresado como exacto, sino que se usan rangos, para que se facilite su aplicación tomando en cuenta consideraciones y adaptaciones del diseño. En la Tabla 9 se presentan los valores marginales y meta en conjunto con su métrica y otra información relevante. En este caso si se incluye cada una de las métricas, debido a que es necesario acompañar cada una con los valores para una correcta definición de especificaciones. [20]

Cabe destacar que, aparte de los datos recaudados por medio de la investigación de los robots, se usó la información recaudada mediante conversaciones con el cliente. De estas conversaciones se desarrolló una compresión más profunda de lo que se desea del nuevo robot, la aplicación móvil existente y las posibles actividades educativas en las que será implementado el robot. Durante estas conversaciones se evadieron los sesgos al mantener la información recolectada centrada en las funciones deseadas del robot y no en cómo se implementarán estas funciones.

Tabla 9. Definición de Valores Meta y Valores Marginales para cada una de las métricas del Rediseño del Robot.

# de Métrica	# de Necesidad	Métrica	Imp	Unidades	Valor Marginal	Valor Ideal
1	3	Cantidad de componentes enfocados a aplicaciones educativas.	5	NA	8-10	10-12
2	6	Características del Atta-Bot actual mantenidas en el rediseño.	4	Lista	Precio, movilidad y accesibilidad.	Precio, movilidad, precisión, accesibilidad,
3	4	Precio del robot.	3	\$	< 250	< 216.18
4	5, 6, 14	Precisión de los giros del robot.	3, 4 y 5	grados	20-25	10-15
5	7	Tiempo de autonomía.	3.5	minutos	> 45	> 90
6	4, 8, 11 y 12	Número de componentes no implementados en la aplicación móvil educativa actual.	3, 3.5, 4, 4.5	NA	8-11	7-8
7	9, 14	Vinculación con la aplicación móvil.	4 y 3.5	Lista	Vinculación mediante cable de datos.	Vinculación mediante Bluetooth.

8	6, 10	Capacidad del sensor de color.	3.5	Lista	Seguidor de línea.	Sensor de luz y color completo.
9	11	Capacidad de manipulación de la garra.	4	Lista	La garra abre y cierra.	La garra es capaz de manipular objetos.
10	6, 13, 14	Distancia de detección de obstáculos.	3, 4 y 5	cm	2-3	4 o más
11	5, 6, 14	Precisión de la distancia recorrida.	4.5	mm	20-30	10-20
12	17	Elemento utilizado para la señalización del proceso de acciones del robot.	4	Lista	Luz LED RGB.	Pantalla pequeña.
13	18	Señalización usada en la carcasa.	4.5	Lista	Garra en el frente.	Pantalla o LED en el frente.
14	19	Tiempo para realizar la conexión de la batería al ser cargada.	4.5	s	60-120	30-60

15	20	Botones implementados.	4.5	Lista	Inicio y paro de emergencia.	Inicio, paro de emergencia y 2 botones programables.
16	21	Estructura modular del diseño.	5 y 4.5	Lista	La mayoría de los componentes no usan soldadura y son modulares.	Ninguno de los componentes usa soldadura y son modulares.
17	3, 15, 22	Robustes eléctrica.	3.5	Lista	Media	Alta
18	23	Elemento de construcción implementado.	4	Lista	Se construye la carcasa.	Se construye la carcasa y se conectan los componentes.
19	24	Dimensiones del robot.	5	cm	< 17 x 19 x 12	< 15.5 x 17.8 x 10.5
20	25	Nivel de accesibilidad del público al software y hardware.	5	Lista	Completamente abierto al público.	Completamente abierto al público.

En conclusión, se determina que las métricas desarrolladas corresponden a verdaderas métricas que pueden ser medidas en el proceso de validación, además que su redacción y directrices son las correctas según el proceso de la metodología. Las métricas, además, cumplen su función de corresponder con las necesidades, eliminan sesgos de todo tipo y los valores determinados a partir de la información de otros robots y de los deseos del cliente son defendibles. Por último, la revisión de métricas, después de la selección de los conceptos, dio como resultado una pequeña cantidad de cambios debido a que no ocurre ninguna

eliminación de métricas, solo cambios en su redacción para hacerlas más fáciles de medir y validar.

3.4 Conceptualización del Proceso de Diseño

Según la metodología seguida existen múltiples formas para el desarrollo de la conceptualización del problema, y en este caso se buscó como primer paso hacer un estudio del problema, iniciando con dividirlo en subproblemas, lo cual ayuda a una comprensión más profunda del mismo. Además, permite definir cuáles de estos subproblemas pueden ser considerados como críticos.

Hecho lo anterior, se realizó una búsqueda de conceptos: primero externa que consiste en la búsqueda de información y soluciones de los usuarios líderes, patentes, expertos, literatura, entre otras fuentes, y una búsqueda interna la cual pone en juego los conocimientos, las habilidades y la experiencia del diseñador.

En este caso la búsqueda interna fue en su mayor parte individual por parte del estudiante, pero también fue asistida por medio del asesor industrial para fomentar la emisión de ideas libres. Después de esto se empleó una exploración sistemática que usó árboles de clasificación de conceptos y tablas de combinación de conceptos para asegurar que se fue adecuadamente exhaustivo en la generación de éstos, así como generar posibles soluciones candidatas que combinen ideas para todos los subproblemas y sean compatibles entre sí.

Por último, y como en cada sección de la metodología, se realizó una reflexión del proceso que fue desarrollado, así como los resultados que fueron obtenidos, para verificar que todos los requerimientos de esta etapa fueron alcanzados de la mejor forma. [20]

Entonces, en primer lugar, la división fue aplicada al problema definido anteriormente y consiste en que el Atta-Bot actual no posee un diseño ideal para un robot educacional.

Debido a que estamos frente a un problema mecatrónico, existen múltiples subsistemas que deben de ser solucionados para abordarlo, por lo que se decidió realizar una descomposición funcional en la que cada uno de los subproblemas son identificados como funciones.

Se partió entonces de una caja negra que, como se observa en la Figura 20, se definieron las entradas y las salidas. Después de esto se realizó un proceso interactivo hasta que se pudo percibir como cada uno de los problemas puede ser solucionado de forma individual [20].

El resultado final se encuentra en la Figura 21.

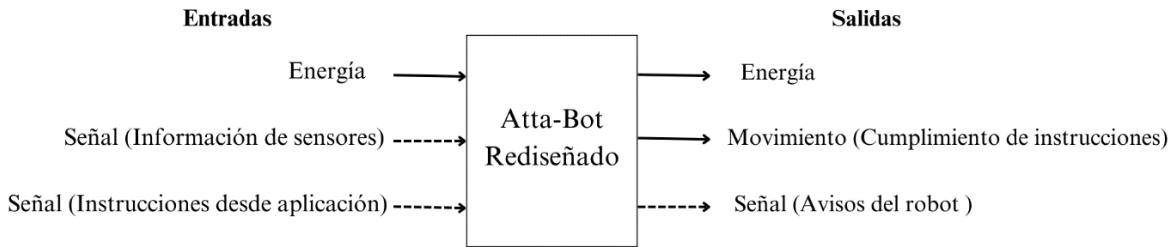


Figura 20. Descomposición Funcional en la Etapa de Caja Negra.

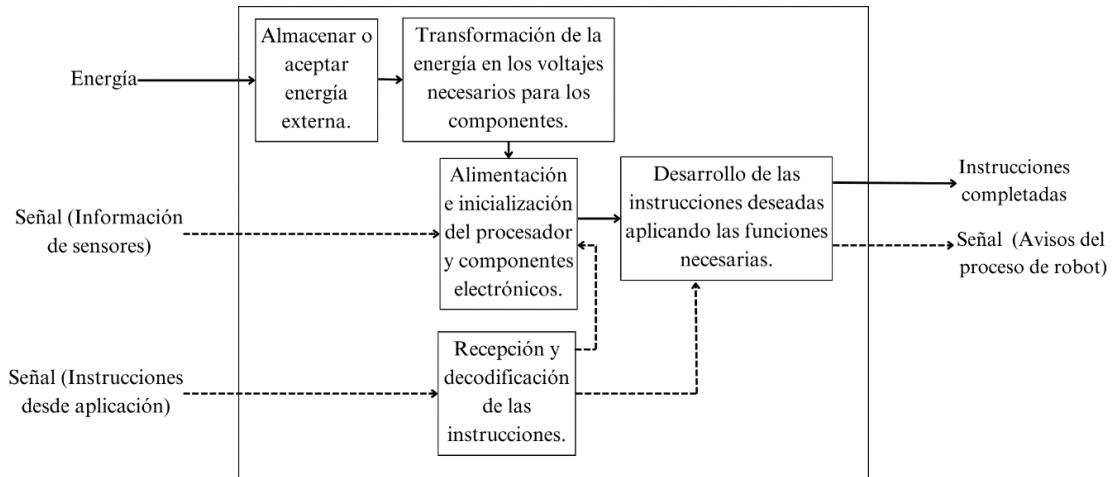


Figura 21. Descomposición Funcional Final.

Para la primera iteración de la descomposición funcional, en donde se buscan identificar las entradas y salidas, se decidió representar a las físicas con una línea sólida y a las digitales con una línea punteada como se reconoce en la Figura 20. Esto se hace con el objetivo de reconocer de forma sencilla su naturaleza y la forma en las que serán manipuladas dentro del robot; además, en el primer esquema se definió como única entrada física la energía del sistema, la cual es necesaria para el desarrollo de las acciones del robot.

Por otro lado, se tienen dos entradas digitales, la primera corresponde a las señales creadas por los sensores que permiten la evaluación del ambiente y son necesarias para el cumplimiento de las acciones del robot, mientras que la segunda corresponde a las

instrucciones que debe de recibir el robot desde la aplicación móvil las cuales indican qué acciones específicas realizará el Atta-Bot.

Como caja negra, se identificó el rediseño del Atta-Bot, ya que este nuevo dispositivo es el responsable de tomar estas entradas y generar las salidas deseadas. Se creó una salida física la cual corresponde a la energía, debido a que si un sistema recibe energía debe de tener una forma de generarla o transformarla; además, se habilitó una segunda salida física que indica el movimiento del robot para cumplir con las instrucciones dadas, aunque cabe destacar que no todas las instrucciones indican movimiento sino una activación de sus sensores y actuadores. Por último, tiene una salida digital que busca representar la señal que provee el Atta-Bot acerca del proceso de cumplimiento de las instrucciones y esta señal busca tener una representación física, aunque también es se encarga de informar al robot el estado en el que se encuentra.

Para la descomposición funcional final, se identifican las mismas entradas iniciales. La energía en primer lugar debe ser almacenada o aceptada de una fuente externa, después de esto y para que esta pueda ser utilizada dentro del robot, se debe de transformar en los voltajes necesarios que son indicados como VCC de los componentes internos del robot. Una vez que se consiguen estos voltajes se alimentan los componentes, entre estos el controlador o cerebro del robot, el cual se inicializa y recibe la información de los sensores y la información dada por la aplicación móvil, la cual debe de pasar por un proceso de recepción y decodificación.

Una vez que se cuente con toda esta información, el procesador utiliza las funciones de su código interno para realizar cada una de las instrucciones deseadas, ya sea movimientos, ciclos o el uso de sensores para interpretar información del ambiente. Logrado esto hay dos salidas: las instrucciones completas y la señal de cumplimiento y desarrollo de estas. Por ende, se tienen tres subproblemas identificados como sistemas, el subsistema de energía, el subsistema de sensores y actuadores y el subsistema que aplica el protocolo de comunicación con la aplicación móvil.

Con la descomposición profunda del problema se realizó el siguiente paso que corresponde a una búsqueda de conceptos.

El primer tipo de búsqueda que se aplicó fue una externa y el punto de vista inicial explorado corresponde a los usuarios líderes, volviéndose a visitar los productos explorados en el estado del arte, ya que estos corresponden a los robots móviles educativos más populares del mercado actual.

Esta vez la observación se hizo desde la perspectiva de los tres subproblemas identificados para el rediseño del Atta-Bot. Además, se exploró como estos problemas son actualmente abarcados en el modelo de Atta-Bot original.

Esta búsqueda consideró que fue externa debido a que abarcó inicialmente un estudio de la documentación actual del robot Atta-Bot y no de una lluvia de ideas en conjunto con los creadores de la versión original. Durante esta búsqueda se encontraron diferentes formas de resolver los subproblemas identificados, por lo que los resultados obtenidos se muestran en la Tabla 10.

Tabla 10. Resultados de la búsqueda externa desde el punto de vista de usuarios líderes.

Robot Existente	Subproblemas Identificados		
	Energía	Sensores y actuadores	Protocolo de comunicación
Atta-Bot Actual	Batería de litio de 3.7 V y 2000mAh conectada a un Boost de 5V. [21]	Los sensores y actuadores seleccionados se encuentran conectados a dos microcontroladores por medio de una placa de circuito impreso. [21]	Se utiliza una comunicación por medio de radio frecuencia, que es posible con la implementación del controlador Feather M0 RFM69 que cuenta con el módulo necesario. [21]

Robot Zowi	Se alimenta con una batería de litio de 4040mAh. [24]	Posee una placa controladora en la que se conectan todos sus sensores y actuadores a una placa de Bq y compatible con Arduino. [14]	Se conecta a la aplicación móvil por medio de Bluetooth. [14]
Robot Mindstorms de Lego	Existe un ladrillo de batería de 2050mAh o también puede ser alimentado con batería AA. [15]	Tiene como corazón un ladrillo inteligente conocido como EV3 y se conectan los componentes de forma modular. [15]	Se conecta a la aplicación móvil por medio de Bluetooth, Wi-Fi o con conexión USB. [15]
Robot Thymio	Batería interna que puede ser cargada desde un puerto en su exterior. Se desconoce su naturaleza. [17]	Sensores y actuadores conectados a una tarjeta madre. Se desconoce su naturaleza. [17]	Se conecta a la aplicación móvil por medio de Bluetooth o Dongle. [17]
Robot Sphero Mini	Batería interna que puede ser cargada desde un puerto en su exterior. Se desconoce su naturaleza. [12]	Sensores y actuadores conectados a una tarjeta madre. Se desconoce su naturaleza. [12]	Se conecta a la aplicación móvil por medio de Bluetooth. [12]
Robot Bee-Bot	Batería interna que puede ser cargada	Sensores y actuadores	No posee un protocolo de

	desde un puerto en su exterior. Se desconoce su naturaleza. [28]	conectados a una tarjeta madre. Se desconoce su naturaleza. [28]	comunicación debido a que no debe establecer conexiones con ninguna aplicación. [28]
--	--	--	--

El próximo paso dentro de esta búsqueda externa corresponde a la investigación de patentes existentes, con una concentración en los tres subproblemas. Además, se buscaron posibles patentes que correspondan a robots educativos para descubrir como estos abordan los mismos subproblemas.

La primer patente encontrada de número CN210189807, tiene como título “Robot de Educación Multifuncional” o “Multifunctional education robot” en inglés. El patrón fue generado en China y describe un robot educacional que se enfoca en poseer una variedad de sensores para que los estudiantes puedan aprender diferentes programas con el uso individual de éstos, así como sus usos en combinación.

Este modelo en particular tiene además dos brazos que son controlados por motores y un tercer motor que controla el movimiento del robot como tal. Por otro lado, implementa un sistema de carga de USB el cual es fácil de acceder e intuitivo, lo que es vital para su uso por parte de estudiantes. Por último, según el diagrama que presenta la patente, parece tener una pantalla en el frente [29]. Este diagrama es presentado en la Figura 22 y ayuda a generar un mejor entendimiento del diseño presentado en la patente.

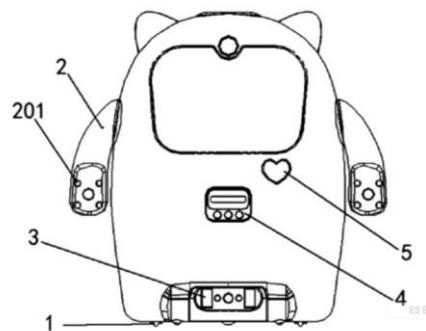


Figura 22. Diagrama de la patente CN210189807. [29]

La segunda patente encontrada tiene el número CN209986949 y también viene de China. En este caso se titula “Robot Educacional” o “Education robot” en inglés y este diseño en particular se destaca por su estructura que le permite el movimiento. Dicha estructura consiste en dos placas conectadas por una serie de bolsas de aire y resortes, como se muestra en la Figura 23, lo que permite que el robot se mueva, pero además, que se disipen las vibraciones generadas por el movimiento para que estas no afecten los componentes electrónicos del robot. Además, se nota que el robot presenta igualmente botones y una pantalla para interactuar con todas sus funciones. [30]

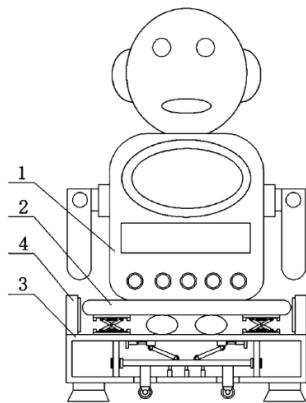


Figura 23. Diagrama de la patente CN209986949. [30]

La tercera patente que se encontró presenta nueva información y posibles soluciones de los subproblemas del robot. Tiene de número CN210361316 y esta patente se titula “Robot para la educación infantil” o “Robot for children education”, también creada en China.

Este robot se caracteriza por su naturaleza modular y su hardware puede ser expandido al añadir nuevas piezas a su cuerpo. La dirección en la que los sensores recolectan datos del ambiente puede variar al mover la orientación de la cabeza. Está diseñado para poder moverse en un plano y su mayor punto de innovación es la capacidad educativa que posee cuando se analiza la relación de los estudiantes a su personalización [31].

En la Figura 24 se encuentra el dibujo proporcionado en la patente que ayuda a comprender mejor su diseño.

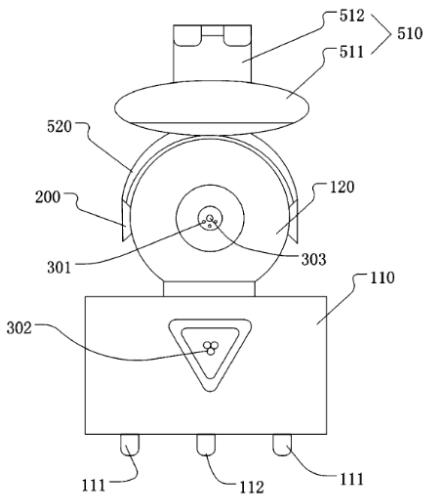


Figura 24. Diagrama de la patente CN210361316. [31]

La última patente que se analizará se titulada “Aparato de robot educacional para niños y el método operativo del mismo” y tiene el número US20110129804. El diseño presentado corresponde a un robot que es realizado en un estilo de creación del usuario. Además, el robot está formado por tres partes: un decodificador, un controlador y un cedente.

El decodificador escanea una carta de entrada en la que se tiene un código o una serie de instrucciones presentadas en la forma de código de barras. Después, el controlador codifica al programa que controlará al robot con lo deseado según la tarjeta de entrada. Por último, el cedente transfiere el código al robot para su ejecución [32].

En la Figura 25 se encuentra el diagrama de la patente que muestra claramente las tres partes, así como el diseño básico del robot móvil educativo.

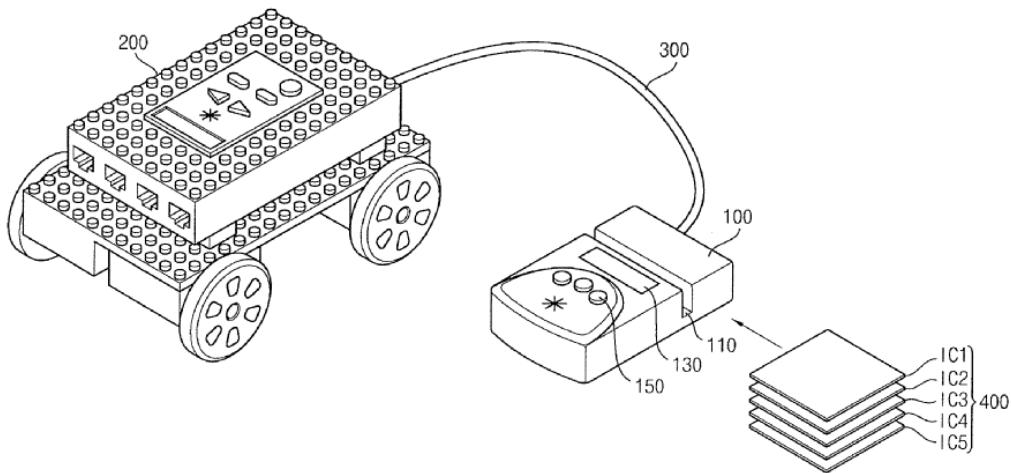


Figura 25. Diagrama de la patente US20110129804. [32]

El siguiente paso que se llevó a cabo durante la búsqueda externa correspondió a un análisis de los diferentes componentes que se encuentran disponibles para el desarrollo del robot, tomando en cuenta los factores que fueron favorecidos en el Atta-Bot original, los cuales corresponden al bajo precio y a la disponibilidad de los componentes mediante los distribuidores del país. Por esta razón se decidió generar una serie de listas en diferentes subcategorías según el tipo de componente, en las que se describen las características de cada una de las opciones disponibles, con el objetivo de generar un conocimiento profundo de los componentes que se encuentran disponibles de forma realista y en el país para el desarrollo del diseño del robot. Estas listas están en la Tabla 11.

Tabla 11. Listas de los componentes disponibles para el diseño del robot. [33]

Categoría	Componentes		
	Código	Descripción	Precio (\$)
Fuente de Energía	ROB-MC008	Sostenedor de baterías AA. Puede sostener 3 baterías AA que equivale a 3.6 Voltios.	2.50
	BAT-CB1366	Batería de polímero de iones de litio. Es fuente de 3.7 V y 600mAh. Es recargable.	8.95

	BAT-CB1539	Batería de polímero de iones de litio. Es fuente de 3.7 V y 400mAh. Es recargable.	7.95
	BAT-CB1529	Batería de polímero de iones de litio. Es fuente de 3.7 V y 100mAh. Es recargable.	5.95
	ADA-1903	“Boost” de voltaje básico que convierte el voltaje a 5V desde 1.8V a 500mA. Además, ayuda a proteger el circuito de picos eléctricos.	10.95
	ADA-2465	“Boost” de voltaje básico que convierte el voltaje a 5V desde cualquier batería de litio de 3.7V. Además, ayuda a proteger el circuito de picos eléctricos y tiene un circuito que permite cargar la batería con una entrada USB-mini.	32.95
Subsistema de movimiento	MG996R-360	Servomotor de rotación continua diseñado con proyectos de robótica en mente. Solo se puede controlar la dirección y velocidad no la posición.	9.95
	MG996R	Servomotor con un rango de 180 grados de rotación. Puede controlar posición, dirección y velocidad.	8.95
	ROB-CBDG01D-M	Set de 2 motores de corriente directa con una relación 48:1 y conectores macho. Voltaje de entrada de 4.5V y torque de 800 gf.m. Existe una versión con conectores hembra.	7.49
	ROB-0309	Motor de engranaje helicoidal de alto torque. Voltaje de entrada de 12 V y torque de 5.6 kg*cm. Hecho por completo de metal, mide 7 cm en su dimensión más grande.	15.95

	ROB-CB1019	Motor DC con encoder adjunto a su lado trasero. Voltaje de entrada de 6 V y un torque de 0.12 kg*cm.	14.95
	ROB-CB0N20	Motor de corriente directa de tamaño N20 con encoders de cuadratura y ruedas. Voltaje de entrada de 3 a 5 V, pequeño tamaño.	14.95
	COM-CB0234	Driver para dos motores, puede dar voltaje de 2.5V a 12V. Driver simple y de bajo costo.	2.49
	ROB-09457	Driver genérico de hasta 2 motores, puede dar un voltaje de salida de 15V máximo.	6.95
	ROB-CB1818	Bola giratoria usada como el tercer punto de apoyo en un robot diferencial. Tamaño de la bola es de 12 mm.	2.75
Detección de obstáculos	ADA-3595	Sensor de proximidad, luz, RGB y gestos. Se conecta mediante un bus de I2C, pero es compatible con STEMMA QT para su fácil conexión.	11.95
	SEN-CB0006	Módulo de detección de obstáculos. Utiliza un emisor y un receptor de luz infrarroja. Posee potenciómetros para su mejor calibración. Distancia: 2 cm a 40 cm.	2.95
	SEN-00241	Set de emisor y receptor de luz infrarroja. No poseen módulos de calibración. Distancia no establecida, depende de su calibración.	4.35
	TCRT5000	Sensor infrarrojo que posee un fototransistor y un emisor de luz. No posee módulo de calibración. Distancia no establecida, depende de su calibración.	1.49

	SEN-CB0005	Módulo de sensor de pista. Utiliza la luz infrarroja para determinar si hay una superficie sólida como pista. Distancia: 2 cm a 40 cm.	1.95
	Sharp GP2Y0A21YK	Sensor infrarrojo de proximidad con una salida análoga. Esta salida no solo determina la detección de un obstáculo, sino la distancia entre el sensor y dicho obstáculo. Distancia: 10 cm a 80 cm	16.95
	HC-SR05	Sensor ultrasónico de distancia. Usa una onda de sonido para medir la distancia entre el sensor y el obstáculo. Distancia: 2 cm a 450 cm.	4.49
	SEN-CB021	Sensor ultrasónico diseñado para ser a prueba de agua. Su comportamiento es igual a otros sensores ultrasónicos. Distancia: de 25 cm a 4.5 m.	19.95
Sensores de posible uso educativo	ADA-4698	Módulo que utiliza un espectrómetro de múltiples canales que permite medir la cantidad de luz presente en las diferentes frecuencias que definen los colores. Se conecta mediante un bus de I2C, pero es compatible con STEMMA QT para su fácil conexión.	25.95
	ADA-3595	Módulo de sensor de proximidad, luz, RGB y gestos. Se conecta mediante un bus de I2C, pero es compatible con STEMMA QT para su fácil conexión.	11.95
	ADA-4566	Módulo de sensores de temperatura y humedad. Se conecta mediante un bus de I2C,	6.95

		pero es compatible con STEMMA QT para su fácil conexión.	
	ADA-5309	Acelerómetro de triple eje. Se conecta mediante un bus de I2C, pero es compatible con STEMMA QT para su fácil conexión.	7.95
	ADA-4366	Magnetómetro de triple eje. Detecta imanes en 3D. Se conecta mediante un bus de I2C, pero es compatible con STEMMA QT para su fácil conexión.	8.95
	ADA-4503	Módulo que combina acelerómetro, giroscopio y IMU en uno solo para 6 grados de libertad.	14.95
	SEN-CB1754	Sensor que escanea códigos de barra o códigos QR. Tiene un tipo de conexión USB. También se puede conectar en un bus de I2C. Diseñado para uso en computadoras.	39.95
Botones	COM-CB1333	Un teclado de membrana con 4 botones que comparten una sola tierra.	1.95
	EB-BUTTON	Módulo que contiene un botón pulsador con la configuración de resistencia de levantamiento o pull up.	2.99
	COM-00097	Botón pulsador miniatura de menos de 1 cm por lado. Creados para uso en protoboards.	0.35
	COM-CB1038	Botones pulsadores de estilo retro de diferentes colores. Diámetro de 30 mm.	1.95
Actuadores de posible uso educativo	ADA-1463	Anillo de neopíxeles, que corresponden a luces LED en tiras manipulables cuyo color y brillo puede ser controlado. Tiene 16 luces.	11.95
	ADA-2226	Configuración de 7 neopíxeles dentro de un círculo de 5 mm de diámetro.	8.95

	COM-CB0214-153A	Matriz 4x4 de neopixeles que utiliza un único pin para su control. Los 16 neopixeles pueden ser configurados de forma individual para crear formas o letras.	7.95
	COM-CB0013	Módulo de una led RGB en la que se puede controlar el color mediante 3 canales PMW.	1.95
	COM-CB0415	LED RGB con cátodo común. Posee una patilla para cada color. No posee un circuito de protección.	0.95
	LCD-CB0079	Pantalla LDC de dos líneas de 16 caracteres. Puede ser controlada por medio de bibliotecas de Arduino. Se conecta por bus de I2C.	5.95
	LCD-CB0081	Pantalla LCD de 4 líneas de 20 caracteres. Puede ser controlada por medio de bibliotecas de Arduino. Se conecta por bus de I2C.	12.95
	LED-CB3945	Módulo redondo de pantalla LCD TFT que permite el uso de color y tiene una diagonal de 1.28 pulgadas. Se conecta por bus de I2C.	16.95
	LCD-CB4557	Pantalla LCD rectangular de pequeño tamaño que permite el uso de color. Tiene una diagonal de 0.96 pulgadas y un área de pixeles de 160x80. Se conecta por bus de I2C.	5.95
Controlador	ROB-CBIBR01	La placa IdeaBoard corresponde a una placa cuyo objetivo es generar una opción ideal para proyectos educacionales del internet de las cosas y de robótica. Posee un chip microcontrolador ESP32-WROOM y permite	39.95

		conexión Bluetooth y Wi-Fi. Tiene dobles puentes H para el control de motores, una led programable, interfaz USB-C, 14 pines de propósito general y 7 pines para componentes que requieren de voltaje directo. Todos sus pines son machos para facilitar la conexión .	
	ADA-5477	La placa Feather ESP32-S3 corresponde a la implementación del microcontrolador ESP32-S3 en una placa con doble núcleo de 240MHz, permite uso de Bluetooth y Wi-Fi; tiene 4MB flash y 2MB PSRAM. Posee 6 pines digitales y 6 pines analógicos para la conexión de componentes. Pines deben ser soldados.	27.95
	DEV-CB02143	La placa que famosamente es conocida como Blue Pill, utiliza un microcontrolador STM32F103 que puede ser configurado con Arduino IDE, pero es considerado un microcontrolador para expertos. Posee 9 pines analógicos y diferentes tipos de pines digitales, deben ser soldados.	16.95
	DEV-CB0204	La placa Developoment Board busca traer la capacidad de WiFi y Bluetooth a un paquete más económico. Esto pone un límite a los pines y las actividades que se pueden realizar con la misma. Tiene alrededor de 7 pines analógicos y 6 digitales.	14.95

	ADA-3027	La placa Adafruit Feather 32u4 FONA está diseñada para ser liviana y fácil de manipular. Tiene una lógica de 3.3V con un núcleo ATmega32u4 de 8MHz. Posee 8 pines de PWM y 10 pines análogos.	49.95
	ARDUINO_UNO	Esta placa de Arduino Uno R3 tiene un núcleo de ATmega16U2. Los pines tienen una entrada hembra y de estos 6 son análogos y 14 son digitales, con 6 de ellos permitiendo señales PWM.	45.95

Con esto se finaliza la búsqueda externa. Es importante destacar que, a través del proceso del desarrollo del diseño del robot, es necesario continuar la búsqueda de información para comprobar que el diseño este siendo realizado de forma correcta. Esto incluye el uso de las hojas de datos de los componentes que fueron seleccionados, así como fuentes que permitan el desarrollo del código interno del Atta-Bot.

El siguiente paso corresponde a una búsqueda interna, la cual se refiere al proceso que utiliza las habilidades, la experiencia y el conocimiento del diseñador. En este caso se realizó una búsqueda interna individual, en la que se contactó a terceros, como el asesor industrial, para fomentar una lluvia de ideas con múltiples perspectivas que ayudara a obtener la solución ideal. Además de la lluvia de ideas, se implementó la metodología de TRIZ en los casos en los que se encontraron contradicciones físicas y técnicas dentro del planteamiento de los candidatos del diseño. [20].

La lluvia de ideas inicial se realizó pensando en el robot como un todo, es decir, esta no se concentró en los subproblemas planteados anteriormente; en su lugar, se realizó en torno al problema general.

En primer lugar, se pensó en eliminar el concepto de doble microcontrolador en el que se basa el Atta-Bot original, ya que el nuevo diseño busca simplificar el robot y en la robótica educativa no se necesita la alta capacidad que es proporcionada por ambos microcontroladores. Esto, además, ayuda a reducir el precio del robot y su tamaño.

La siguiente idea consiste en la simplificación del sistema de detección de obstáculos, ya que el nuevo diseño no tiene un uso de exploración por lo que no es necesario una detección avanzada, el robot actual los identifica, define la posición en la que se encuentran midiendo la distancia entre el robot y el obstáculo y los guarda en su memoria, mientras que el nuevo robot, solo debe detectarlos y correr una rutina para asegurar su seguridad.

Otra de las ideas principales consiste en el uso de bibliotecas avanzadas en la programación del robot. El código de la primera iteración del robot realiza muchas de las acciones de control y lectura de encoder sin uso de bibliotecas, por lo que implementar éstas simplifica el código y en algunos pasos puede realizar estos procesos de una mejor manera. También se discutieron otras ideas en torno al código y su estructura general, y en esta discusión se concluyó que la mejor forma de implementar las acciones que debe realizar el robot es por medio de una máquina de estados en la que, según las instrucciones dadas, se pase de un estado a otro para completar las tareas.

Después de lo anterior se realizaron lluvias de ideas centradas en los subproblemas del diseño. En primer lugar, en términos de energía se sugirieron principalmente fuentes de energía eléctrica en forma de batería como baterías de litio o baterías comunes AA. Se decidió además no considerar fuentes que no sean móviles, como conexiones directas a tomacorrientes o fuentes estáticas. Por otro lado, se consideró explorar otras formas de energía como la mecánica para el movimiento del robot. Además de esto se sugirió la búsqueda de un boost que sea capaz de cargar la batería o una placa que tenga un circuito de carga de batería para asegurar que el robot sea fácil de cargar.

Para las señales de los sensores, la discusión se centró en los sensores necesarios para generar la información requerida por la placa de microcontrolador para el funcionamiento correcto del robot. En primer lugar, se comentó que el robot necesita información acerca de su movimiento para el control de éste. El movimiento del robot será controlado por una aplicación en la que se definen distancias y giros específicos que deben de ser controlados,

por lo que se requiere un flujo de información con el que se define qué movimiento está siendo realizado y cómo se debe de mover el robot para completar la acción; por esta razón se consideran sensores que midan la rotación de las ruedas, como encoders rotatorios.

Además, se busca que el robot pueda reconocer obstáculos en su camino, detenerse y alejarse del obstáculo para evitar colisiones, por lo que se necesitan sensores que sean capaces de interpretar los obstáculos en el ambiente y suplementen esta información al robot, por lo que se sugieren sensores de proximidad como sensores infrarrojos o ultrasónicos. Este mismo tipo de sensores se consideran para detectar el suelo debajo del robot, ya que se busca que si éste se encuentra en una zona insegura en la que la superficie en la que se mueve puede tener vacíos en los que puede sufrir daños, estos se puedan detectar y el robot se detenga y aleje del peligro. En el caso de las señales de inicio y paro durante la lluvia de ideas surgieron múltiples soluciones para este subproblema en la forma de botones, interruptores o potenciómetros. Todos estos componentes permiten alimentar la información deseada, además son simples e intuitivos a la hora de implementar el robot a las aulas educativas.

Para el subproblema del protocolo de comunicación del robot con la aplicación se determinó una lista de los posibles protocolos aplicables a esta escala, estos corresponden a la comunicación serial, Bluetooth, Wi-Fi y radio frecuencia. Estos son protocolos que pueden ser desarrollados con el uso de diferentes placas de microcontroladores y son compatibles con el tipo de ambiente en el que se usará el robot.

Además, como parte de la discusión y las tormentas de ideas, se determinó que existe un cuarto subproblema que no surgió desde el inicio del procedimiento. Este consiste en la implementación de sensores y actuadores que pueden ser implementados en funciones educativas. Para este subproblema se creó mediante la lluvia de ideas una lista de posibles tipos de componentes, esta consiste en servomotores, luces LED, pantallas LCD, sensores de color, sensores de luz, micrófonos, parlantes, garras y sensores de temperatura.

Dentro de la búsqueda interna surgió una contradicción técnica y física. Esta consiste en que se busca mejorar la facilidad para la fabricación del robot con el objetivo de implementar un factor de construcción en su uso educativo, especialmente la conexión de los cables conectores. Pero, se busca mantener la durabilidad del robot para que las conexiones de este

no fallen de manera fácil. En este caso se aplicó la metodología TRIZ para idear posibles soluciones, los resultados se encuentran en la Tabla 12 y la

Tabla 13.

Tabla 12. Análisis de contradicciones, parámetros y principios de inventiva.

Variables para analizar	Contradicción física	Parámetros de Altshuller	Principios de inventiva.
Incrementar: Facilidad de fabricación del robot.	Cables conectores.	32. Facilidad de fabricación.	1. Segmentación, 3. Calidad local, 10. Acción anticipada, 32. Cambio de color.
Objetivo: Facilidad de fabricación.			
Perjudica: Durabilidad.	Pines de componentes.	14. Resistencia.	

Tabla 13. Desarrollo de propuesta de inventiva y solución.

Solución 1	
Principio de Inventiva	1. Segmentación
Propuesta de Inventiva	Se toma un enfoque de segmentación en el que se divide en partes independientes y fácil de desarmar.

Descripción de la Solución	Se crea un diseño en el que los cables tienen una estructura similar a jumpers, además de esto en cada componente se tienen pines con entradas macho o hembra. De esta forma, aunque el componente puede ser fácilmente conectado y desconectado, si el cable sufre de una fuerza mayor se desconectará y no se romperá.
Solución 2	
Principio de Inventiva	10. Acción anticipada
Propuesta de Inventiva	Debido a que se busca que el diseño sea fácil de armar, pero resistente se toman medidas anticipadas para la resistencia a posibles fuerzas aplicadas a los cables de conexión.
Descripción de la Solución	Generar una estructura que se coloque sobre la placa, similar a un “shell” que permita conectar los cables y después asegurarlos con un sistema de tornillo, como las entradas de alimentación que tienen las placas.

De esta forma concluye la búsqueda interna. Los resultados de ambas búsquedas deben de ser organizados y combinados para finalizar esta fase de la metodología. Para esto se generaron cinco árboles de clasificación de conceptos, uno para cada subproblema generado y uno para la decisión sobre la placa de microcontrolador que será usada para el robot. Estos árboles buscan presentar todos los conceptos posibles dentro de una categoría. Si se realiza una combinación que incluya un concepto por cada categoría se puede generar un posible candidato de diseño. Este es el proceso que se aplicará después de generar los árboles y será desarrollado con la ayuda de una tabla de combinación que permite plantear todos los diferentes candidatos de diseño [20].

Los árboles de clasificación de conceptos de la fuente de energía, los sensores que obtienen información para el movimiento, el protocolo de comunicación, los componentes para propósitos educativos y placas de microcontroladores se encuentran en la Figura 26, la Figura 27, la Figura 28, la Figura 29 y la Figura 30 respectivamente.

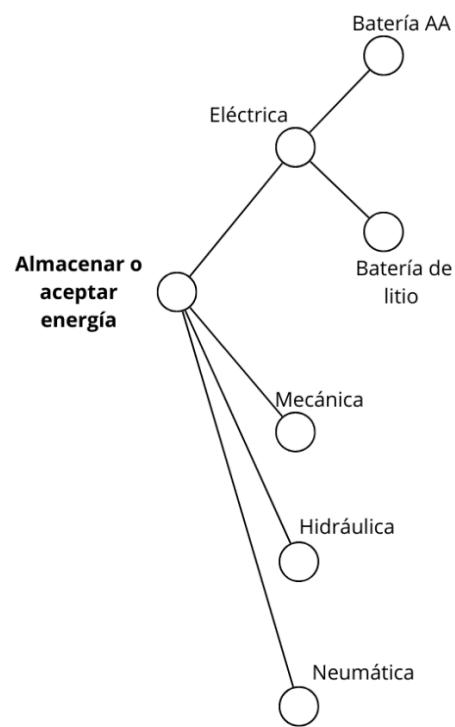


Figura 26. Árbol de clasificación de conceptos de la fuente de energía.

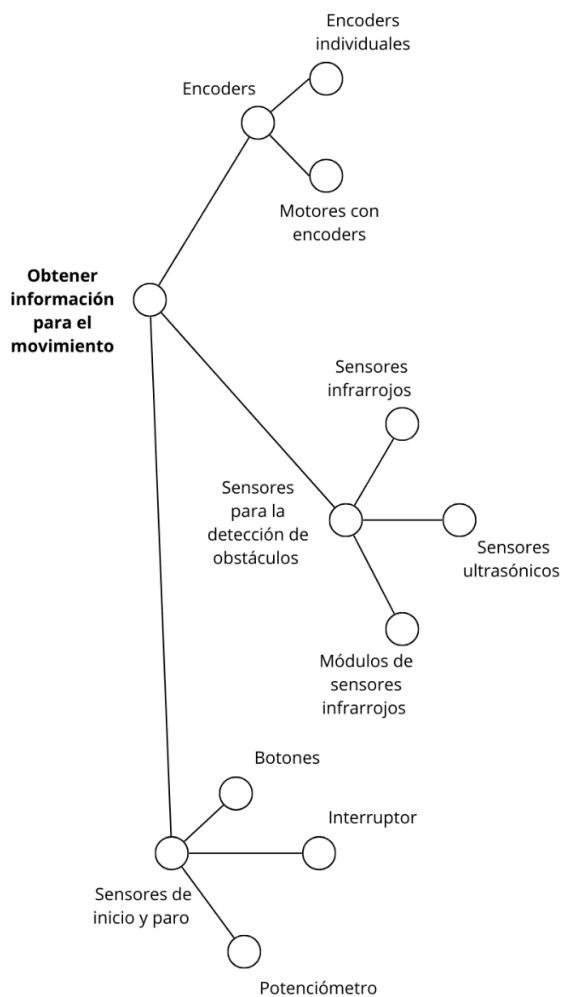


Figura 27. Árbol de clasificación de conceptos de los sensores de movimiento.

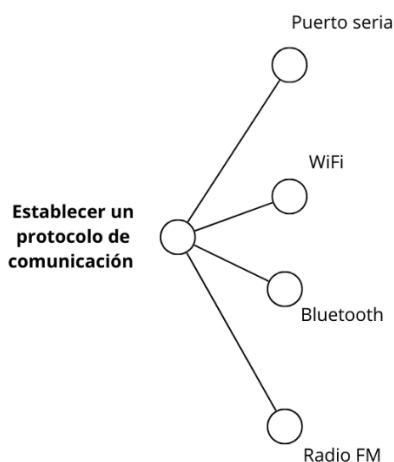


Figura 28. Árbol de clasificación de conceptos de los protocolos de comunicación.

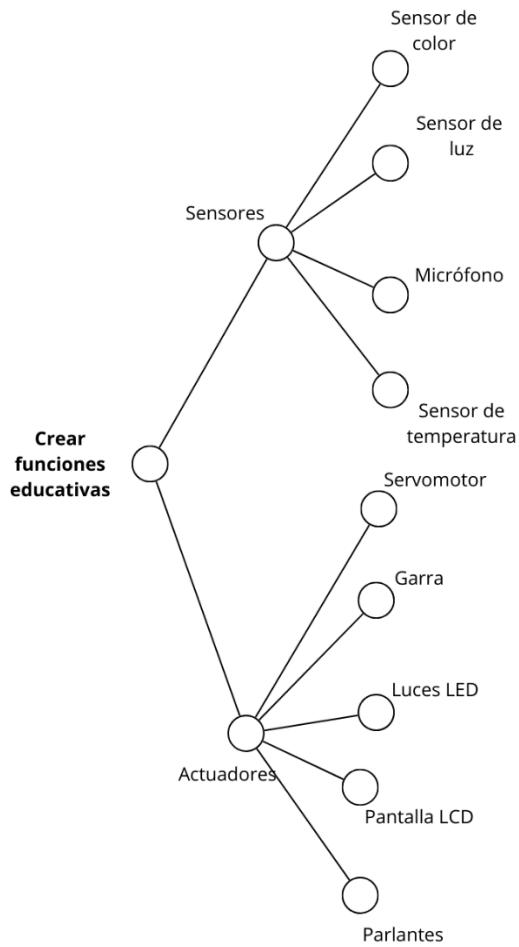


Figura 29. Árbol de clasificación de conceptos de los componentes educativos.

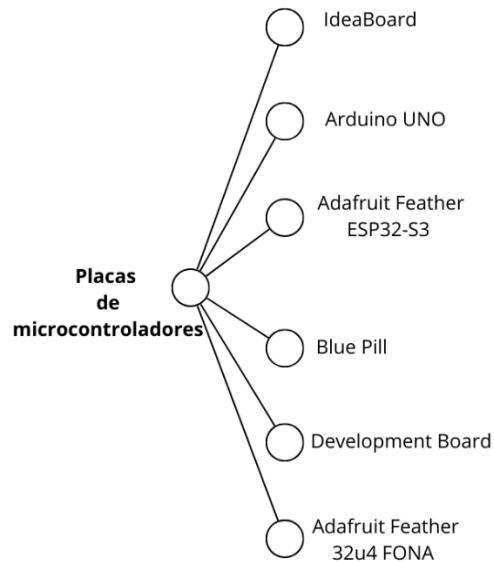


Figura 30. Árbol de clasificación de conceptos de placas de microcontroladores.

Para la tabla de combinación de conceptos se presentan aquellos que fueron considerados como más prominentes de cada uno de los árboles de clasificación ya generados. Estas combinaciones traen a la luz las soluciones candidatas del diseño final del robot educativo y fueron creadas con el objetivo de abarcar todos los posibles casos [20]. Se obtuvieron seis combinaciones, las cuales son presentadas en las tablas de combinación que corresponden a la Tabla 14, la Tabla 15, la Tabla 16, la Tabla 17, la Tabla 18 y la Tabla 19, con diagramas que representan su arquitectura electrónica en la Figura 31, la Figura 32, la Figura 33, la Figura 34, la Figura 35 y la Figura 36.

Tabla 14. Tabla de combinación para el primer candidato a la solución.

Fuente de energía	Sensores de movimiento	Protocolo de comunicación	Componentes educativos	Placa de microcontroladores	
Energía eléctrica: Baterías AA.	Encoder individual.	Puerto serial.	Sensor de color.	Blue Pill	
	Interruptor		Sensor de Luz		
	Sensores ultrasónicos.		Garra		
			Parlantes		
			Luces LED		
			Pantalla LCD		

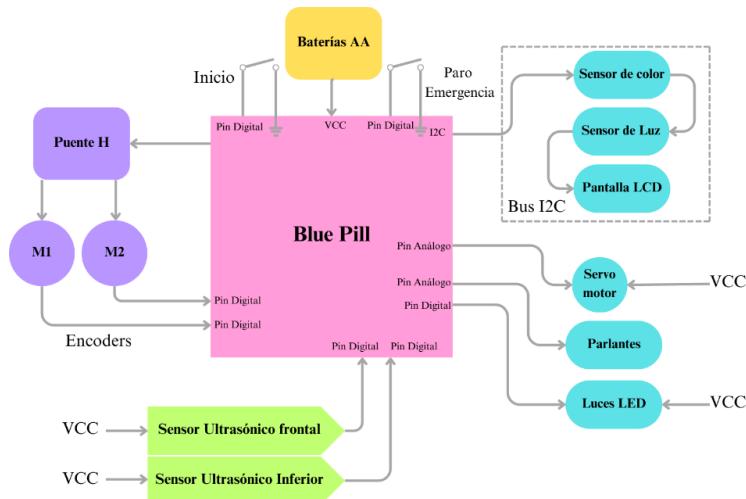


Figura 31. Diagrama del primer candidato a la solución.

Tabla 15. Tabla de combinación para el segundo candidato a la solución.

Fuente de energía	Sensores de movimiento	Protocolo de comunicación	Componentes educativos	Placa de microcontroladores
Energía eléctrica: Baterías AA.	Encoders individuales. Sensores ultrasónicos. Interruptor	Bluetooth	Sensor de color. Sensor de Luz Garra Servomotor Luces LED Pantalla LCD	IdeaBoard

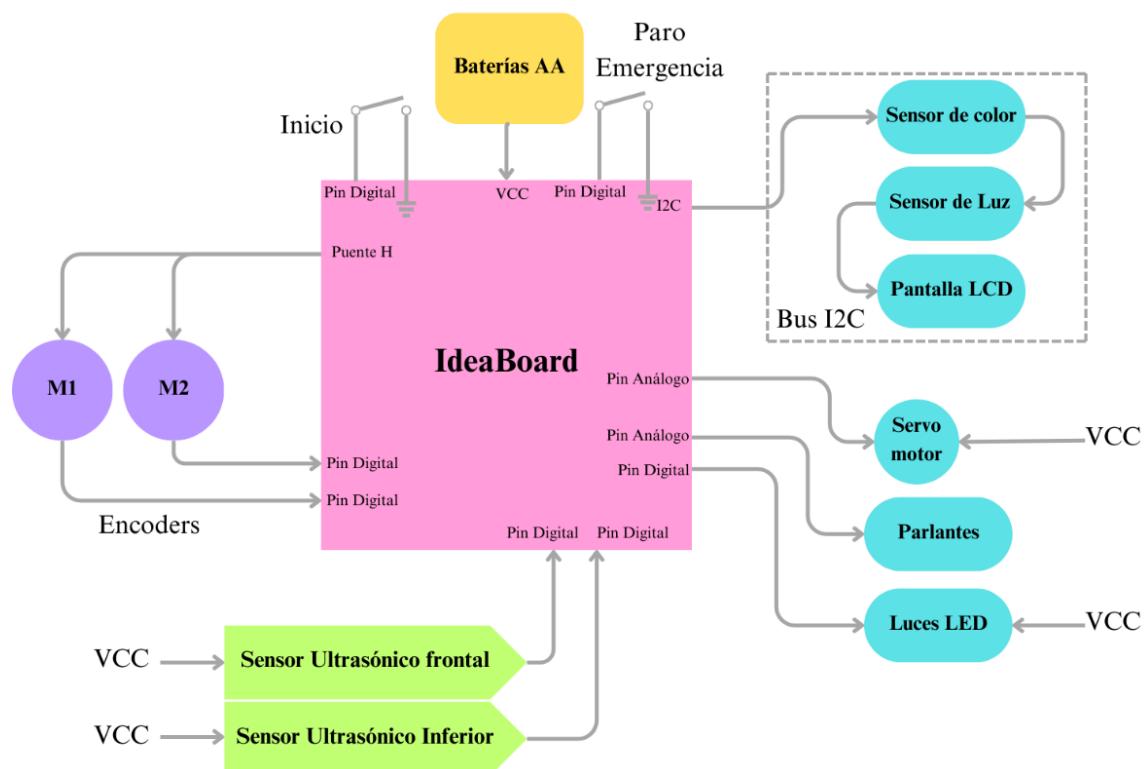


Figura 32. Diagrama del segundo candidato a la solución.

Tabla 16. Tabla de combinación para el tercer candidato a la solución.

Fuente de energía	Sensores de movimiento	Protocolo de comunicación	Componentes educativos	Placa de microcontroladores
Energía eléctrica: Batería de litio.	Motores con encoders Módulos de sensores infrarrojos Botones	Bluetooth	Sensor de color. Sensor de Luz Garra Servomotor Luces LED Pantalla LCD	IdeaBoard

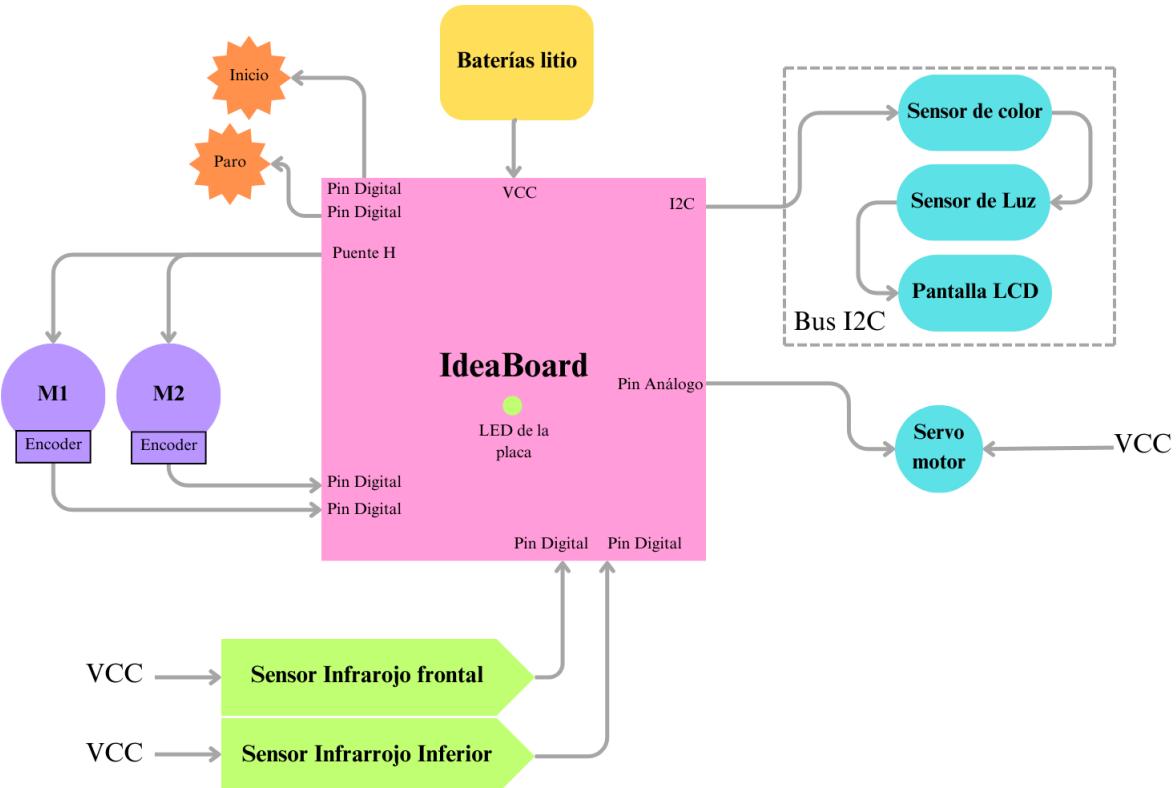


Figura 33. Diagrama del tercer candidato a la solución.

Tabla 17. Tabla de combinación para el cuarto candidato a la solución.

Fuente de energía	Sensores de movimiento	Protocolo de comunicación	Componentes educativos	Placa de microcontroladores	
Energía eléctrica: Batería litio	Motores con Encoders	Bluetooth	Sensor de color.	Arduino UNO	
	Módulos de sensores infrarrojos		Micrófono		
	Interruptor		Garra		
			Parlante		
			Luces LED		
			Pantalla LCD		

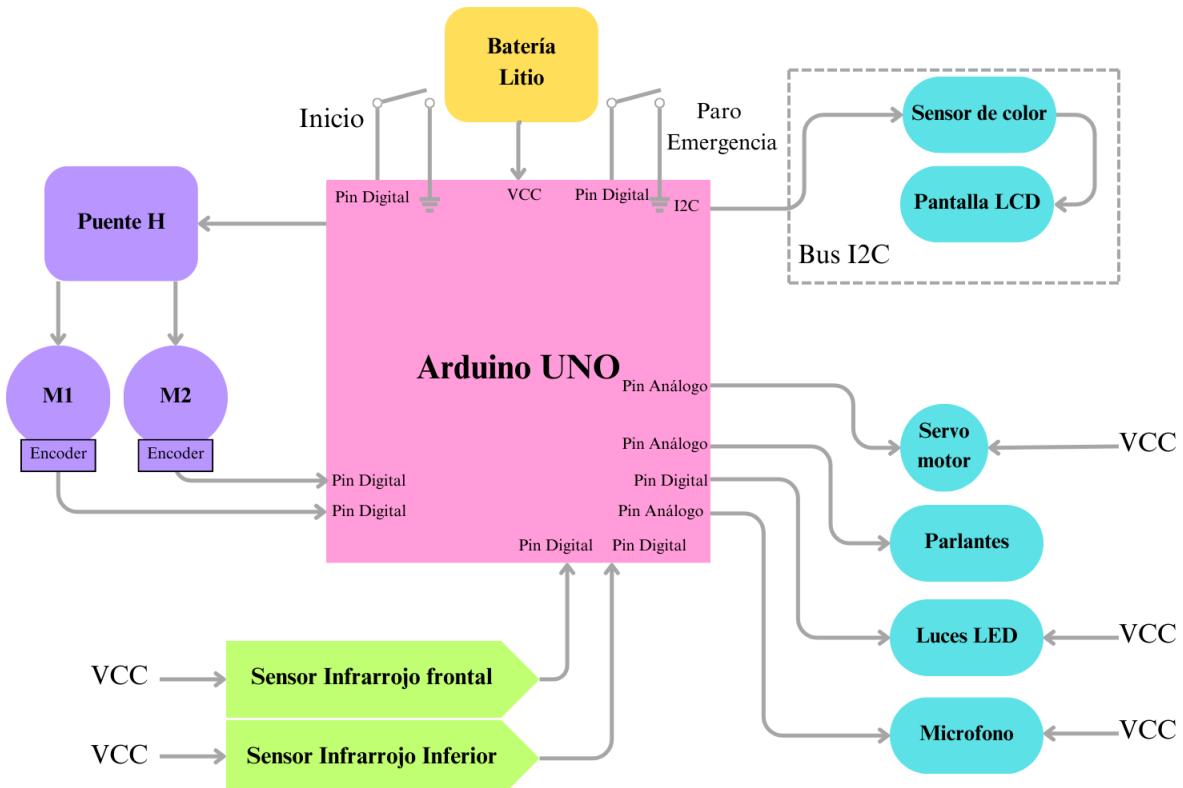


Figura 34. Diagrama del cuarto candidato a la solución.

Tabla 18. Tabla de combinación para el quinto candidato a la solución.

Fuente de energía	Sensores de movimiento	Protocolo de comunicación	Componentes educativos	Placa de microcontroladores
Energía eléctrica: Batería litio	Motores con Encoders	Bluetooth	Sensor de color.	Arduino UNO
	Sensores ultrasónicos.		Micrófono	
Energía Mecánica	Botones		Garra Parlante Luces LED Pantalla LCD	

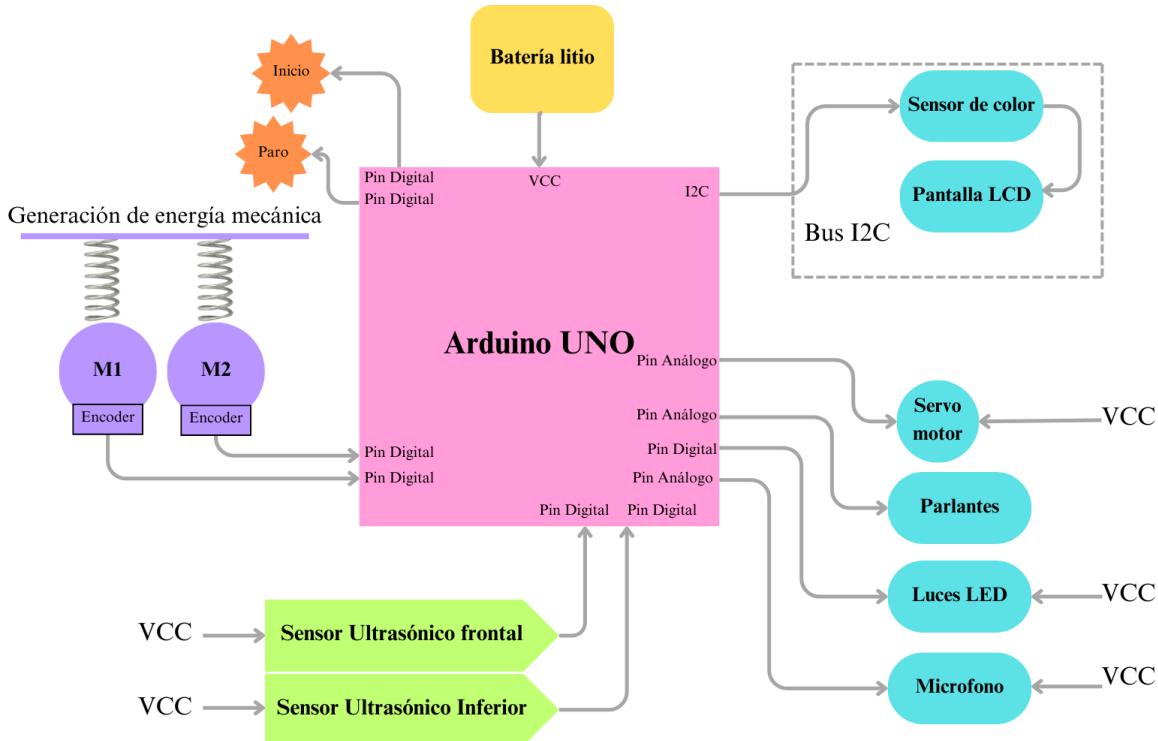


Figura 35. Diagrama del quinto candidato a la solución.

Tabla 19. Tabla de combinación para el sexto candidato a la solución.

Fuente de energía	Sensores de movimiento	Protocolo de comunicación	Componentes educativos	Placa de microcontroladores
Energía eléctrica: Batería litio	Motores con Encoders Módulos de sensores infrarrojos. Botones	Wi-Fi	Sensor de color. Sensor de temperatura Garra Luces LED Parlante Pantalla LCD	Adafruit Feather ESP32-S3

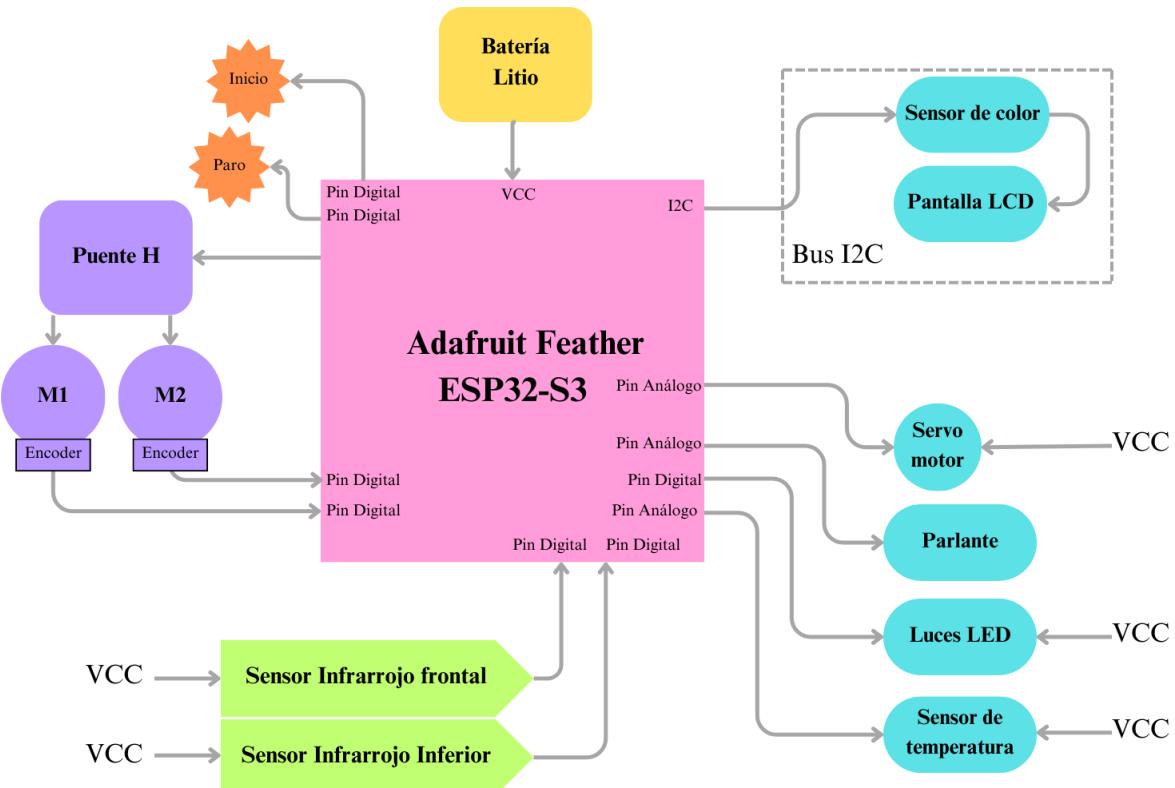


Figura 36. Diagrama del sexto candidato a la solución.

Con esto se concluye el proceso de conceptualización del proceso de diseño. Se puede inferir del proceso que se exploraron todas las posibles opciones a la hora de generar los conceptos y que además de esto, fueron combinados para formar propuestas de diseño de forma que se cubren todas las bases y la variedad brindada por los métodos de búsqueda externa e interna. Además, si más adelante se determinara que la combinación de diferentes propuestas es ideal, estas son compatibles entre sí y pueden ser combinadas. Aunque el problema pudo haber sido descompuesto en otras formas alternativas, con una mayor cantidad de subproblemas, la combinación de estos siempre traería a el mismo tipo de candidatos de diseño, por lo que esta descomposición también es exitosa.

3.5 Proceso de Selección y Evaluación de Conceptos

La etapa de selección del concepto corresponde al último paso de la metodología para llegar a la propuesta de diseño. Debido a que en el proceso que se llevó a cabo se tienen múltiples conceptos que pueden ser elegidos como el que será desarrollado y validado, la construcción de prototipos de todos estos conceptos para determinar cuál es el diseño ideal es un proceso exhaustivo y en muchos casos innecesario o incluso no es posible. Por esta razón, usando la información que se recolectó con las necesidades y que se definió con las especificaciones se puede elegir el diseño que en papel es ideal para ser construido y validado, aquí se encuentra la importancia de esta etapa.

Para esta selección se desarrollan dos fases, la primera de las fases se conoce como el filtrado y esta fase es la responsable de eliminar aquellos conceptos que son claramente desfavorables, ayuda a mejorar conceptos, permite la combinación de conceptos en caso de que esto sea el camino ideal y da una idea del rendimiento de los conceptos según lo que se busca en el diseño final.

Para la primera etapa se construyó una matriz en forma de tabla, en esta se colocó un concepto por evaluar en cada columna, y en cada fila un criterio que será evaluado. Estos criterios se desarrollaron tomando en cuenta las necesidades planteadas anteriormente. Además, se eligió al segundo concepto como la referencia, ya que este es un concepto sencillo de entender y aparentemente neutro y facilita la comparación con los conceptos planteados para su rediseño, ya que en base a este se clasificaron el resto como “+” si es mejor en ese criterio que la referencia, “0” si es igual y “-” si es peor. En la Tabla 20 se encuentra el resultado.

Tabla 20. Etapa de filtrado de la selección de conceptos.

Criterios de selección	Conceptos					
	A Primer concepto	B Segundo concepto	C Tercer concepto	D Cuarto concepto	E Quinto concepto	F Sexto concepto
Rediseño del Atta-Bot.	0	0	0	0	0	0
Simplificación del Atta-Bot.	0	0	0	0	-	0
Inclusión del aspecto educativo.	0	0	0	0	-	-
Precio.	+	0	0	-	-	-
Calidad del sistema de movimiento.	0	0	+	+	+	+
Precisión de los movimientos.	0	0	0	0	-	0
Tiempo de autonomía.	0	0	+	+	+	+
Eliminación de componentes no educativos.	0	0	0	0	-	0
Vinculación con aplicación móvil.	-	0	0	0	0	0
Sensor de color	0	0	0	0	0	0
Garra	0	0	0	0	0	0
Sostenedor de lápices	0	0	0	0	0	0

Sistema de detección de obstáculos.	0	0	+	+	0	+
Precisión de acciones.	0	0	0	0	0	0
Seguridad	0	0	+	+	+	+
Indicador de las acciones realizadas.	0	0	0	0	0	0
Señalización del frente.	0	0	0	0	0	0
Fácil de cargar la batería.	0	0	-	-	-	-
Botones de inicio y paro.	0	0	+	0	+	+
Capacidad modular	0	0	+	0	+	+
Robustes	0	0	+	+	-	+
Menor tamaño	0	0	0	0	-	0
Software y hardware abierto	0	0	0	0	0	0
Suma +	1		7	5	5	7
Suma 0	21	23	15	17	10	13
Suma -	1		1	2	8	3
Evaluación neta	0	0	6	3	-3	4
Lugar	4	4	1	3	5	2
Continuación	No	No	Si	Si	No	Si

Como se observa en la Tabla 20 el concepto ganador de la primera etapa corresponde al tercer concepto planteado con una evaluación neta 6, de segundo lugar se tiene el sexto concepto

con una evaluación de 4 y de tercer lugar se tiene el cuarto concepto con una evaluación de 3. Estos son los conceptos con los que se decidió continuar debido a que fueron aquellos que destacaron como la mejor probabilidad de ser el concepto ganador.

No se decidió mejorar ni combinar ninguno de los conceptos perdedores, esto se debe a que estas acciones tendrían como resultado un concepto igual a uno de los ganadores de la etapa. Estos conceptos ya serán evaluados más tarde por lo que no sería una comparación necesaria. Por ende, se eliminan los conceptos A, B y E.

La segunda etapa es conocida como la evaluación y consiste en la verdadera matriz de selección y evaluación de conceptos. Para esta matriz se mantuvieron los mismos criterios, pero esta vez por cada criterio se decidió uno de los conceptos como la referencia que corresponde a aquél que sea neutral en la categoría siendo evaluada. Además, esta vez se calificó cada uno de los criterios con un número del 1 al 5 en el que 3 es la referencia, 2 es ligeramente menor, 1 es notablemente menor, 4 es ligeramente mayor y 5 es notablemente mayor. Solo se usaron los conceptos ganadores de la etapa anterior.

Por último, se incorporó una columna de evaluación ponderada, el objetivo de esta corresponde a incorporar el peso de la importancia dada a las necesidades por parte del cliente, por ende, a cada criterio se le asignó un porcentaje de importancia, la suma de estos iguala a 100%, este porcentaje se multiplicó por la clasificación dada para obtener la evaluación ponderada, la suma de estas para cada concepto definió al ganador. Los resultados se encuentran en la Tabla 21.

Tabla 21. Etapa de evaluación de la selección de conceptos

Criterios de selección	Peso	Concepto					
		A Tercer Concepto		B Cuarto Concepto		C Sexto Concepto	
		Calificación	Evaluación ponderada	Calificación	Evaluación ponderada	Calificación	Evaluación ponderada
Rediseño del Atta-Bot.	7%	3	0.21	3	0.21	3	0.21

Simplificación del Atta-Bot.	5%	3	0.15	3	0.15	3	0.15
Inclusión del aspecto educativo.	7%	2	0.14	3	0.21	3	0.21
Precio.	1%	4	0.04	2	0.02	3	0.03
Calidad del sistema de movimiento.	1%	4	0.04	3	0.03	3	0.03
Precisión de los movimientos.	3%	3	0.09	3	0.09	3	0.09
Tiempo de autonomía.	2%	3	0.06	3	0.06	3	0.06
Eliminación de componentes no educativos.	5%	4	0.2	3	0.15	3	0.15
Vinculación con aplicación móvil.	7%	3	0.21	3	0.21	2	0.21
Sensor de color	2%	4	0.08	3	0.06	3	0.06
Garra	2%	3	0.06	3	0.06	3	0.06
Sostenedor de lápices	3%	3	0.09	3	0.09	3	0.09
Sistema de detección de obstáculos.	3%	3	0.09	3	0.09	3	0.09
Precisión de acciones.	7%	4	0.28	3	0.21	3	0.21
Seguridad	7%	3	0.21	2	0.14	3	0.21

Indicador de las acciones realizadas.	5%	3	0.15	3	0.15	3	0.15
Señalización del frente.	3%	3	0.09	3	0.09	3	0.09
Fácil de cargar la batería.	5%	3	0.15	3	0.15	3	0.15
Botones de inicio y paro.	5%	5	0.25	3	0.15	5	0.25
Capacidad modular	5%	5	0.25	2	0.1	3	0.15
Robustes	5%	5	0.25	2	0.1	3	0.15
Menor tamaño	3%	3	0.09	2	0.06	3	0.09
Software y hardware abierto	7%	3	0.21	3	0.21	3	0.21
Total de puntos		3.39		2.79		3.1	
Lugar		1		3		2	

En conclusión, como se puede observar en la Tabla 21, el ganador del proceso de evaluación corresponde al tercer concepto. Se evitaron sesgos durante el proceso ya que las calificaciones se otorgaron basados únicamente en la parte del diseño que evalúa el criterio y no en otras partes. El tercer y sexto concepto tienen un total de puntos similar, por lo que ambos pueden ser opciones para la solución del problema, pero, en este caso, se procedió a diseñar a profundidad el concepto ganador, construir un prototipo y validar el mismo. Es importante reconocer que este es un proceso de reiteración por lo que en el si en el proceso de diseño se encuentran claras fallas con el concepto ganador, siempre existe la posibilidad de mejorar el concepto o elegir proceder con el proceso con el segundo lugar de la evaluación.

Por último, se debe de reconocer que el concepto está diseñado para cumplir las acciones básicas listadas por el cliente. Es decir, este diseño genera un robot educativo basado en el

robot existente Atta-Bot que se puede conectar con una aplicación móvil para ser alimentado instrucciones. Estas instrucciones corresponden a avance, retroceso, giro a la derecha, giro a la izquierda, reconocimiento de obstáculos y creación de ciclos. Todas las acciones de movimiento pueden ser configuradas con una medida exacta, ya sea en centímetros o grados, y efectuar estas con precisión.

Capítulo 4.0 Propuesta de Diseño

En esta sección se desarrollará en profundidad el proceso de diseño del Atta-Bot educativo. Detallando factores de este como la lista de componentes eléctricos elegidos, la arquitectura eléctrica y la estructura y lógica de su código.

4.1 Diseño Electrónico

Como se puede observar en la Figura 33 ya se tiene un diseño básico de la estructura electrónica que tendrá el robot. El siguiente paso que se realizó corresponde a la elección de los componentes individuales que se implementaran en el diseño. Estos componentes serán elegidos de la Tabla 11 la cual presenta aquellos disponibles dentro de los distribuidores costarricenses ya que, como se ha mencionado anteriormente, una de las ventajas del diseño original del robot corresponde a que todas sus partes pueden ser conseguidas dentro del país. La lista de componentes BOM se encuentra en el Apéndice 1.

El único componente que ha sido elegido anteriormente es la placa de microcontrolador que corresponde a la IdeaBoard. Esta placa no solo es elegida debido a que se encontraba en el concepto ganador, sino que posee otros beneficios ya que fue diseñada en Costa Rica para su uso en proyectos de robótica e internet de las cosas. En primer lugar, tiene una gran ventaja para la implementación de las conexiones modulares debido a que esta presenta pines machos en todos sus pines. Además de esto la placa tiene un par de puentes H que son usados como el driver para cada uno de los motores lo que simplifica las conexiones que deben ser realizadas y elimina la necesidad de la implementación de un driver externo. Por otro lado, la placa posee un neopixel, el cual puede ser controlado. Este puede ser usado para las pruebas del código, así como señalar diferentes problemas que pueden surgir durante la ejecución de las instrucciones. Además, la placa puede tener todos sus pines con una lógica de 3.5V o puede separar la mitad de los pines en 5V y 3.5V. Por último, posee la capacidad de usar los protocolos de Wi-Fi y Bluetooth para la conexión del robot con la aplicación móvil. En la Figura 37 se encuentra la placa IdeaBoard. [33]



Figura 37. Placa IdeaBoard elegida para el diseño. [33]

Los siguientes componentes que se deben elegir corresponden a los motores. Como se describió anteriormente existe una necesidad de mantener el diseño del subsistema de movimiento similar al del Atta-Bot. Esto se debe a que la precisión y el funcionamiento del sistema original es ideal para la aplicación educativa que se busca establecer. Además, el concepto define que estos motores deben de tener un encoder. Por último, deben de ser pequeños para adecuarse al tipo de robot que se busca diseñar. Por estas razones se eligió el mismo tipo de motores CD usados en el diseño original del Atta-Bot que tienen encoders y ruedas. [33] Estos motores son identificados con el código ROB-CB1019 y se pueden encontrar en la Figura 38.



Figura 38. Motores CD elegidos para el diseño. [33]

El siguiente componente por elegir corresponde a la batería. Según el concepto ganador esta es una batería de litio. El factor de elección para esta batería es el amperaje de esta, ya que todas las baterías litio disponibles proporcionan un voltaje de 3.7V. Entre las opciones se tienen amperajes de 600mAh, 2500mAh, 400mAh y 100mAh. Debido a que los mayores consumidores de amperaje corresponden a los motores y servomotores la decisión es tomada en torno a estos. El diseño original del Atta-Bot posee dos motores CD y un servomotor, este también es el caso para este nuevo robot, por lo que se decidió utilizar la misma batería de litio de 2500mAh para este nuevo diseño, la cual se encuentra en la Figura 39.



Figura 39. Batería de litio de 2500mAh elegida para el diseño. [33]

Como se mencionó anteriormente las baterías de litio proporcionan un voltaje de 3.7V, pero para componentes como los motores CD, servomotores y algunos sensores es necesario un voltaje mayor, por esta razón se necesita un “boost” o amplificador para la batería. Existen dos opciones disponibles en el país, una de estas es un componente que eleva el voltaje a 5V y protege en contra de picos de corriente, el segundo componente además añade un circuito de carga para la batería [33]. Por esta razón se eligió el segundo componente de código ADA-2465 ya que una de las necesidades del robot corresponde a la facilidad de su carga. En la Figura 40 se encuentra la imagen del PowerBoost. Con este componente se termina el subsistema de fuente de energía.

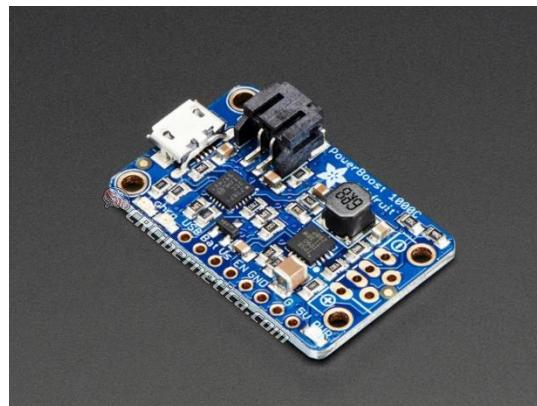


Figura 40. PowerBoost elegido para el diseño. [33]

Para continuar con el subsistema de movimiento se debe de seleccionar un tercer punto de apoyo ya que el robot está diseñado para ser diferencial. Entre las opciones se tienen dos tipos de ball caster uno de plástico y uno de metal, en este caso se eligió nuevamente el componente usado en el robot original debido a que tiene una buena compatibilidad con los motores y las ruedas además que entre ambas opciones se adapta mejor al diseño por que al estar hecho de plástico, su peso es menor. En la Figura 41 se encuentra la imagen del componente cuyo código corresponde a ROB-CB1818.



Figura 41. Ball caster seleccionado para el diseño. [33]

Para los botones, debido a que se desea tener múltiples de estos y que tengan un pequeño tamaño se decide usar un teclado de botones, el de menor tamaño posee 4 botones, por lo que serán usados dos para el inicio y el paro y el resto pueden ser implementados para otras acciones educativas en futuras iteraciones del robot, su código corresponde a COM-CB1333 y se encuentra en la Figura 42.

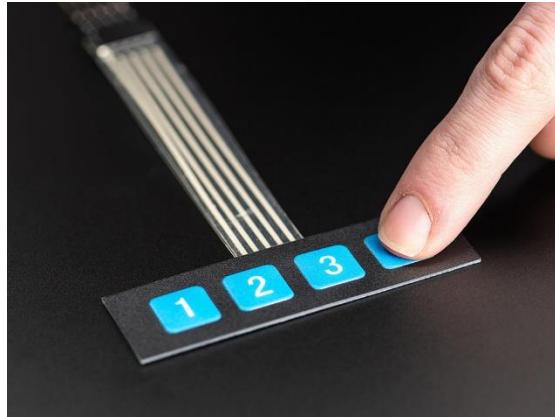


Figura 42. Teclado de botones elegido para el diseño. [33]

Para el reconocimiento de obstáculos, según el concepto ganador se busca implementar sensores infrarrojos frontales en un módulo que contiene un circuito de calibración, esta es la mejor opción debido a que facilita su conexión sin la necesidad de soldadura. Además, se eligió un sensor infrarrojo sobre uno de ultrasonido porque estas tienden a ser más precisos en sus versiones más simples. Existe una única opción de este tipo disponible, su código corresponde a SEN-CB0006 y su imagen se encuentra en la Figura 43. Cabe destacar que tiene una lógica de 3.3V a 5V la cual es compatible con la placa IdeaBoard y funciona al incorporar un receptor y un emisor de luz infrarroja, al encontrar un obstáculo la luz se refleja en este, llega al receptor y produce una señal digital baja para indicar la existencia del obstáculo [33]. Se colocan tres en el diseño final. El módulo posee dos potenciómetros para calibrar el nivel de sensibilidad del sensor.

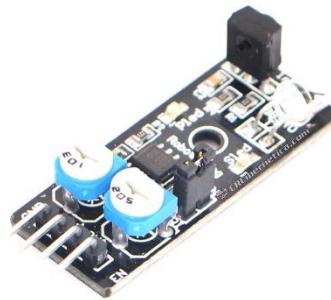


Figura 43. Módulo de sensor infrarrojo seleccionado para el diseño. [33]

Los sensores elegidos anteriormente corresponden a los que serán usados en el frente del robot para detectar obstáculos, pero también son necesarios sensores en la parte inferior para que el robot conozca si se encuentran agujeros, vacíos u otros peligros en la superficie en la que se mueve. En este caso también se seleccionaron módulos con sensores infrarrojos que funcionan de la misma forma que los descritos anteriormente. La diferencia consiste en que los elegidos para la parte inferior fueron estructurados para este uso y el emisor y receptor están apuntando hacia abajo y no hacia el frente. Su lógica también es de 3.3V o 5V por lo que es compatible con la placa seleccionada [33]. Se colocan dos en el diseño final. Su código corresponde a SEN-CB0005 y su imagen se encuentra en la Figura 44. Con esto se completó la elección de los componentes para la detección de obstáculos.



Figura 44. Módulo de sensor infrarrojo inferior seleccionado. [33]

Se inicia la elección de los componentes que serán aplicados para las funciones educativas fuera del movimiento del robot. En primer lugar, según el concepto ganador se incorpora un sensor de color y un sensor de luz. Por esta razón se eligió el sensor APDS9960 que corresponde a un sensor económico de proximidad, luz, RGB, y gestos. De esta forma, se cubren ambas necesidades con un solo componente. Además, el sensor posee una conexión Qwiic, debido a que funciona con el protocolo I2C. Este tipo de conexión es ideal para conexiones modulares y la placa IdeaBoard tiene un puerto Qwiic por lo que este sensor es compatible y puede ser conectado sin soldadura [33]. En la Figura 45 se encuentra una imagen del sensor.

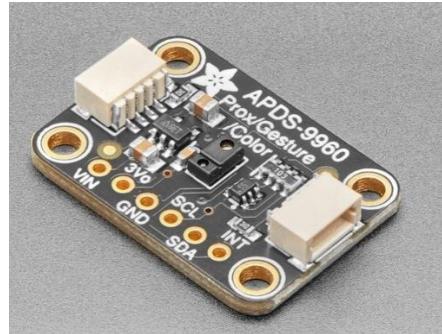


Figura 45. Sensor de luz y color elegido para el diseño. [33]

Para la garra, se busca que esta tenga un único grado de libertad, por lo que es necesario únicamente un motor para generar el movimiento. Los motores ideales para este tipo de aplicación corresponden a los servomotores. En este caso se decide usar un servomotor con un rango de 180° que permite un mayor nivel de control sobre el movimiento y la posición. Este corresponde al servomotor MG90S, en la Figura 46 se encuentra una imagen de este. El servomotor es compatible con la placa Ideaboard también y posee cables con entradas hembra por lo que puede ser conectado de forma modular y no necesita soldadura. [33]



Figura 46. Servomotor elegido para el diseño. [33]

El siguiente actuador que debe ser seleccionado corresponde a la pantalla LCD. En este caso se desea que la pantalla sea de un tamaño adecuado para el robot, además de esto la pantalla debe estar enfocada para comunicar mensajes a jóvenes, por esta razón, se prefiere una pantalla a color. Dentro de la búsqueda interna de las posibles pantallas, el modelo que mejor se adapta a estas necesidades corresponde a una pantalla de 160x80 pixeles a color y una

diagonal de 0.96 pulgadas [33]. Esta es identificada con el código LCD-CB4557, Se representa en la Figura 47.



Figura 47. Pantalla LCD a color elegida para el diseño. [33]

A parte de estos componentes que representan las necesidades descritas por el cliente, para la correcta construcción del robot se necesita de otros componentes auxiliares para su funcionamiento. El primero de estos corresponde a un interruptor, este será colocado entre tierra y el pin de “enable” del PowerBoost. Este pin, al ser conectado a tierra detiene el flujo de corriente de la batería a la placa microcontroladora, apagando el robot [33]. Para este interruptor se eligió el mismo que utiliza el diseño original debido a que su tamaño y su simplicidad es ideal para este uso, su código corresponde a COM-CB0372 y su imagen se encuentra en la Figura 48.



Figura 48. Interruptor elegido para el diseño. [33]

Para las conexiones, como se discutió en la sección de búsqueda interna, se desarrolló el concepto de conexiones con pines hembras y machos, para evitar el uso de soldadura y mantener una conexión modular para el robot. Se decidió esta opción debido a que el proceso de diseño de un Shell para la IdeaBoard significaría una mayor cantidad de tiempo y su precio

sería mayor. Para realizar las conexiones de la mejor forma posible se eligió un cable para conexiones UL2464 de 1 metro y un set de construcción de jumpers. Sus códigos son CAB-CB5215 y KIT-CB0012 respectivamente y sus imágenes se encuentran en la Figura 49 y la Figura 50 respectivamente.



Figura 49. Cables elegidos para el diseño. [33]



Figura 50. Kit de construcción de jumpers seleccionado para el diseño. [33]

Para la conexión de la pantalla y el sensor de color y luz se utilizan cables Qwiic. En el caso del sensor, este posee dos puertos hembra al igual que en la placa IdeaBoard, por lo que el primero de los cables debe ser un cable macho-macho. Por otro lado, la pantalla no posee un puerto, por lo que el segundo cable debe ser macho a 4 pines [33]. Sus códigos corresponden a CAB-CB4925 y ADA-4209. Se representan en la Figura 51 y la Figura 52.



Figura 51. Cable Qwiic macho-macho elegido para el diseño. [33]

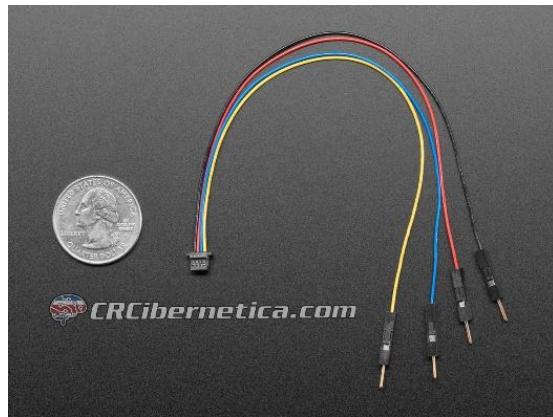


Figura 52. Cable Qwiic macho a 4 pines elegido para el diseño. [33]

Por último, se debe tener una forma de sostener los componentes dentro de la carcasa para que estos se mantengan en su lugar, para esto se eligen separadores, tuercas y tornillos de nylon M3. Se utiliza el material de nylon debido a que este no es conductor por lo que no interrumpe las conexiones ni crea cortos en el circuito [33]. Estos pueden ser conseguidos como un set con el código TOL-CB0513 y su imagen se muestra en la Figura 53.



Figura 53. Kit de tornillos, separadores y tuercas de nylon elegido para el diseño. [33]

Una vez se seleccionaron todos los componentes se realizó un diagrama electrónico en el que se especifican cada una de las conexiones del circuito del robot. Este puede ser utilizado en el futuro para la construcción del robot tanto para el cliente como para la futura aplicación del robot en aulas donde será construido. El resultado se encuentra en la Figura 54.

Además, se realizó un diagrama de la forma en la que los componentes serán estructurados físicamente en el robot. Como se mencionó anteriormente se utilizaron tres sensores infrarrojos frontales, el primero de estos se posiciona en el centro de la cara frontal y los dos restantes a cada lado con un ángulo de 45° , esto con el objetivo de abarcar un mayor rango a la hora de detectar obstáculos. Por otro lado, los dos sensores infrarrojos inferiores se posicionan frente a cada una de las ruedas para que sean capaces de detectar cualquier peligro antes que el robot avance por esta zona. El diagrama de la distribución física de los componentes se encuentra en la Figura 55. Con esto concluye el desarrollo del diseño electrónico del robot.

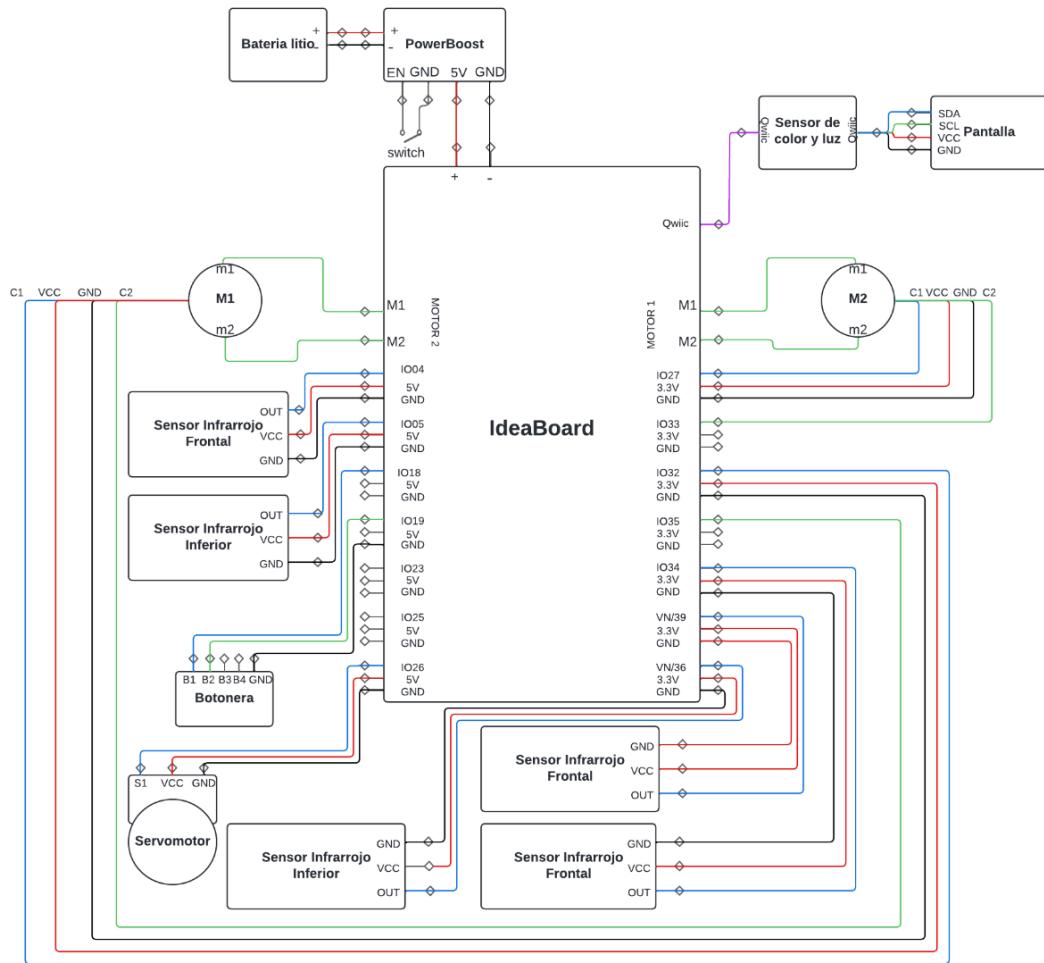


Figura 54. Diagrama electrónico del Atta-Bot.

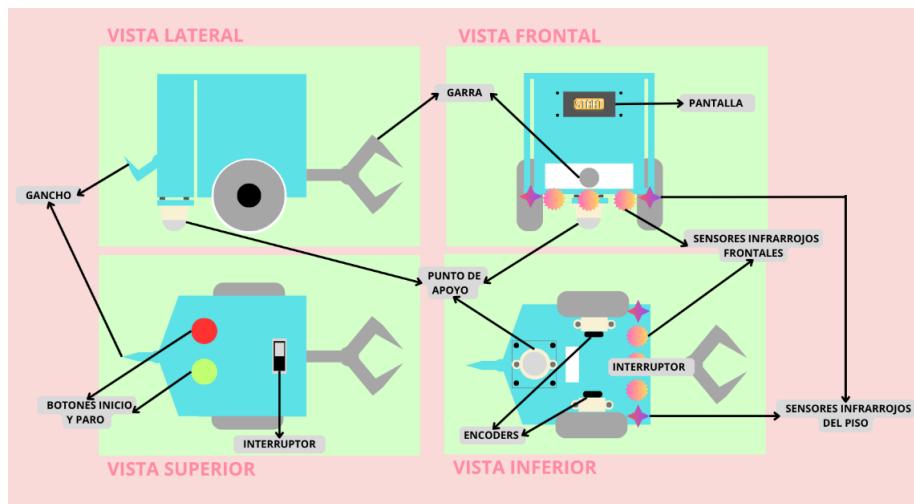


Figura 55. Diagrama de la distribución física de los componentes del Atta-Bot.

4.2 Diseño del código

Una vez se completó el diseño electrónico se procedió con el diseño del código. La primera etapa del desarrollo del código corresponde a decidir qué lenguaje será utilizado. Se tienen dos opciones, las cuales corresponden a Circuit Python y Arduino IDE. Circuit Python es el uso del lenguaje de Python para la programación de placas de microcontroladores, este es el lenguaje recomendado por los creadores de la IdeaBoard y la placa posee algunas bibliotecas en este lenguaje que permiten acceso a sus características más deseables. Por otro lado, Arduino IDE corresponde al lenguaje más común para la programación de microcontroladores, basándose en C++. Este tiene una amplia selección de bibliotecas que sirven de apoyo en el desarrollo del código como aquellas que pueden ser usadas para la lectura de encoders, el control de neopixeles, entre otros. Además, el código del Atta-Bot original fue desarrollado en Arduino IDE.

Al tomar en cuenta todos estos aspectos se determina que el lenguaje ideal para el proyecto corresponde a Arduino IDE, con este se puede usar el código del diseño anterior como una base. Por otro lado, se busca incorporar múltiples bibliotecas que solo se encuentran en este idioma para la simplificación del código y en algunos casos mejorar los resultados.

Como se describió en la búsqueda interna, una de las ideas cruciales de la lluvia de ideas corresponde al uso de una máquina de estados como la base del código. Este fue el mismo método aplicado en el código de la versión de enjambre del robot y es ideal para sistemas simples que necesitan de una constante realimentación de información. Por ende, se construyó una máquina de estados para este rediseño. En la Figura 56 se encuentra el diagrama de estados que define la lógica de cambio de estado y los estados establecidos. Como se puede observar en el diagrama se tiene nueve estados en total, los cuales corresponden a ESPERA, LEE_MEMORIA, MOVERSE, GIRAR, DETENERSE, CICLO, OBSTACULOS, MOVIMIENTO_OBSTACULOS y NADA. Cada uno de estos estados desempeña diferentes acciones que permiten el cumplimiento de las instrucciones programadas en la aplicación móvil.

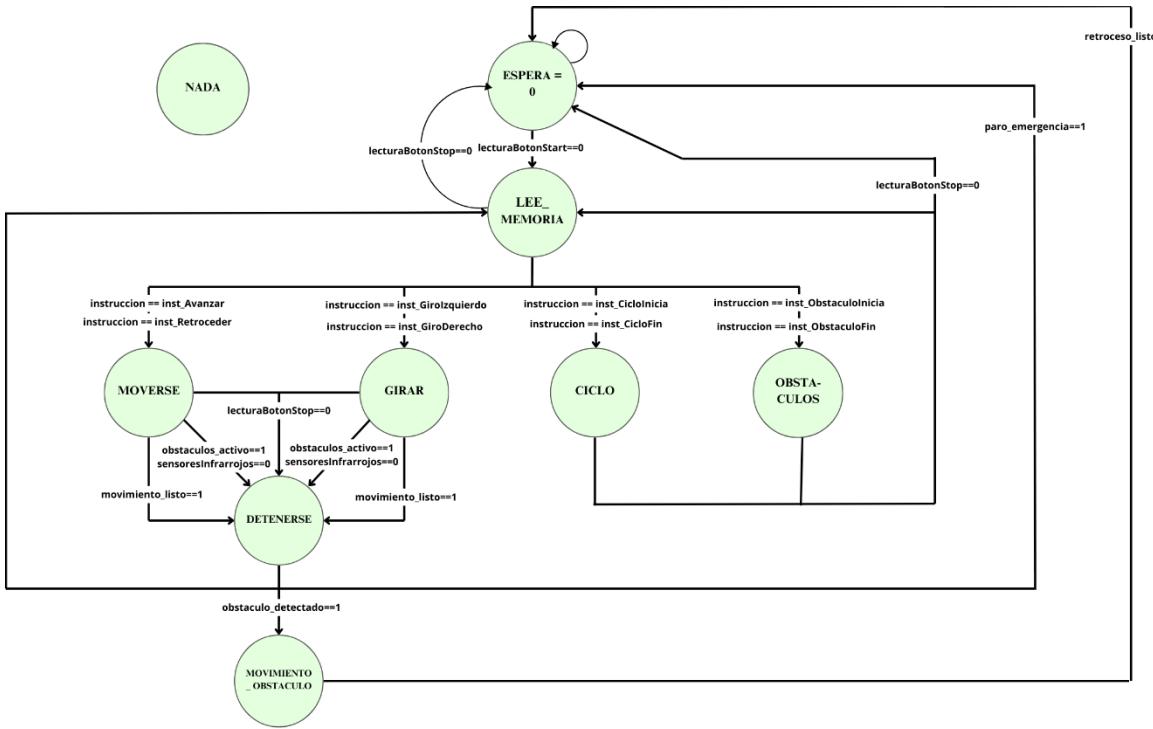


Figura 56. Diagrama de estados del Atta-Bot

El estado **ESPERA** corresponde a aquel en el que se inicializa el código, en este estado la máquina se encuentra constantemente en ciclo esperando la llegada de instrucciones desde la aplicación móvil por medio del protocolo de Bluetooth. Este mensaje llega codificado desde la aplicación y no será descodificado en este estado. Su lógica de cambio de estado depende de la señal creada al presionar el botón de inicio lo cual hace que la máquina pase al estado **LEE_MEMORIA**.

En este nuevo estado se llama a la función `Interpreta_mensajeBLE`, esta corresponde a la función que interpreta el mensaje recibido de la app de programación revisando el texto enviado por Bluetooth BLE. Primero interpreta de cinco en cinco los caracteres después de esto los guarda en un array global al interpretar qué instrucción es y qué valor tienen. Los grupos de caracteres tienen diferentes significados esto corresponden a los siguientes: ATINI marca el inicio de las instrucciones, ATFIN define el final de las instrucciones, OBINI representa el inicio de la habilitación de la detección de obstáculos, OBFIN se usa para señalizar el final de la detección de obstáculos, CIFIN corresponde a la instrucción que finaliza las acciones dentro de un ciclo, CI seguido de tres números representa el inicio de las instrucciones de un ciclo y los números definen cuantos ciclos se realizarán, el comando

AV seguido de tres números se usa para el avance y los números son la distancia de avance en centímetros, esto mismo con RE pero en este caso se retrocede y por último, los caracteres GI o GD seguidos de tres números definen que se gira a la izquierda o a la derecha respectivamente y los números son los grados que se girará. El parámetro mensaje corresponde a la cadena de quinientos doce caracteres máximos recibidos.

La lógica de estado siguiente está definida por la instrucción que es leída, si la instrucción corresponde a avanzar o retroceder se pasa al estado MOVERSE y en el caso de retroceder se cambia el valor de la distancia a negativo, si la instrucción es giro derecho o giro izquierdo se pasa al estado GIRAR y en el caso de giro izquierdo se cambia el valor del ángulo a negativo. Si la instrucción es el inicio o final de un ciclo se pasa al estado CICLO, si es el inicio o final del reconocimiento de obstáculos se pasa al estado OBSTACULOS y si se presiona el botón de paro de emergencia se pasa al estado de ESPERA.

El estado MOVERSE es el encargado de moverse el línea recta ya sea como avance o como retroceso. Para esto utiliza una serie de funciones, la primera de ellas corresponde a ConfiguraEscribePuenteH la cual se encarga de activar las terminales del motor correspondientes usando el signo de la distancia como referencia para determinar si se quiere avanzar o retroceder. La siguiente función es calculaDistanciaLinealRecorrida la cual realiza el cálculo de la distancia lineal recorrida por cada rueda. Esta utiliza el contador de encoders de la biblioteca ENCODERS para determinar la distancia avanzada por el robot al dividir este número entre los pulsos por milímetros para calcular la distancia en milímetros. La función ControlVelocidadRueda es usada para implementar el lazo de control PID de velocidad en una rueda. Para esta función se debe tener un muestreo en tiempos constantes, no aparecen las constantes de tiempo en la ecuación, sino que se integran con las constantes Ki y Kd. Se recibe la posición de referencia, la posición actual medida con los encoders, el error acumulado por referencia, y el valor anterior del error por referencia, y con esto se aplica la ecuación del control PID. La función entrega el valor de ciclo de trabajo (PWM) que se enviará al motor CD. Esta función está basada en el código desarrollado en la iteración inicial de Atta-Bot por lo que no se calcularon nuevamente las ecuaciones en este proyecto. Por último, la función AvansarDistanciaDeseada es aquella que recibe la distancia en centímetros y ejecuta el avance o retroceso del carro mediante el cálculo de la velocidad y la

aplicación del control de esta. Inicia con el cálculo de la velocidad actual del robot creando un ciclo con milisegundos para poder definir la cantidad de pulsos en una cantidad de tiempo y a partir de esto determinar la velocidad, después de esto, usa esta velocidad para implementar el control en cada rueda con la función correspondiente, además calcula la distancia que está siendo recorrida para determinar cuándo se ha llegado al movimiento deseado, a este punto reinicia las variables y convierte el booleano de avanceListo en verdadero.

Cabe destacar que en el robot se implementa nuevamente un doble lazo de control PID, como es el caso en el diseño original. Uno de estos lazo controla la velocidad del robot y el otro controla los giros. Ambos controles obtienen su constantes por medio de pruebas prácticas. Además, en el código se usa la biblioteca ENCODERS la cual está diseñada para contar el número de flancos positivos que genera el sensor encoder en cada motor, esta biblioteca depende de una rápida latencia por lo que se evita el uso de paros de tiempo en el código.

La lógica de estado siguiente depende de diferentes factores. Si el movimiento es completado se pasa al estado DETENERSE, si se presiona el botón de paro de emergencia se activa la variable paro_emergencia e igualmente se pasa al estado DETENERSE, lo mismo ocurre si se detecta un obstáculo por parte de cualquiera de los sensores si la acción de detección de obstáculos se encuentra activa, solo que esta vez la variable que se activa antes del cambio de estado corresponde a obstaculo_detectado.

El estado GIRAR se encarga de realizar los giros a la derecha y a la izquierda. La primera de las funciones que utiliza corresponde a ConfiguraEscribePuenteHGiros la cual activa las terminales de los motores dependiendo de la dirección del giro. ConvDistAngular corresponde a la función que se encarga de calcular la distancia angular en grados que se ha desplazado el robot, usando la cantidad de pulsos y calculando el arco que crean estos pulsos con el uso de la distancia del centro del robot a la rueda. La función ControlPosGiroRueda implementa el control PID por posición en una rueda, similar a la función de control de la velocidad esta función se basa en el diseño original del Atta-Bot por lo que las ecuaciones no fueron definidas dentro de este proyecto. EstadoEstacionario es la función que revisa el estado de la acción de control y si se mantiene varios ciclos en cero, asume que ya está en el estado estacionario. Por último, la función Giro recibe la cantidad de grados por girar y

ejecuta el giro del robot sobre su propio eje, llamando a todas las funciones anteriores para calcular la distancia angular, aplicar el control y detener el movimiento una vez se cumple con lo pedido.

La lógica de estado siguiente es igual a la de estado MOVERSE, Si el movimiento se ha completado se pasa al estado DETENERSE, si se presiona el botón de paro de emergencia se activa la variable paro_emergencia e igualmente se pasa al estado DETENERSE, lo mismo ocurre si se detecta un obstáculo por parte de cualquiera de los sensores si la acción de detección de obstáculos se encuentra activa, solo que esta vez la variable que se activa antes del cambio de estado corresponde a obstaculo_detectado.

El estado DETENERSE no utiliza ninguna función, ya que solo debe de detener los robot, por lo que desactiva todas las terminales en ambos motores. Además, se incorpora una pausa para diferenciar cada una de las instrucciones que realiza el robot. La lógica de estado siguiente es simple, si se presiona el botón de paro de emergencia se pasa al estado ESPERA, este mismo es el caso si la variable de paro_emergencia se encuentra activa. Si la variable obstaculo_detectado esta activa se pasa al estado MOVIMIENTO_OBSTACULO, en cualquier otro caso se pasa al estado LEE_MEMORIA.

El estado CICLO no usa funciones. Usa una lógica en la que, si es el inicio de las instrucciones del ciclo utiliza apuntadores para recorrer las instrucciones dentro de este, y si es el final del ciclo reinicia todas las variables, moviendo el apuntador al inicio, esto lo repite hasta que se realizan la cantidad de ciclos deseados. Su lógica de estado siguiente es simple, si se presiona el botón de paro de emergencia se pasa al estado ESPERA, en cualquier otro caso pasa al estado LEE_MEMORIA.

El estado OBSTACULOS activa la variable de detección de obstáculo o la desactiva dependiendo de la instrucción leída. Su lógica de estado siguiente es igual a la del estado CICLO, si se presiona el botón de paro de emergencia se pasa al estado ESPERA, en cualquier otro caso pasa al estado LEE_MEMORIA.

El estado MOVIMIENTO_OBSTACULO genera el movimiento después de detectar el obstáculo, se desea que en estos casos el robot retroceda y se detenga por completo con el objetivo de alejarlo del peligro por lo que se llama nuevamente a la función

AvansarDistanciaDeseada con una distancia de -5 centímetros. Después de esto se desactivan las terminales de los motores. Su lógica de estado siguiente solo tiene dos variaciones, si termina el retroceso pasa al estado ESPERA y si se presiona el botón de paro de emergencia activa la variable paro_emergencia y pasa al estado DETENERSE.

Por último, el estado NADA se encuentra vacío, este se define en caso de que ocurra un fallo en la máquina de estados y se encarga de ser un posible estado al que se puede pasar. Este tipo de estados son incorporados en todas las máquinas de este tipo. El código completo puede ser encontrado en el Apéndice 2.

Como se mencionó anteriormente los componentes como la pantalla, el sensor de color y la garra no buscan ser implementados por completo durante este proyecto debido a que este corresponde a la primer iteración del robot educativo, por esta razón su control no fue implementado dentro de la máquina de estados. Además, la aplicación que está siendo realizada por otro integrante del proyecto PROE también se encuentra en sus primeras iteraciones y no ha implementado el uso de estos componentes, por lo que aún no es claro el tipo de instrucciones que definirán su uso. Con esto se concluye la descripción del diseño y desarrollo del código.

Capítulo 5.0 Resultados y Análisis

Durante este capítulo se exponen los resultados del diseño presentado anteriormente, así como un análisis de estos al ser comparados con la métricas definidas anteriormente. También se realiza un análisis económico para comprender el verdadero valor del diseño generado.

5.1 Pruebas de validación

La estrategia de validación para el proyecto se realizó con base en la construcción de un prototipo del Atta-Bot diseñado. Este prototipo se sometió a diferentes pruebas para corroborar que cumpla con las métricas establecidas durante la etapa de especificaciones de la metodología. No todas las métricas deben ser validadas por medio de pruebas físicas, por lo que solo se tomaron en cuenta aquellas en donde aplica este tipo de criterios. En el anexo 3 se encuentran todas las medidas realizadas durante el proceso de validación, por lo que en esta sección se presentaran únicamente los resultados.

En primer lugar, se busca validar la precisión de los movimientos en línea recta realizados por el robot, por esta razón se diseñó la primera prueba de validación. Durante esta prueba se programó el robot desde la aplicación móvil para que avanzara a una distancia, después de esto se colocó sobre un pliegue de papel en donde los centros de la rueda se alinean con la línea de inicio, se oprime el botón de inicio y se deja que el robot avance, una vez finaliza el movimiento se marca y mide la distancia avanzada. Este proceso se realiza 30 veces para cada una de las distancias y se prueban cuatro niveles de distancias. En la Tabla 22 se encuentra un resumen de los resultados de la prueba. Además, en la Figura 57 se encuentra una imagen del proceso de validación. Las mediciones se encuentran en el Apéndice 3.

Tabla 22. Resultados de la validación de las distancias recorridas por el Atta-Bot.

Resultados	Niveles (mm)			
	100	200	300	900
Promedio (mm)	98.6 ± 2.6	197.9 ± 5.0	301.4 ± 3.4	900.4 ± 6.1
Moda (±1mm)	98	199	305	905
Máximo (±1mm)	104	204	308	910
Mínimo (±1mm)	91	182	295	886
Desviación estándar (mm)	2.6	5.0	3.4	6.1
Porcentaje de error (%)	-1.4	-1.1	0.5	0.04

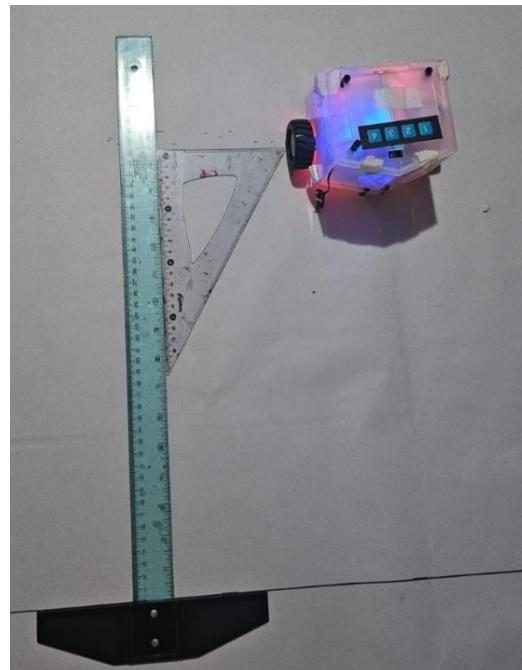


Figura 57. Imagen del proceso de validación de las distancias recorridas por el Atta-Bot.

Como un auxiliar para el análisis, se desarrolló un diagrama de cajas de bigotes para cada uno de los niveles, estos pueden ser encontrados en la Figura 58, la Figura 59, la Figura 60 y la Figura 61.

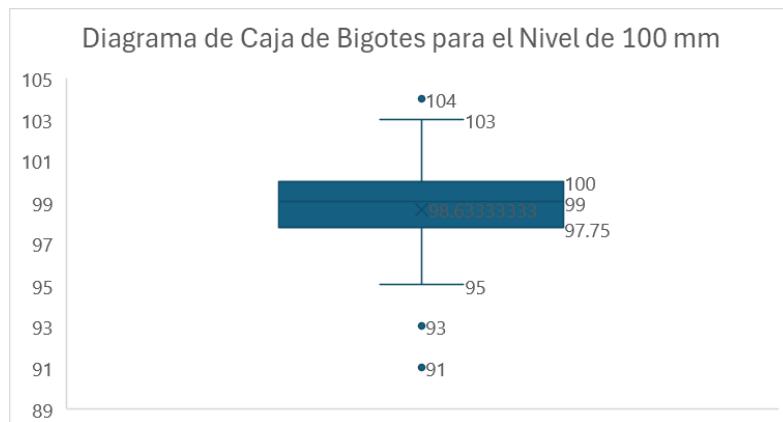


Figura 58. Diagrama de Caja de Bigotes para el nivel de 100mm

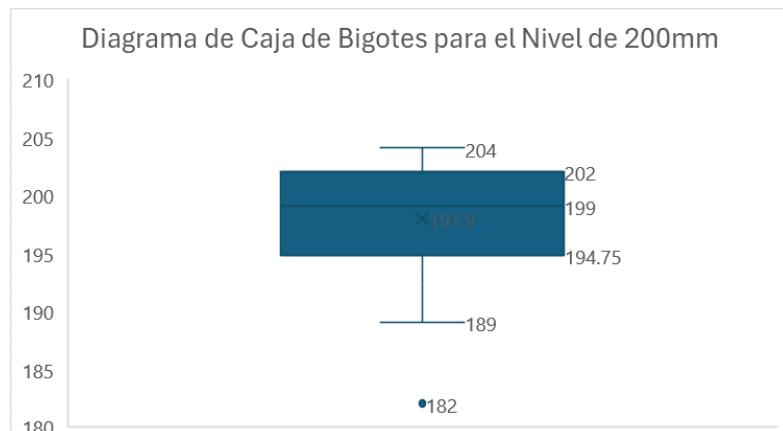


Figura 59. Diagrama de Caja de Bigotes para el nivel de 200mm

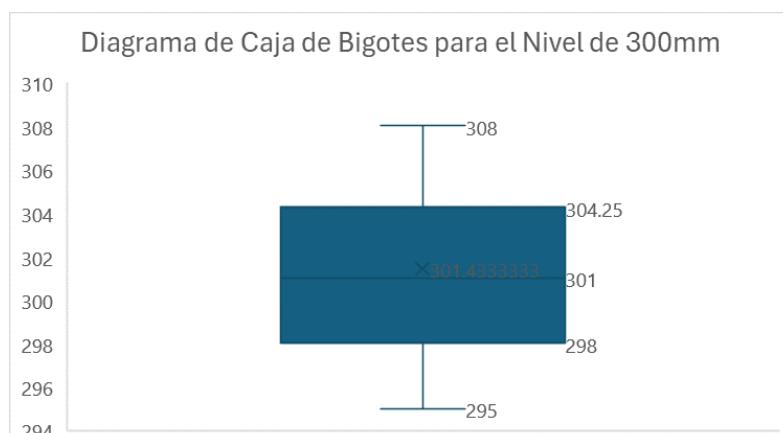


Figura 60. Diagrama de Caja de Bigotes para el nivel de 300mm

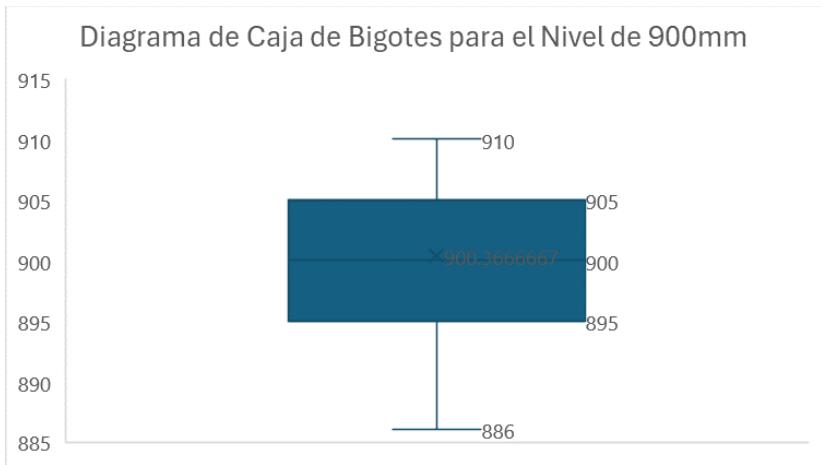


Figura 61. Diagrama de Caja de Bigotes para el nivel de 900mm

Como se puede observar en los promedios de las mediciones, los resultados obtenidos se encuentran dentro del rango de valores ideales definidos durante la etapa de las especificaciones, en donde se aclaró que las distancias pueden variar de 10 a 20 milímetros de la distancia deseada. Además de esto, en la Tabla 22 se presentan otros valores de interés, por ejemplo, si se analizan los máximos y mínimos de cada nivel se aprecia que ninguno de estos excedió los valores ideales de precisión mencionados anteriormente. Por otro lado, ninguno de los porcentajes de error excede el 2%. La desviación estándar se encuentra en un rango de 2.60mm a 5.01mm según lo que se tiene una alta precisión y mejora conforme aumenta la distancia. Por estas razones se puede concluir que el robot cumple con la métrica de precisión del movimiento.

La segunda prueba realizada busca validar la precisión de los giros. Durante esta prueba se programa el robot para que gire una cantidad de grados exacta, variando entre giros a la izquierda y giros a la derecha. Se coloca nuevamente en un pliegue de papel con la línea de inicio y el eje central del robot marcados y se inicia el cumplimiento de instrucciones con el botón. Una vez que el robot finaliza su giro se marca el papel y se mide el ángulo. Este proceso es realizado 30 veces para cada ángulo y se prueba el robot con 6 diferentes niveles de ángulos que abarcan el espectro de uso del robot. El resumen de los resultados de esta prueba se puede observar en la Tabla 23. Además de esto la Figura 62 muestra un ejemplo del proceso de desarrollo de la prueba. También se realizaron diagramas de caja de bigotes para cada nivel, los cuales pueden ser encontrados en la Figura 63, la Figura 64, la Figura 65, la Figura 66, la Figura 67 y la Figura 68.

Tabla 23. Resultados de la validación de los giros realizados por el Atta-Bot.

Resultados	Niveles (°)					
	20	45	90	120	180	240
Promedio (°)	19.2 ± 1.7	44.7 ± 2.3	89.9 ± 2.4	119.1 ± 2.3	179.4 ± 2.9	238.9 ± 2.3
Moda (±0.5°)	20.0	44.0	91.0	120.0	178.0	237.0
Máximo (±0.5°)	23.0	50.0	94.0	125.0	185.0	243.0
Mínimo (±0.5°)	16.0	40.0	85.0	115.0	173.0	235.0
Porcentaje de error (%)	-3.8	-0.7	-0.1	-0.8	-0.4	-0.5
Desviación estándar (°)	1.7	2.3	2.4	2.3	2.9	2.3



Figura 62. Imagen del proceso de validación de los giros realizados por el Atta-Bot.

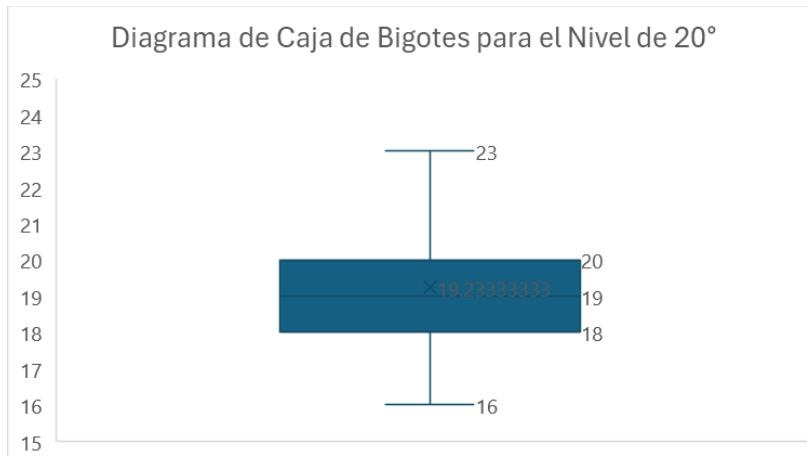


Figura 63. Diagrama de Caja de Bigotes para el nivel de 20°

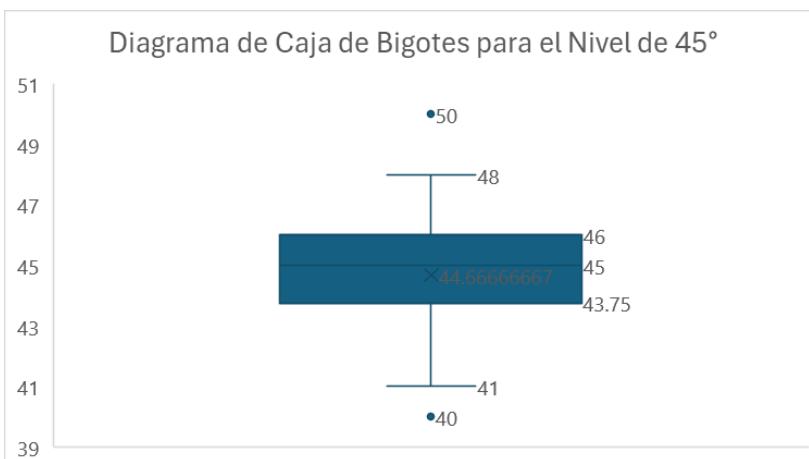


Figura 64. Diagrama de Caja de Bigotes para el nivel de 45°

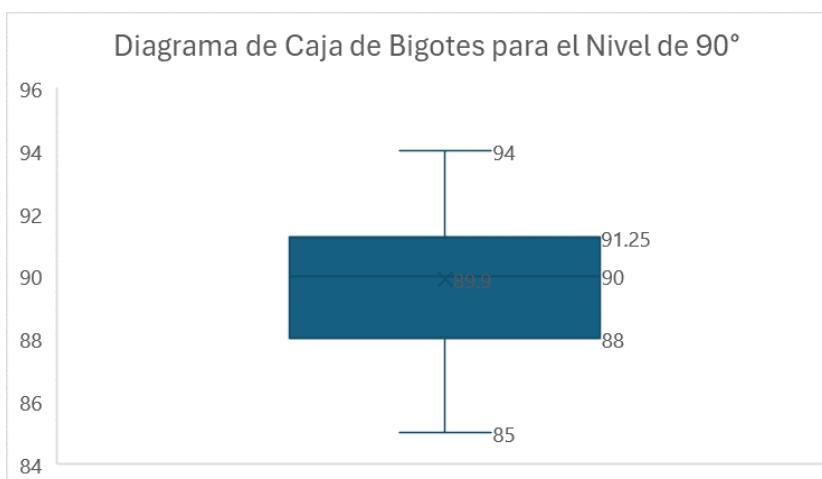


Figura 65. Diagrama de Caja de Bigotes para el nivel de 90°

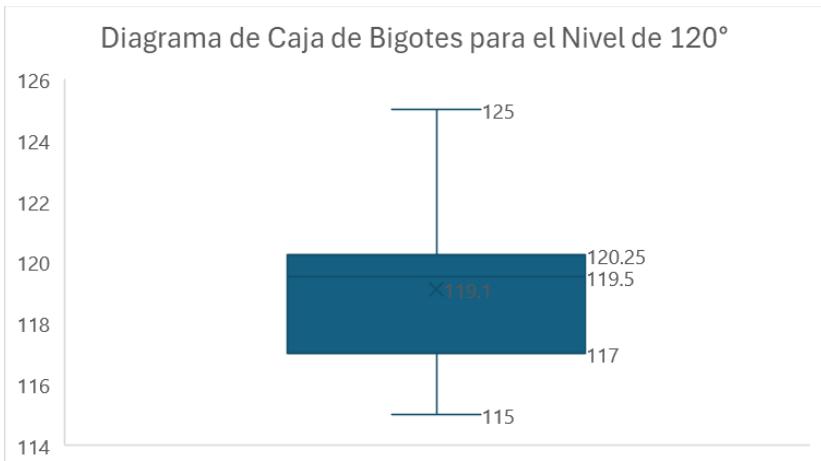


Figura 66. Diagrama de Caja de Bigotes para el nivel de 120°

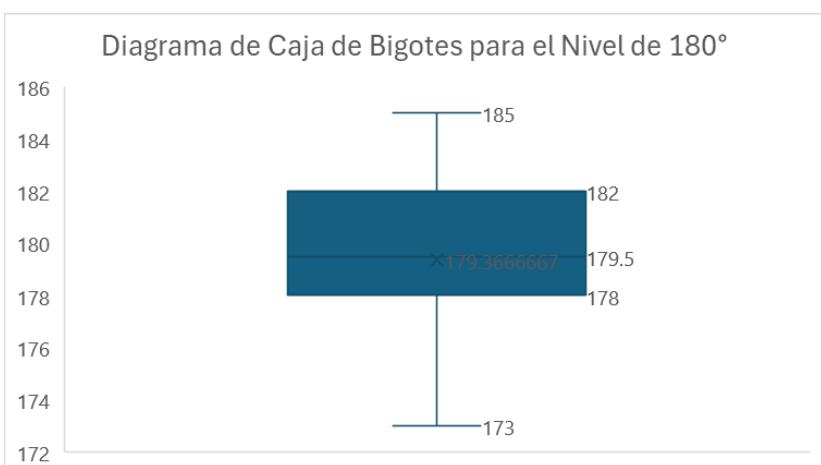


Figura 67. Diagrama de Caja de Bigotes para el nivel de 180°

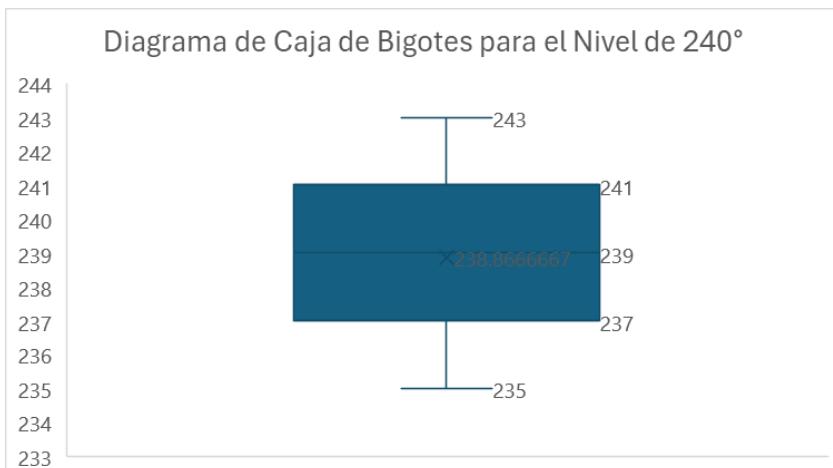


Figura 68. Diagrama de Caja de Bigotes para el nivel de 240°

Al observar los resultados, los promedios indican que todos los niveles de prueba cumplen con los valores ideales que fueron definidos en las especificaciones, que permiten que el giro varie entre 10° y 15° del valor programado. Además, en esta misma tabla se muestra que los valores mínimos y máximos también se encuentran dentro del rango establecido de fallo permitido. Cabe destacar que los porcentajes de error son aún menores en esta prueba, ninguno de estos supera el 4% pero la gran mayoría se mantiene incluso bajo el 0.5%. En este caso se tiene un mayor porcentaje de error entre menor es el ángulo, ya que todos los niveles varían una cantidad similar de grados, pero estos tienen un mayor efecto en los porcentajes de error de ángulos más pequeños. La desviación estándar varía de 1.72° a 2.96° lo cual corresponde a una alta precisión. Por ende, se puede concluir que el robot ha sido validado con respecto a la precisión de sus giros.

La siguiente prueba que se diseñó busca validar la capacidad del robot de detectar obstáculos, para esta prueba se programó que el robot avanzara 20 centímetros y que estuviera activa la función de detención de obstáculos, después de esto se colocó sobre un pliego de papel y se colocaron obstáculos de diferentes formas y tamaños frente al robot. Una vez el robot detectara el obstáculo y se detuviera se creaba una marca en este punto del papel en el punto del robot más cercano al obstáculo y se midió la distancia entre este punto y el obstáculo detectado, esta prueba se realizó 30 veces y sus resultados se encuentran en Tabla 24.

Tabla 24. Resultados de la validación de la detención de obstáculos del Atta-Bot.

Resultados	Distancia(mm)
Promedio ($\pm 17.6\text{mm}$)	73.4
Moda ($\pm 1\text{mm}$)	87
Máximo ($\pm 1\text{mm}$)	101
Mínimo ($\pm 1\text{mm}$)	40
Desviación estándar(mm)	17.6

Como se puede observar, la menor distancia a la que se detuvo el robot es de 40 milímetros, la cual está dentro del rango de valores ideales de 4 o más centímetros establecidos en las métricas de las especificaciones. Esto significa que en su peor rendimiento se adecuó al valor deseado. Su moda corresponde a 87 milímetros la cual es una distancia en la que se asegura

que el robot estará seguro de cualquier daño que puede ser ocasionado por los obstáculos. Es de igual importancia destacar que además el robot nunca sufrió una colisión con ninguno de los obstáculos durante la prueba, por lo que se puede deducir que detectó el 100% de los obstáculos.

La siguiente prueba que se diseñó y aplicó busca validar el tiempo de autonomía del robot. En la etapa de necesidades se definió como valor ideal 90 minutos, ya que este corresponde a la duración de dos lecciones completas para estudiantes de colegio. Para la prueba se cargó la batería del robot al máximo y se creó un ciclo de 999 repeticiones de diferentes distancias de avances y giros, después de esto se colocó el robot en una zona segura y se inició la ejecución de las instrucciones al presionar el botón. En este mismo momento se inició un cronómetro para medir el tiempo de autonomía. Esta prueba fue realizada una única vez, debido a que esta tuvo una duración de aproximadamente 15 horas para poder ser aplicada, lo que incluye 4 horas de carga de batería y 11 horas de prueba. El tiempo final de autonomía del robot que se obtuvo en la prueba es de 9 horas 51 minutos y 3 segundos. Cabe destacar que el robot se mantuvo en contacto con el piso todo el tiempo para que la potencia consumida por los motores fuera de acuerdo con el torque necesario para mover su peso, es decir ninguna parte de la prueba fue realizada en condiciones no realistas del uso del robot educativo. Además, el robot encendió su luz de baja batería a las 9 horas y 10 minutos. Estos datos permiten validar los valores ideales de tiempo de autonomía del robot.

La última prueba que se diseñó y aplicó en el robot valida su facilidad de carga, para esta prueba se midió el tiempo que toma conectar el circuito de carga del PowerBoost a un puerto USB-mini, se tomaron 30 medidas del tiempo y la acción fue realizada por 4 diferentes personas, los resultados se encuentran en la Tabla 25.

Tabla 25. Resultados de la validación de la facilidad de carga del Atta-Bot.

Medidas	Tiempo (s)
Promedio ($\pm 9.8s$)	42.7
Moda ($\pm 0.5s$)	45.0
Máximo ($\pm 0.5s$)	60.0
Mínimo ($\pm 0.5s$)	21.0
Desviación estándar	9.8

Como se puede observar el promedio del tiempo se encuentra dentro de los valores ideales definidos anteriormente, que corresponden de 60 a 30 segundos. El mínimo y máximo de los resultados también cumplen con esta especificación, por lo que se considera que se validó el robot en este aspecto.

Con esto concluyen las pruebas realizadas para validar las métricas establecidas. Aparte de esto cabe destacar que todas las métricas que no fueron probadas de estas formas fueron cumplidas según el diseño ya sea en sus valores marginales o ideales

5.2 Análisis económico

El análisis económico que se desarrolló para el proyecto corresponde a la creación de una lista de todos los factores económicos que forman parte del rediseño del Atta-Bot. En primer lugar, se tiene la lista de los componentes y sus precios. Después de esto se encuentra un desglose del salario atribuido al diseñador a lo largo del proyecto y por último se desglosa el costo de todos los productos que fueron utilizados para el desarrollo del proyecto como softwares, computadores, entre otros. Toda esta información fue recopilada en la Tabla 26.

Tabla 26. Análisis económico del proyecto.

Componentes	
Nombre	Costo en CR
Motor cd con encoder	\$29.9
Batería Lipo	\$17.95
Boost Batery mánager	\$10.95
Soporte Motor	\$1.25
Ball Caster	\$2.75
IdeaBoard	\$39.95
Servomotor	\$6.95
Infrarrojo delantero	\$8.85
Infrarrojo abajo	\$3.90
Sensor de color	\$11.38

Pantalla		\$5.95	
Interruptor general		\$1.25	
Keypad		\$1.95	
Kit conectores		\$8.95	
Kits espaciadores		\$15.95	
Cable		\$3.95	
STEMMA QT cable		\$1.49	
STEMMA QT cable macho-macho		\$1.95	
Total		\$175.27	
Salario del investigador			
Salario por hora	Número de horas trabajadas a la semana	Numero de semana del proyecto	Total
₡1200	20	16	₡ 384000 ≈ \$750
Activos utilizados			
Activo	Valor monetario asociado		
Software Lucid charts	\$7.95		
Software Canva	\$5		
Computadora portátil	\$486.083		
Total	\$499.033		
Total			
Componentes	\$175.27		
Salario del investigador	\$750		
Activos utilizados	\$499.033		
Total	\$1424.303		

Como se puede observar, se estima que el costo total del proyecto se acerca a los \$1424.303. Este corresponde a un análisis económico detallado, en el cual se toman en cuenta gastos únicos, como el costo del salario para el diseñador, las suscripciones de los softwares usados para el diseño y la computadora en la que se desarrolló el código.

Si se toma un enfoque centrado en el costo de la fabricación de un solo robot, tomando en cuenta los componentes necesarios únicamente y no la mano de obra, es donde se puede apreciar el verdadero impacto económico del proyecto. El costo de los componentes del robot se suma a un total de \$175.25, lo cual es menor al costo actual de los componentes del Atta-Bot original de \$216.18 y también es menor al costo del Robot Thymio y Lego Mindstorms. Además, este es el precio de los componentes dentro del país, por lo que se debe de tomar en cuenta que otros robots educativos deben de ser importados, lo que incrementa el precio final, por lo que se puede asumir que será también más económico que el robot Zowi e incluso Bee-Bot.

Por otro lado, el precio de la mano de obra no está siendo tomado en cuenta debido a que el objetivo principal de este robot es su uso educativo, y su construcción es parte de esto. Por esta razón en el caso del Atta-Bot diseñado en este proyecto, no existe un costo de manufactura, a diferencia del robot original. Esto se debe a que el robot será suplementado en partes para ser construido en las aulas.

6.0 Conclusiones y Recomendaciones

En este último capítulo se buscar cerrar el proyecto generando una lista de conclusiones y recomendaciones considerando los resultados y el proceso de diseño desarrollado anteriormente.

6.1 Conclusiones

- El diagnóstico realizado del Atta-Bot de enjambres destacó nueve ventajas que comprobaron su capacidad para ser rediseñado exitosamente como un robot educativo.
- La arquitectura electrónica diseñada que cumple en un 100% los valores marginales de las especificaciones definidas y un 90% los valores ideales.
- El código diseñado cumple el 100% de la tareas establecidas como necesidades del robot y permite la precisión de 1.72° a 2.96° en los giros y de 2.60mm a 5.01mm en el movimiento en línea recta
- El prototipo del Atta-Bot educacional construido validó de forma exitosa el diseño del robot en términos de locomoción, detección de obstáculos, autonomía y facilidad de uso.

6.2 Recomendaciones

- Una ampliación del sistema de detención de obstáculos que sea útil para los movimientos en reversa del Atta-Bot.
- La aplicación de un tercer lazo de control PID para controlar la distancia recorrida por el Atta-Bot, similar al usado en el control del giro, para obtener movimientos aún más precisos.
- La incorporación de un código que controle la garra, la pantalla LCD y el sensor de luz y color en la máquina de estados para su uso en ambientes educativos.

Referencias bibliográficas

- [1] S. K. Saha, Introducción a la Robótica, Mexico D.F.: Mc Graw Hill Educación, 2010.
- [2] E. Ruiz-Velasco, Robotica Pedagógica, México D.F: Somece, 1998.
- [3] A. Barrientos, F. L. Peñín, C. Balaguer y R. Aracil, Fundamentos de Robótica, Madrid: McGraw Hill , 1997.
- [4] M. E. Moran, «Evolution of robotic arms,» *J Robotic Surg*, vol. 1, pp. 103-111, 2007.
- [5] M. Köseoğlu, O. M. Çelik y Ö. Pektaş, «Design of an autonomous mobile robot based on ROS,» *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, pp. 1-5, 2017.
- [6] R. Aracil, C. Balager y M. Armada, «Robots de servicio,» *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 5, nº 2, pp. 6-13, 2008.
- [7] J. Cornejo, J. A. Cornejo-Aguilar y J. P. Perales-Villarroel, «Innovaciones internacionales en robótica médica para mejorar el manejo del paciente en Perú,» *Revista de la Facultad de Medicina Humana*, vol. 19, nº 4, pp. 105-113, 2019.
- [8] Y. Archimowics y S. Martinez, «EdRo - Un enjambre de robots,» UR. FING, 2011.
- [9] J. M. Garcia, «Robótica Educativa . ¿Modelo para armar?,» *Virtualidad, Educacion y Ciencia*, vol. 6, nº 10, pp. 77-90, 2015.
- [10] E. Afari y M. S. Khine, «Robotics as an Educational Tool: Impact of Lego,» *International Journal of Information and Education Technology*, vol. 7, nº 6, pp. 437-442, 2017.
- [11] B. Ortega-Ruipérez, «Elección de simuladores de robótica en la formación de docentes según su etapa educativa,» Octaedro, 2021.

- [12] Sphero, «Sphero mini,» [En línea]. Available: <https://sphero.com/products/sphero-mini>. [Último acceso: 15 Febrero 2024].
- [13] C. Tecles del Romero, «Speech recognition with the Zowi robot,» Valencia, 2019.
- [14] J. Penalva, «Probamos Zowi, un robot con cerebro Arduino que puede dar más de lo que aparenta,» xacata, 2 Noviembre 2019. [En línea]. Available: <https://www.xataka.com/analisis/probamos-zowi-un-robot-con-cerebro-arduino-que-puede-dar-mas-de-lo-que-aparenta>. [Último acceso: 15 Febrero 2024].
- [15] Lego, «Lego Mindstorms,» Lego, [En línea]. Available: <https://www.lego.com/es-es/product/lego-mindstorms-ev3-31313>. [Último acceso: 16 Febrero 2024].
- [16] D. Arnau Vera, P. D. Diago Nebot y J. A. Gonzalez-Caler Somoza, «Elementos de resolución de problemas en primeras edades escolares con Bee-bot,» *Edma 0-6: Educacion Matemática en la Infancia*, vol. 7, nº 1, pp. 12-41, 2018.
- [17] F. Mondana, M. Bonani, R. Fanny, M. Briod, L. Pereye, P. Rétomaz y S. Magnenat, «Bringing Robotics to Formal Education,» *IEEE Robotics and Atumation Magezine*, pp. 77-85, 2017.
- [18] M. Jimenez Castro y R. J. Cerdas González, «La robótica educativa como agente promotor del estudio por la ciencia y la tecnología en la región atlántica de Costa Rica,» de *Congreso Iberoamericano de Ciencia, Tecnología, Innovacion y Educación*, Buenos Aires, Argentina, 2014.
- [19] A. L. Acuña Zuñiga, «Diseño y Administración de los Proyectos de Robotica Educativa: Lecciones Aprendidas,» *Teoría de la Educación. Educación y Cultura en la Sociedad de la Información*, vol. 13, nº 3, pp. 6-27, 2012.
- [20] K. Ulrich y S. Eppinger, Diseño y desarrollo de productos, México D.F.: Mc Graw Hill Education , 2004.
- [21] J. C. Brenes-Torres, R. Solis-Ortega y C. Calderón-Arce, «Atta-Bots: plataforma de bajo costo y facil acceso para robotica de enjambres,» Cartago, 2023.

- [22] J. Romero-Perigault, N. Rodríguez, D. Rodríguez, L. Custodio, D. Hernadez y F. Merchan, «Tratamiento digital de un pulso ultrasónico para la navegación de un robot autónomo móvil mediante análisis geométrico,» *Revista de Iniciación Científica*, vol. 6, 2021.
- [23] Kingston Tecnology, «Explicación de USB-C,» Kingston Tecnology, Diciembre 2020. [En línea]. Available: <https://www.kingston.com/es/blog/mobile-lifestyle/usb-c-explained>. [Último acceso: 13 Marzo 2024].
- [24] M. González, «Zowi, la apuesta de BQ por los juguetes inteligentes y programables,» Xataka, 14 Octubre 2015. [En línea]. Available: <https://www.xataka.com/robotica-e-ia/zowi-la-apuesta-de-bq-por-los-juguetes-inteligentes-y-programables>. [Último acceso: 28 Marzo 2024].
- [25] J. Heinerman, M. Rango y A. Eiben, «Evolution, Individual Learning, and Social Learning in a Swarm of Real Robots.,» 2015.
- [26] L. España, «Sensor Infrarrojo EV3,» LRobotica, [En línea]. Available: https://www.esmindstorms.com/sensor-infrarrojo-ev3/#google_vignette. [Último acceso: 20 Marzo 2024].
- [27] M. Nava, J. Guzzi, R. Chavez-Garcia, L. M. Gambardella y A. Giusti, «Demo: Learning to Perceive Long-Range Obstacles Using Self-Supervision from Short-Range Sensors.,» de *AAAI Conference on Artificial Intelligence.*, 2019.
- [28] RO-BOTICA, «BEE-BOT® Robot infantil programable,» RO-BOTICA, 2021. [En línea].
- [29] Z. Zhong, X. Lu, D. Wu, J. Weng, M. Wei, C. Deng, J. Fang, S. Wen y P. Huang, «Multifunctional education robot». China Patente CN210189807, 27 Marzo 2020.
- [30] S. Su, «Education Robot». China Patente CN209986949, 24 Enero 2020.
- [31] Y. Sun y T. Li, «Robot for children education». China Patente CN210361316, 21 Abril 2020.

- [32] Y.-S. Choi y M.-Y. Lee, «Educational robot apparatus for children and method of operating the same.». United States of America Patente US20110129804, 2 Junio 2011.
- [33] CRCibernetica, «Productos,» [En línea]. Available: <https://www.crcibernetica.com>. [Último acceso: 27 2 2024].
- [34] *Materiales para la construcción de 1 agente robot de PROE*, Cartago, 2021.
- [35] Europa Press Portaltic, «Las exigentes pruebas a las que se someten los 'smartphones': agua, caídas o temperaturas extremas.,» Europa Press, 19 Septiembre 2021. [En línea]. Available: <https://www.europapress.es/portaltic/gadgets/noticia-exigentes-pruebas-someten-smartphones-agua-caidas-temperaturas-extremas-20210919103036.html>. [Último acceso: 13 Marzo 2024].
- [36] ELEGOO, «Elegoo UNO Kit,» Amazon , 1 Mayo 2019. [En línea]. Available: https://www.amazon.com/inteligente-seguimiento-ultras%C3%B3nico-educativo-adolescentes/dp/B07KPZ8RSZ/ref=sr_1_1_sspa?dib=eyJ2IjoiMSJ9.872K7-m4rOqTfu-9HaXITZy8qWIkmR48-Yd8cYraVUCTRXoiyMeB6srwT5vlkt_wcZTXjRktnS9c2cnYefTi-OduwteGwgmlX6cGIq5l5coGvCaRpY46SQZh8.

Apéndices

Apéndice 1. BOM del Atta-Bot Educativo

Materiales para la construcción de 1 agente de Atta-Bot educativo

Cantidad	Nombre	# de parte	Link	Costo CR /u
2	Motor cd con encoder	ROB-CB0N20	https://www.crcibernetica.com/n20-gearmotor-with-encoder-and-wheel/	\$14.95
1	Batería Lipo	ADA-328	https://www.crcibernetica.com/lithium-ion-polymer-battery-3-7v-2500mah/	\$17.95
1	Boost Batery mánager	ADA-2465	https://www.crcibernetica.com/powerboost-1000-charger-rechargeable-5v-lipo-usb-boost-1a-1000c/	\$10.95
2	Soporte Motor	ROB-CB02443	https://www.crcibernetica.com/micro-gearmotor-support-bracket/	\$0.75
1	Ball Caster	ROB-CB1818	https://www.crcibernetica.com/3pi-n20-ball-caster/	\$2.75
1	IdeaBoard	ROB-CBIB01	https://www.crcibernetica.com/crcibernetica-ideaboard/	\$39.95
1	Servomotor	MG90S	https://www.crcibernetica.com/mg90s-micro-servo-with-metal-gears/	\$6.95
3	Infrarrojo delantero	SEN-CB0006	https://www.crcibernetica.com/infrared-obstacle-avoidance-module/	\$2.95
2	Infrarrojo abajo	SEN-CB0005	https://www.crcibernetica.com/track-sensor-module/	\$1.95
1	Sensor de color	ADA-3595	https://www.crcibernetica.com/apds9960-proximity-light-rgb-and-gesture-sensor-stemma-qt-qwiic/	\$11.38
1	Pantalla	LCD-CB4557	https://www.crcibernetica.com/full-color-lcd-display-module-0-96inch/	\$5.95
1	Interruptor general	COM-CB0372	https://www.crcibernetica.com/slide-switch/	\$1.25
1	Keypad	COM-CB1333	https://www.crcibernetica.com/4-button-membrane-keypad/	\$1.95
1	Kit conectores	KIT-CB0012	https://www.crcibernetica.com/dupont-connector-assorted-kit/	\$8.95
1	Botón rojo	COM-CB1038RED	https://www.crcibernetica.com/30mm-arcade-style-push-button/	\$1.95
1	Kits espaciadores	TOL-CB0513	https://www.crcibernetica.com/m3-nylon-screw-black-hex-screw-nut-nylon-pcb-standoff-assortment-kit-250pcs/	\$15.95
1	Cable	CAB-CB5215-4:4	https://www.crcibernetica.com/u12464-wiring-cable-1m/	\$3.95
1	STEMMA QT cable	ADA-4397	https://www.crcibernetica.com/stemma-qt-qwiic-jst-sh-4-pin-cable-with-premium-female-sockets-150mm-long/	\$1.49
1	STEMMA QT cable	CAB-CB4925	https://www.crcibernetica.com/stemma-qt-qwiic-jst-sh-4-pin-cable-20cm/	\$1.95

Apéndice 2. Código completo

```
#define ENCODER_DO_NOT_USE_INTERRUPTS

#include <ArduinoBLE.h>

#include <string>

#include <Encoder.h>

using namespace std;

//constantes del robot empleado

const int tiempoMuestreo=100000; //unidades: micro segundos

const float pulsosPorRev=45; //cantidad de pulsos de una única salida

const int factorEncoder=4; //cantidad de salidas de encoders que se están detectando
(pueden ser 2 o 4)

const float circunferenciaRueda=139.5;//Circunferencia de la rueda = 139.5mm

const float pulsosPorMilimetro=((float)pulsosPorRev)/circunferenciaRueda;

const float distanciaCentroARueda=61;// Radio de giro del carro, es la distancia en mm
entre el centro y una rueda.

const float conversionMicroSaMin=1/(60 * 1000000); // factor de conversion microsegundo
(unidades del tiempo muestreo) a minuto

const float conversionMicroSaSDiv=1000000; // factor de conversion microsegundo
(unidades del tiempo muestreo) a segundo

const float tiempoMuestreoS= (float)tiempoMuestreo/conversionMicroSaSDiv;

//constantes para control PID movimiento lineal para tiempoMuestreo=100000 us

const float velRequerida=120.0; //unidades mm/s
```

```

const float KpVel=1; //constante control proporcional
const float KiVel=0.65; //constante control integral
const float KdVel=0.0; //constante control derivativo

//constantes para control PID de giro
const float KpGiro=4.0; //constante control proporcional
const float KiGiro=1.1;//constante control integral
const float KdGiro=0.17; //constante control derivativo

//Constantes para la implementación del control PID real
const int errorMinIntegralVelocidad=-255;
const int errorMaxIntegralVelocidad=255;
const int errorMinIntegralGiro=-110;
const int errorMaxIntegralGiro=110;
const int limiteSuperiorCicloTrabajoVelocidad=200;
const int limiteInferiorCicloTrabajoVelocidad=90;
int limiteSuperiorCicloTrabajoGiro=150;
int limiteInferiorCicloTrabajoGiro=0;
const int limiteSuperiorCicloTrabajoGiroCalibracion=50;
const int limiteInferiorCicloTrabajoGiroCalibracion=-50;
const int cicloTrabajoMinimo= 20;
const int minCiclosEstacionario= 20;
const float correccionVelRuedas = 0.5;

```

```
// Muestreo velocidad  
  
unsigned long previousMillis=0;  
  
long intervaloTiempoMuestreo=100;  
  
long contPulsosPasadoIzq=0;  
  
long contPulsosPasadoDer=0;  
  
long contPulsosIzq=0;  
  
long contPulsosDer=0;
```

//Variables que almacenan el desplazamiento angular de cada rueda

```
float posActualRuedaDerecha = 0.0;  
  
float posActualRuedaIzquierda = 0.0;
```

//Variables del valor de velocidad en cada rueda

```
float velActualDerecha = 0.0;  
  
float velActualIzquierda = 0.0;
```

//Valores acumulados para uso en las ecuaciones de control

//Para Giro

```
float errorAnteriorGiroDer = 0;  
  
float errorAnteriorGiroIzq = 0;  
  
float sumErrorGiroDer = 0;  
  
float sumErrorGiroIzq = 0;
```

```
int contCiclosEstacionarioGiro = 0;
```

```
//Para Velocidad
```

```
float errorAnteriorVelDer = 0;
```

```
float errorAnteriorVelIzq = 0;
```

```
float sumErrorVelDer = 0;
```

```
float sumErrorVelIzq = 0;
```

```
//Para Coordenadas
```

```
float distLinealRuedaDerecha = 0;
```

```
float distLinealRuedaIzquierda = 0;
```

```
//Variables para avance
```

```
int distanciaAvanzada = 0;
```

```
bool giroTerminado = 1; //Se hace esta variable global para saber cuando se está en un giro  
y cuando no
```

```
// variables para los pines de la ideaboard
```

```
long contadorPulsosIzquierda = 0;
```

```
long contadorPulsosDerecha = 0;
```

```
const int motorIzquierdoA = 12;
```

```
const int motorIzquierdoB = 14;  
const int motorDerechoA = 13;  
const int motorDerechoB = 15;  
const int sensorInfrarrojoFrontal = 34;  
const int sensorInfrarrojoDerecho = 39;  
const int sensorInfrarrojoIzquierdo = 4;  
const int sensorTrackerDerecho=5;  
const int sensorTrackerIzquierdo=36;  
const int pin_boton_Start=19;  
const int pin_boton_Stop=18;
```

```
//Iniciados de contador de encoders  
Encoder encoderIzquierdo(27, 33);  
Encoder encoderDerecho(32, 35);
```

```
//variables máquina de estados  
enum posibles_Estados {ESPERA=0, LEE_MEMORIA, MOVERSE, GIRAR,  
DETENERSE, CICLO, OBSTACULOS, MOVIMIENTO_OBSTACULO, NADA};  
posibles_Estados estado = ESPERA;  
bool giro_listo = false;  
bool movimiento_listo = true;
```

```

bool obstaculos_activo = false;

bool primer_ciclo = true;

bool retroceso_listo = true;

bool obstaculo_detectado=false;

bool paro_emergencia=false;

//variables para comunicación Bluetooth

string mensajeBLE="ATINIOBINICI999AV200GD090CIFINOBFINATFIN";

//string mensajeBLE="";

BLEService servicio("4fafc201-1fb5-459e-8fcc-c5c9c331914b");

BLEStringCharacteristic caracteristico("beb5483e-36e1-4688-b7f5-ea07361b26a8",
BLERead | BLEWrite, 512);

//apuntadores y variables para la lista de instrucciones

enum posibles_Instrucciones {inst_Avanzar=1, inst_Retroceder, inst_GiroIzquierdo,
inst_GiroDerecho, inst_CicloInicia, inst_CicloFin, inst_ObstaculoInicia,
inst_ObstaculoFin};

short lista_instrucciones[100][2];

short inst_final = 0;

short inst_actual = 0;

short instrucion = 0;

short valor_instrucion = 0;

short inst_inicio_ciclo = 0;

short cantidad_ciclos = 0;

```

```
void setup() {  
  
    pinMode(motorIzquierdoA, OUTPUT);  
  
    pinMode(motorDerechoA, OUTPUT);  
  
    pinMode(motorIzquierdoB, OUTPUT);  
  
    pinMode(motorDerechoB, OUTPUT);  
  
    pinMode(sensorInfrarrojoFrontal,INPUT);  
  
    pinMode(sensorInfrarrojoDerecho,INPUT);  
  
    pinMode(sensorInfrarrojoIzquierdo,INPUT);  
  
    pinMode(sensorTrackerDerecho,INPUT);  
  
    pinMode(sensorTrackerIzquierdo,INPUT);  
  
    pinMode (pin_boton_Start, INPUT_PULLUP);  
  
    pinMode(pin_boton_Stop,INPUT_PULLUP);  
  
    encoderIzquierdo.write(0);  
  
    encoderDerecho.write(0);  
  
    Serial.begin(115200);
```

//Inicialización comunicación bluetooth BLE

```
if (!BLE.begin()) {  
  
    Serial.println("starting BLE failed!");  
  
    while (1);
```

```
    }

BLE.setLocalName("MyBLE");

BLE.setAdvertisedService(servicio);

servicio.addCharacteristic(caracteristico);

BLE.addService(servicio);

BLE.advertise();
```

```
}
```

```
void loop() {

BLEDevice central = BLE.central();

contadorPulsosIzquierda = encoderIzquierdo.read();

contadorPulsosDerecha = encoderDerecho.read();

Serial.print("motor 1:");

int lecturaBotonStart=digitalRead( pin_boton_Start );

int lecturaBotonStop=digitalRead( pin_boton_Stop );

int lecturaInfrarrojoFrontal=digitalRead(sensorInfrarrojoFrontal);

int lecturaInfrarrojoDerecho=digitalRead(sensorInfrarrojoDerecho);

int lecturaInfrarrojoIzquierdo=digitalRead(sensorInfrarrojoIzquierdo);
```

```
//Máquina de estados principal
```

```
switch (estado) {
```

```

case ESPERA: {

    //Serial.print("Stop: ");

    //Serial.println(digitalRead( pin_boton_Stop ) );

    //Serial.println("Estado: Espera (recibe mensaje BLE)");

    //delay(1000);

    //Leer comunicación por BLE

    if (central) {

        //Serial.print("Conectado al central: ");

        //Serial.println(central.address());

        if (central.connected()) { //era un while

            if (caracteristico.written()) {

                //Serial.print("Recibido: ");

                mensajeBLE = string( caracteristico.value().c_str() );

                //Serial.println(mensajeBLE.c_str());

                caracteristico.writeValue("");

            }

        }

        //Lógica de estado siguiente

        if (!lecturaBotonStart) {

```

```

//Serial.println("boton");

Interpreta_mensajeBLE(mensajeBLE);

//Serial.println(mensajeBLE.c_str());

delay (1000);

estado = LEE_MEMORIA;

inst_actual = 0;

}

break;

}

case LEE_MEMORIA: {

//Serial.println("lee memoria");

if ( inst_actual != inst_final ){

instrucion = lista_instrucciones[inst_actual][0];

//Serial.print(instrucion);

valor_instrucion = lista_instrucciones[inst_actual][1];

inst_actual++;

} else {

inst_actual = 0;

}

//Lógica estado siguiente

```

```

if (!lecturaBotonStop || inst_actual == inst_final) {

    estado = ESPERA;

}else if (instrucion == inst_Avanzar) {

    estado = MOVERSE;

}else if (instrucion == inst_Retroceder) {

    estado = MOVERSE;

    valor_instrucion = valor_instrucion * -1;

}else if (instrucion == inst_GiroIzquierdo) {

    estado = GIRAR;

    valor_instrucion = valor_instrucion * -1;

}else if (instrucion == inst_GiroDerecho) {

    estado = GIRAR;

}else if (instrucion == inst_CicloInicia || instrucion == inst_CicloFin) {

    estado = CICLO;

}else if (instrucion == inst_ObstaculoInicia || instrucion == inst_ObstaculoFin) {

    estado = OBSTACULOS;

}

```

```

//Serial.print("Estado: Lee Memoria -> ");

//Serial.print(posibles_Instrucciones(instrucion));

//Serial.print(" valor: ");

//Serial.println(valor_instrucion);

//delay(2000);

break;

}

case MOVERSE: {

//Ejecuta el estado de avanzar o retroceder

//(es el mismo pero con distancia negativa)

//Serial.println("Estado: Moverse (adelante o atrás)");

//delay(2000);

// movimiento

movimiento_listo=AvansarDistanciaDeseada(valor_instrucion,
contadorPulsosDerecha, contadorPulsosIzquierda);

//Lógica estado siguiente

if (!lecturaBotonStop) {

paro_emergencia=true;

```

```

estado = DETENERSE;

}else if ( movimiento_listo ) {

//movimiento_listo = false;

estado = DETENERSE;

}

else if (obstaculos_activo) {

if (!lecturaInfrarrojoFrontal||!lecturaInfrarrojoDerecho||!lecturaInfrarrojoIzquierdo){

obstaculo_detectado=true;

estado=DETENERSE;

}

break;

}

case GIRAR: {

//Ejecuta el estado de girar derecha o izquierda

//(es el mismo pero con ángulo negativo)

//Serial.println("Estado: Girar (derecha o izquierda)");

//delay(2000);

//giro
}

```

```

movimiento_listo=Giro(valor_instrucion);

//Lógica estado siguiente

if (!lecturaBotonStop) {

    paro_emergencia=true;

    estado = DETENERSE;

}

}else if ( movimiento_listo ) {

    //movimiento_listo = false;

    estado = DETENERSE;

}

else if (obstaculos_activo) {

    if (!lecturaInfrarrojoFrontal||!lecturaInfrarrojoDerecho||!lecturaInfrarrojoIzquierdo){

        obstaculo_detectado=true;

        estado=DETENERSE;

    }

}

break;

}

case DETENERSE: {

    //***Poner los motores en OFF
}

```

```

//Serial.println("Estado: Detenerse (siempre luego de cada movimiento)");

ConfiguraEscribePuenteH(1,0,0);
ConfiguraEscribePuenteH(-1,0,0);
ConfiguraEscribePuenteHGiro(1,0,0);
ConfiguraEscribePuenteHGiro(-1,0,0);
delay(500);

// reset de encoders

encoderIzquierdo.write(0);
encoderDerecho.write(0);

if (!lecturaBotonStop) {
    paro_emergencia=true;
}

//Lógica estado siguiente

if (paro_emergencia){
    paro_emergencia=false;
    estado= ESPERA;
} else if (obstaculo_detectado){
    estado = MOVIMIENTO_OBSTACULO;
} else {
    estado = LEE_MEMORIA;
}

```

```

break;

}

case CICLO: {

if( instrucion == inst_CicloInicia && primer_ciclo ){ //inicio ciclo, primera vez
    cantidad_ciclos = valor_instrucion; //guarda cantidad de ciclos, primero
    primer_ciclo = false;
    inst_inicio_ciclo = inst_actual-1; //almacena apuntador a inicio de ciclo
    cantidad_ciclos--; //resta el primer ciclo que se va a ejecutar

}else if( instrucion == inst_CicloInicia && !primer_ciclo ) { //inicio de ciclo, resto
de veces
    cantidad_ciclos--; //resta 1 a la cantidad de ciclos

}else if( instrucion == inst_CicloFin && cantidad_ciclos != 0 ) { //final de ciclo,
ciclos pendientes
    inst_actual = inst_inicio_ciclo; //devuelve el apuntador al inicio del ciclo

}else if( instrucion == inst_CicloFin && cantidad_ciclos == 0 ) { //final de ciclo,
última vez
    primer_ciclo = true; //reset de variable

}

```

```

//Serial.print("Estado: Ciclo -> ¿Inicia ciclo? ");
//Serial.print(instrucion == inst_CicloInicia);

//Serial.print(" Ciclos pendientes: ");
//Serial.println(cantidad_ciclos);

//delay(5000);

//Lógica estado siguiente

if (!lecturaBotonStop) {

    estado = ESPERA;

} else {

    estado = LEE_MEMORIA;

}

break;

}

case OBSTACULOS: {

    /*Activa la bandera de si detectar obstáculos y luego la desactiva

    if ( instrucion == inst_ObstaculoInicia ){

        obstaculos_activo = true;

    }else if ( instrucion == inst_ObstaculoFin ) {

        obstaculos_activo = false;
}

```

```

}

//Serial.print("Estado: Obstáculos -> ");

//Serial.println(obstaculos_activo);

//delay(4000);

//Lógica estado siguiente

if (!lecturaBotonStop) {

    estado = ESPERA;

} else {

    estado = LEE_MEMORIA;

}

break;

}

case MOVIMIENTO_OBSTACULO: {

    // *Genera el movimiento despues que un obstaculo se detect, el cual corresponde a un
    retroceso

    retroceso_listo=AvansarDistanciaDeseada(-50, contadorPulsosDerecha,
    contadorPulsosIzquierda);

    obstaculo_detectado=false;

    // Lógica estado siguiente

    if (!lecturaBotonStop) {

```

```

paro_emergencia=true;

estado = DETENERSE;

} else if (retroceso_listo){

ConfiguraEscribePuenteH(-1,0,0);

estado = ESPERA;

}

break;

}

case NADA: {

break;

}

}


```

//FUNCIONES DE INTERPRETACION DE MENSAJES

```

//*****  

*****  

//Función que interpreta el mensaje recibido de la app de programación  

//  

//Revisa el mensaje que se recibió por Bluetooth BLE y revisa de 5 en 5

```

```

//los caracteres. Los guarda en un array global luego de interpretar que
//instrucción es y que valor tienen

//



//@param mensaje Cadena de 512 caracteres máximo recibidos

//



//*****



void Interpreta_mensajeBLE (string mensaje) {

    for (int i = 0; i < mensaje.length(); i=i+5 ) {

        string comando = mensaje.substr(i, 5);

        string instrucion = mensaje.substr(i, 2);

        string valor_instrucion =  mensaje.substr(i+2, 3);





        //if (comando.compare("ATINI")==0) {

            if (comando == "ATINI"){

                //Serial.println("mensaje viene bien");





            }else if (comando == "ATFIN"){

                //Serial.println("Mensaje finalizado");

                inst_final = i/5;

            }

        }

    }

}


```

```

}else if (comando == "OBINI"){

//Serial.println("Obstaculos habilitados");

lista_instrucciones[inst_actual][0] = inst_ObstaculoInicia;

lista_instrucciones[inst_actual][1] = 0;

inst_actual++;

}

}else if (comando == "OBFIN"){

//Serial.println("Obstaculos deshabilitados");

lista_instrucciones[inst_actual][0] = inst_ObstaculoFin;

lista_instrucciones[inst_actual][1] = 0;

inst_actual++;

}

}else if (comando == "CIFIN"){

//Serial.println("Ciclo finaliza");

lista_instrucciones[inst_actual][0] = inst_CicloFin;

lista_instrucciones[inst_actual][1] = 0;

inst_actual++;

}

}else if (instrucion == "CI" && comando != "CIFIN"){

//Serial.print("Ciclo inicia: ");

short valor = stoi (valor_instrucion);

//Serial.println(valor);

```

```

lista_instrucciones[inst_actual][0] = inst_CicloInicia;

lista_instrucciones[inst_actual][1] = valor;

inst_actual++;

}

}else if (instrucion == "AV"){

//Serial.print("Avanza: ");

short valor = stoi (valor_instrucion);

//Serial.println(valor);

lista_instrucciones[inst_actual][0] = inst_Avanzar;

lista_instrucciones[inst_actual][1] = valor;

inst_actual++;

}

}else if (instrucion == "RE"){

//Serial.print("Retrocede: ");

short valor = stoi (valor_instrucion);

//Serial.println(valor);

lista_instrucciones[inst_actual][0] = inst_Retroceder;

lista_instrucciones[inst_actual][1]= valor;

inst_actual++;

}

}else if (instrucion == "GI"){

//Serial.print("Gira izquierda: ");

short valor = stoi (valor_instrucion);

```

```

//Serial.println(valor);

lista_instrucciones[inst_actual][0] = inst_GiroIzquierdo;

lista_instrucciones[inst_actual][1] = valor;

inst_actual++;

}

}else if (instrucion == "GD"){

//Serial.print("Gira derecha: ");

short valor = stoi (valor_instrucion);

//Serial.println(valor);

lista_instrucciones[inst_actual][0] = inst_GiroDerecho;

lista_instrucciones[inst_actual][1] = valor;

inst_actual++;

}

}

inst_actual = 0;

}

```

//////// FUNCIONES DE MOVERSE (AVANCE Y RETROCESO)///////////

```

***** ****
***** ****
//Función que determina si es avance, o retroceso en base a los valores de distancia y

```

```

//configura los pines del Puente H

//Utiliza el valor de distancia para determinar si se va a avanzar o retroceder

//despues de esto escribe los valores de PMW acordes a la accion del control

//



//@param distanciaDeseada la distancia que se movera el robot en cm

//@param pmwRuedaDer el valor de pmw para la rueda derecha basado en el control PID

//@param pmwRuedaIzq el valor de pmw para la rueda izquierda basado en el control PID

//



//****************************************************************************
*****



void ConfiguraEscribePuenteH (int distanciaDeseada, int pwmRuedaDer, int
pwmRuedaIzq) {

    if (distanciaDeseada >= 0) {

        analogWrite(motorIzquierdoA, pwmRuedaDer);

        analogWrite(motorDerechoB, pwmRuedaIzq);

    } else if (distanciaDeseada<= 0) {

        analogWrite(motorIzquierdoB, pwmRuedaDer);

        analogWrite(motorDerechoA, pwmRuedaIzq);

    }

}

```

```

}

//*****
*****



//Función que realiza el cálculo de la distancia lineal recorrida por cada rueda

//



//Utiliza el contador de encoders de la biblioteca ENCODERS para determinar la distancia avanzada

//por el robot al dividir este contador entre los pulsos por milímetros para clacular la distacia en mm

//



//



//@param contadorPulsosIzquierda el contador de pusos del encoder izquierdo

//@param contadorPulsosDerecha el contador de pusos del encoder derecho

//@return DistanciaLineal la distancia avanzada de forma lineal en mm

//



//*****
*****



float calculaDistanciaLinealRecorrida(long contadorPulsosIzquierda, long contadorPulsosDerecha) {

    float distLinealRuedaDerecha = -contadorPulsosIzquierda / pulsosPorMilimetro;
}

```

```

float distLinealRuedaIzquierda = contadorPulsosDerecha / pulsosPorMilimetro;

float DistanciaLineal = (distLinealRuedaDerecha + distLinealRuedaIzquierda) / 2.0;

//Serial.print(DistanciaLineal);

return DistanciaLineal;

}

//*****
*****



//Funcion para implementar el control PID por velocidad en una rueda.

//



//Se debe tener un muestreo en tiempos constantes, no aparecen las constantes de tiempo en
la ecuación,



//sino que se integran con las constantes Ki y Kd. Se recibe la posición de referencia, la
posición actual



//(medida con los encoders), el error acumulado (por referencia), y el valor anterior del
error (por referencia)



//Entrega el valor de ciclo de trabajo (PWM) que se enviará al motor CD



//



//



//@param velRef la velocidad que se busca alcanzar



//@param velActual velocidad actual del robot



//@param sumErrorVel suma del error de la velocidad que debe ser parametro global



//@param errorAnteriorVel error anterior de la velocidad que debe ser parametro global

```

```

//@return pidTermVel valor pid para ser colocado en los motores para el avance o retroceso

// ****
*****



int ControlVelocidadRueda( float velRef, float velActual, float& sumErrorVel, float&
errorAnteriorVel ) {

    float errorVel = velRef - velActual; //se actualiza el error actual

    //se actualiza el error integral y se restringe en un rango, para no aumentar sin control
    //como son variables que se mantienen en ciclos futuros, se usan variables globales

    sumErrorVel += errorVel * tiempoMuestreoS; //Anteriormente se comportaba como
    producto de sumatoria cuando debía ser sumatoria de productos

    sumErrorVel = constrain(sumErrorVel, errorMinIntegralVelocidad,
    errorMaxIntegralVelocidad);

    //error derivativo (diferencial)

    float difErrorVel = (errorVel - errorAnteriorVel) / tiempoMuestreoS / 10; //Se divide el
    factor derivativo entre 10 debido a la inestabilidad de la velocidad por falta de precisión en
    la medición

    //ecuación de control PID
}

```

```

float pidTermVel = ((KpVel * errorVel) + (KiVel * sumErrorVel) + (KdVel * difErrorVel));
//Agregada división entre 5 porque PID saturaba, buscar donde está el error

//Se limita los valores máximos y mínimos de la acción de control, para no saturarla
pidTermVel = constrain( (int)pidTermVel, limiteInferiorCicloTrabajoVelocidad,
limiteSuperiorCicloTrabajoVelocidad);

//actualiza el valor del error para el siguiente ciclo
errorAnteriorVel = errorVel;

return ((int)pidTermVel);

}

//*****
*****



//Función que ejecuta el avance o retroceso del carro mediante el cálculo de la velocidad y
//la aplicación del control de la misma
//


//Inicia con el cálculo de la velocidad actual del robot creando un ciclo con millis para
//poder definir la cantidad de pulsos en una cantidad de tiempo y de aquí determinar la
velocidad, después

//de esto, usa esta velocidad para implementar el control en cada rueda, además calcula la
distancia que está

```

```

//recorriendose para determinar cuando se ha llegado a la distancia deseada, a esto punto
reinicia las variables y

//convierte el booleano de avanceListo en un true.

//



//@param distanciaDeseada la distancia que se busca avanzar o retroceder, un numero en
mm

//@param contadorPulsosIzquierda el contador de pulsos del encoder izquierdo
//@param contadorPulsosDerecha el contador de pulsos del encoder derecho

//@return avanceListo booleano que indica que se ha avanzado o retrocedido la distancia
deseada

//



//***** *****
*****



bool AvansarDistanciaDeseada (int distanciaDeseada, long pulsosEncoderIzq, long
pulsosEncoderDer) {

    float(contPulsosIzq= pulsosEncoderIzq);

    float(contPulsosDer= pulsosEncoderDer);

    // crea una condicion de tiempo usando millis para aplicar un tiempo de muestreo si usar
    delays ya que no son compatibles con la biblioteca de encoders

    unsigned long currentMillis=millis();

    if ((currentMillis-previousMillis) >= intervaloTiempoMuestreo){

```

```

previousMillis=currentTime;

//se calculan las velocidades de ambas ruedas

float velActualIzquierda = ((float)(contPulsosIzq - contPulsosPasadoIzq) /
pulsosPorMilimetro) / (float)(intervaloTiempoMuestreo*0.001); //velocidad en mm por s

float velActualDerecha = ((float)(contPulsosDer - contPulsosPasadoDer) /
pulsosPorMilimetro) / (float)(intervaloTiempoMuestreo*0.001); //velocidad en mm por s

velActualDerecha= -1.0 * velActualDerecha; // debido a la lectura de encors, la
velocidad derecha es negativa, por lo que se transforma a positiva.

contPulsosPasadoIzq = contPulsosIzq;

contPulsosPasadoDer = contPulsosDer;

float velSetPoint= velRequerida;// velocidad deseada de las ruedas

if (distanciaDeseada<0){ //Si la distancia deseada es negativa, significa retroceder y por
ende la logica de lecturas de encoders es cambiada para que siempre sea positivos

    velActualDerecha= -1.0 * velActualDerecha;

    velActualIzquierda=-1.0*velActualIzquierda;

}

//Serial.print("vel izq: ");

//Serial.println(velActualIzquierda);

//Serial.print("vel der: ");

//Serial.println(velActualDerecha);

//float errorVelocidad = velActualDerecha - velActualIzquierda; //Error entre la
velocidad de las dos ruedas

```

```

float velSetPointDerecha = velSetPoint;// se define la velocidad objetivo para cada rueda
float velSetPointIzquierda = velSetPoint;

//se aplica el lazo de control PID de velocidad para obtener un valor PMW para cada
rueda

int cicloTrabajoRuedaDerecha = ControlVelocidadRueda(velSetPointDerecha,
velActualDerecha, sumErrorVelDer, errorAnteriorVelDer);

//Serial.print("pid der: ");

//Serial.println(cicloTrabajoRuedaDerecha);

int cicloTrabajoRuedaIzquierda = ControlVelocidadRueda(velSetPointIzquierda,
velActualIzquierda, sumErrorVelIzq, errorAnteriorVelIzq);

//Serial.print("pid izq: ");

//Serial.println(cicloTrabajoRuedaIzquierda);

//Se configura el puente H

ConfiguraEscribePuenteH (distanciaDeseada, cicloTrabajoRuedaDerecha,
cicloTrabajoRuedaIzquierda);

// calculo de la distancia

float distanciaAvanzada=
calculaDistanciaLinealRecorrida(contadorPulsosIzquierda,contadorPulsosDerecha);

// define si el avance fue finalizado

bool avanceListo = false;

if (abs(distanciaAvanzada) >= abs(distanciaDeseada)) {

sumErrorVelDer=0;

errorAnteriorVelDer=0;

```

```
sumErrorVelIzq=0;  
errorAnteriorVelIzq=0;  
distanciaAvanzada=0;  
avanceListo=true;  
  
}  
  
return avanceListo;  
}  
}
```

////////// FUNCIONES DE GIRO (GIRO DERECHO E IZQUIERDO)//////////

```
////////////////////////////////////////////////////////////////////////  
*****  
//Función que determina si es giro derecho o giro izquierdo en base a los valores de giro  
//y configura los pines del Puente H  
//  
//Utiliza el valor del giro para determinar si se va a girara a la derecha o izquierda
```

```

//despues de esto escribe los valores de PMW acordes a la accion del control

//



//@param giroDeseado el giro que se movera el robot en grados

//@param pmwRuedaDer el valor de pmw para la rueda derecha basado en el control PID

//@param pmwRuedaIzq el valor de pmw para la rueda izquierda basado en el control PID

//


//*****



void ConfiguraEscribePuenteHGiros (int giroDeseado, int pwmRuedaDer, int
pwmRuedaIzq) {

    if (giroDeseado >= 0) {

        analogWrite(motorIzquierdoB, pwmRuedaDer);

        analogWrite(motorDerechoB, pwmRuedaIzq);

    } else if (giroDeseado<= 0) {

        analogWrite(motorIzquierdoA, pwmRuedaDer);

        analogWrite(motorDerechoA, pwmRuedaIzq);

    }

}

```

```

//*****
*****



//Función que calcula la distancia angular (en grados) que se ha desplazado el robot.

//


//Para robots diferenciales, en base a un giro sobre su eje (no arcos). Recibe la cantidad de
//pulsos de la rueda. Devuelve un ángulo en grados. Esto lo hace al usar el contador de los
encoders

//y transformarlo en la distancia de la circunferencia obtenida.



//


//


//@param cantPulsos contador de pulsos del encoder de la rueda correspondiente
//@return despAngular desplazamiento angular del robot

//


//*****
*****



float ConvDistAngular(float cantPulsos) {

    //Para robots diferenciales, en base a un giro sobre su eje (no arcos)

    //Recibe la cantidad de pulsos de la rueda. Devuelve un ángulo en grados.

    float despAngular = cantPulsos / (distanciaCentroARueda * pulsosPorMilimetro) * 180 /
PI;

```

```

    return despAngular;

}

//*****
*****



//Funcion para implementar el control PID por posición en una rueda.

//


//Se debe tener un muestreo en tiempos constantes, no aparecen las constantes de tiempo en
la ecuación,


//sino que se integran con las constantes Ki y Kd. Se recibe la posición de referencia, la
posición actual


//(medida con los encoders), el error acumulado (por referencia), y el valor anterior del
error (por referencia)

//Entrega el valor de ciclo de trabajo (PWM) que se enviará al motor CD

//


//


//@param posRef el ángulo que se busca alcanzar

//@param posActual ángulo actual del robot

//@param sumErrorGiro suma del error del giro que debe ser parametro global

//@param errorAnteriorGiro error anterior del giro que debe ser parametro global

//@return pidTermGiro valor pid para ser colocado en los motores para el avance o
retroceso

//



```

```

//*****
*****



int ControlPosGiroRueda( float posRef, float posActual, float& sumErrorGiro, float&
errorAnteriorGiro ) {

    float errorGiro = posRef - posActual; //se actualiza el error actual

    //Serial.print("error giro: ");

    //Serial.println(errorGiro);

    //se actualiza el error integral y se restringe en un rango, para no aumentar sin control

    //como son variables que se mantienen en ciclos futuros, se usan variables globales

    sumErrorGiro += errorGiro * tiempoMuestreoS;

    sumErrorGiro = constrain(sumErrorGiro, errorMinIntegralGiro, errorMaxIntegralGiro);

    //error derivativo (diferencial)

    float difErrorGiro = (errorGiro - errorAnteriorGiro) / tiempoMuestreoS;

    //ecuación de control PID

    float pidTermGiro = (KpGiro * errorGiro) + (KiGiro * sumErrorGiro) + (KdGiro *
difErrorGiro);
}

```

```

//Se limita los valores máximos y mínimos de la acción de control, para no saturarla
pidTermGiro = constrain( int (pidTermGiro), limiteInferiorCicloTrabajoGiro,
limiteSuperiorCicloTrabajoGiro);

//Restricción para manejar la zona muerta del PWM sobre la velocidad
if (-cicloTrabajoMinimo < pidTermGiro && pidTermGiro < cicloTrabajoMinimo) {
    pidTermGiro = 0;
}

//actualiza el valor del error para el siguiente ciclo
errorAnteriorGiro = errorGiro;

return pidTermGiro;
}

//*****
*****



//Función que revisa el estado de la acción de control y si se mantiene varios ciclos en
//cero, asume que ya está en el estado estacionario
//
//Utiliza el valor de PMW para cada motor y si determina que estos son cero, determina que
//el giro fue completado.

```

```

//



//@param contCiclosEstacionario el contador de ciclos que se llevan a cabo en 0

//@param pmwRuedaDer el valor de pmw para la rueda derecha basado en el control PID

//@param pmwRuedaIzq el valor de pmw para la rueda izquierda basado en el control PID

//@return estadoEstacionarioAlcanzado booleano que determina que el estado estacionario
fue alcanzado

//


//*****



bool EstadoEstacionario (int pwmRuedaDer, int pwmRuedaIzq, int&
contCiclosEstacionario) {

    //Recibe los ciclos de trabajo en cada rueda y una variable por referencia donde se
almacenar cuantos ciclos seguidos se llevan

    //Devuelve una variable que es TRUE si ya se alcanzó el estado estacionario


bool estadoEstacionarioAlcanzado = false;

if (pwmRuedaDer == 0 && pwmRuedaIzq == 0) {

    contCiclosEstacionario++;

    if (contCiclosEstacionario > minCiclosEstacionario) {

        //ResetContadoresEncoders();

        contCiclosEstacionario = 0;

        estadoEstacionarioAlcanzado = true;
    }
}
}

```

```

    }

} else {

    contCiclosEstacionario = 0;

}

return estadoEstacionarioAlcanzado;

}

//*****
*****



//Función que ejecuta el giro del robot sobre su propio eje.

//



//Usa control PID en cada rueda. Este control PID permite el movimiento hasta llegar al
giro

//desado, una vez aca se usa la funcion de estado estacionario para definir que el giro ha
//terminado por completo y si es asi define el booleano de giro terminado como real.

//


//@param grados la cantidad de grados que se desea girar
//@return giroListo booleano que determina que el giro ha sido realizado

//


//*****
*****



bool Giro(float grados) {

```

```
grados = 1.2 * grados;  
bool negativo = false;  
posActualRuedaDerecha = abs(ConvDistAngular(contadorPulsosIzquierda));  
posActualRuedaIzquierda = abs(ConvDistAngular(contadorPulsosDerecha));
```

```
if (grados<0){
```

```
    negativo = true;
```

```
    grados = -grados;
```

```
}
```

```
int cicloTrabajoRuedaDerecha = ControlPosGiroRueda( grados, posActualRuedaDerecha,  
sumErrorGiroDer, errorAnteriorGiroDer );  
  
//Serial.print("PIDder: ");  
  
//Serial.println(cicloTrabajoRuedaDerecha);
```

```
int cicloTrabajoRuedaIzquierda = ControlPosGiroRueda( grados,  
posActualRuedaIzquierda, sumErrorGiroIzq, errorAnteriorGiroIzq);
```

```
//Serial.print("PIDizq: ");
```

```
//Serial.println(cicloTrabajoRuedaIzquierda);
```

```
if (negativo){
```

```
    grados=-grados;
```

```
    negativo=false;
```

```
}
```

```
ConfiguraEscribePuenteHGiro (grados,cicloTrabajoRuedaDerecha,  
cicloTrabajoRuedaIzquierda);
```

```
bool giroListo = EstadoEstacionario (cicloTrabajoRuedaDerecha,  
cicloTrabajoRuedaIzquierda, contCiclosEstacionarioGiro);
```

```
if (giroListo){
```

```
    sumErrorGiroDer = 0;
```

```
    errorAnteriorGiroDer = 0;
```

```
    sumErrorGiroIzq = 0;
```

```
    errorAnteriorGiroIzq = 0;
```

```
}
```

```
return giroListo;
```

```
}
```

Apéndice 3. Medidas tomadas durante la validación

Tabla 4.1 Medidas tomadas para la validación del movimiento

Medidas	Niveles (± 1) mm			
	100	200	300	900
1	99	182	298	889
2	98	192	305	895
3	95	192	296	909
4	93	202	305	905
5	98	203	300	905
6	99	196	299	903
7	98	194	300	909
8	99	189	297	899
9	98	204	304	897
10	100	199	303	886
11	99	199	301	895
12	100	204	298	897
13	97	202	300	905
14	98	203	295	903
15	91	200	298	910
16	98	195	304	900
17	102	192	308	909
18	100	197	301	900
19	101	198	308	894
20	99	200	301	895
21	104	199	305	899
22	98	196	301	894
23	97	203	305	906
24	99	202	303	907
25	97	199	302	905
26	97	204	304	898

27	100	193	305	893
28	102	200	298	905
29	103	198	297	899
30	100	200	302	900

Tabla 4.2 Medidas tomadas para la validación de los ángulos

Medidas	Niveles ($\pm 0.5^\circ$)					
	20	45	90	120	180	240
1	16.0	43.0	94.0	117.0	185.0	235.0
2	19.0	48.0	90.0	115.0	175.0	235.0
3	21.0	46.0	93.0	120.0	178.0	240.0
4	18.0	46.0	88.0	123.0	178.0	237.0
5	20.0	50.0	93.0	120.0	182.0	238.0
6	17.0	47.0	91.0	116.0	176.0	237.0
7	19.0	40.0	92.0	122.0	179.0	239.0
8	20.0	44.0	91.0	120.0	182.0	236.0
9	20.0	45.0	94.0	125.0	179.0	238.0
10	23.0	41.0	92.0	122.0	183.0	241.0
11	19.0	43.0	88.0	120.0	173.0	239.0
12	20.0	40.0	90.0	120.0	180.0	240.0
13	21.0	44.0	91.0	119.0	181.0	243.0
14	20.0	45.0	91.0	118.0	178.0	241.0
15	18.0	44.0	93.0	121.0	181.0	239.0
16	20.0	47.0	90.0	120.0	178.0	235.0
17	17.0	47.0	91.0	115.0	178.0	237.0
18	18.0	46.0	89.0	119.0	176.0	237.0
19	17.0	44.0	88.0	121.0	180.0	239.0
20	17.0	41.0	89.0	120.0	184.0	242.0
21	18.0	45.0	89.0	117.0	182.0	237.0

22	23.0	46.0	85.0	118.0	181.0	240.0
23	22.0	45.0	91.0	116.0	176.0	241.0
24	19.0	45.0	87.0	116.0	182.0	242.0
25	19.0	42.0	89.0	118.0	178.0	237.0
26	18.0	44.0	85.0	118.0	182.0	238.0
27	18.0	46.0	86.0	119.0	173.0	239.0
28	20.0	44.0	89.0	120.0	180.0	243.0
29	19.0	45.0	90.0	121.0	182.0	240.0
30	21.0	47.0	88.0	117.0	179.0	241.0

Tabla 4.3 Medidas tomadas para la validación de la detección de obstáculos

Medidas	Distancia($\pm 1\text{mm}$)
1	101
2	98
3	95
4	93
5	94
6	91
7	87
8	87
9	87
10	84
11	81
12	80
13	80
14	80
15	77
16	77
17	76

18	75
19	69
20	68
21	66
22	59
23	59
24	52
25	55
26	51
27	50
28	47
29	42
30	40

Tabla 4.4 Medidas tomada para la validación de la facilidad de carga.

Medidas	Tiempo (±0.5s)
1	32.0
2	37.0
3	38.0
4	34.0
5	58.0
6	35.0
7	21.0
8	45.0
9	37.0
10	35.0
11	58.0
12	45.0

13	29.0
14	37.0
15	48.0
16	49.0
17	33.0
18	60.0
19	38.0
20	56.0
21	49.0
22	39.0
23	51.0
24	45.0
25	48.0
26	45.0
27	32.0
28	43.0
29	45.0
30	59.0