

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y  
ELECTRÓNICA

CARRERA DE INGENIERÍA EN  
ELECTRÓNICA, AUTOMATIZACIÓN Y  
CONTROL

PROYECTO DE GRADO PARA LA  
OBTENCIÓN DEL TÍTULO EN INGENIERÍA  
“DESARROLLO DE UN SISTEMA DE ROBÓTICA  
COOPERATIVA ENTRE DOS ELEMENTOS ROBOTS  
TIPO ROBONOVA”

ELABORADO POR:

ANA KAREN APOLO PEÑALOZA

ANABEL ALEJANDRA VELASCO BARRERA

SANGOLQUÍ – ECUADOR

2013

# **ESCUELA POLITÉCNICA DEL EJÉRCITO**

## **INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL**

### **DECLARACIÓN DE RESPONSABILIDAD**

**APOLO PEÑALOZA ANA KAREN**

**VELASCO BARRERA ANABEL ALEJANDRA**

El proyecto de grado denominado “Desarrollo de un sistema de robótica cooperativa entre dos elementos robots tipo ROBONOVA”, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Sangolquí, 29 de mayo del 2013

---

Apolo Peñaloza Ana Karen

---

Velasco Barrera Anabel Alejandra

# **ESCUELA POLITÉCNICA DEL EJÉRCITO**

**INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y**

**CONTROL**

**AUTORIZACIÓN**

Nosotras, Apolo Peñaloza Ana Karen y Velasco Barrera Anabel Alejandra

Autorizamos a la Escuela Politécnica del Ejército la publicación, en la biblioteca virtual de la Institución del trabajo “Desarrollo de un sistema de robótica cooperativa entre dos elementos robots tipo ROBONOVA”, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Sangolquí, 29 de mayo del 2013

---

Apolo Peñaloza Ana Karen

---

Velasco Barrera Anabel Alejandra

# **ESCUELA POLITÉCNICA DEL EJÉRCITO**

## **INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y**

### **CONTROL**

### **CERTIFICADO**

Ing. Alejandro Chacón

Ing. Alexander Ibarra

Que el trabajo titulado “Desarrollo de un sistema de robótica cooperativa entre dos elementos robots tipo ROBONOVA”, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la ESPE, en el Reglamento de Estudiantes de la Escuela Politécnica del Ejército.

Debido a que se trata de un trabajo de investigación recomiendan su publicación.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf). Autorizan a la Srta. Apolo Peñaloza Ana Karen y a la Srta. Velasco Barrera Anabel Alejandra que lo entreguen al Ingeniero Luis Orozco, en su calidad de Coordinador de la Carrera

Sangolquí, 29 de mayo del 2013

---

Ing. Alejandro Chacón

DIRECTOR

---

Ing. Alexander Ibarra

CODIRECTOR

## CERTIFICACIÓN

Certificamos que el presente proyecto de grado, **“DESARROLLO DE UN SISTEMA DE ROBÓTICA COOPERATIVA ENTRE DOS ELEMENTOS ROBOTS TIPO ROBONOVA”**, fue desarrollado en su totalidad por las señoritas: Ana Karen Apolo Peñaloza, y Anabel Alejandra Velasco Barrera bajo nuestra dirección.

Atentamente:

---

Ing. Alejandro Chacón

DIRECTOR

---

Ing. Alexander Ibarra

CODIRECTOR

# ***Agradecimiento***

***A Dios***

*Por permitir cumplir mis sueños, iluminar mi vida.*

***A mis padres Luis y Carmita***

*Por ser mi apoyo incondicional, confiar siempre en que puedo realizar mis sueños por más difíciles que se vean, por luchar conmigo para cumplir esta meta con su apoyo, aconsejándome, escuchándome y sobre todo brindando su amor, muchas gracias por ser mis padres LUIS Y CARMITA, les debo muchas cosas y he aprendido mucho.*

***A mis hermanos Bryan y Mauricio***

*Por estar conmigo en todos los momentos, aportando ideas y criticando otras, gracias por el tiempo prestado para cumplir mi sueño.*

***A mis profesores***

*Por haber sido un pilar para mis conocimientos, y permitirme haber entrado en el maravilloso mundo del saber.*

***A mis amigos***

*No quiero nombrar uno a uno, tal vez me olvidaría algún nombre, son muchas las personas que me apoyaron en el transcurso de este objetivo, por eso quiero dar un sincero agradecimiento a todos mis amigos.*

*Ana Karen Apelo P.*

## *Agradecimiento*

*A:*

*Dios Por haberme acompañado y guiado a lo largo de mi carrera.*

*Mis padres Esthela y Luis Alfonso por ser mi apoyo incondicional en esta meta cumplida y a lo largo de mi vida, por enseñarme que los sueños son posibles si se tiene perseverancia, gracias padres por su amor infinito.*

*Mis hermanas Joselyn y Ma. Fernanda por ser parte fundamental de mi vida, por ser mis confidentes, por escucharme y aconsejarme en esta etapa que ha culminado y a lo largo de mi vida, hermanas gracias por apoyarme y quererme tanto como yo las quiero a ustedes.*

*Mi hija Danna Alejandra por iluminar día a día mi camino solo con una sonrisa y sin la necesidad decir mucho me animaste a seguir luchando para cumplir una de las metas en mí vida, ser profesional.*

*Daniel por acompañarme en esta última etapa de mi carrera profesional, por estar a mi lado y ser mi fortaleza cuando lo necesito.*

*Mis profesores por los conocimientos impartidos, por el tiempo invertido en mi educación, por la confianza depositada en mí.*

*Edwin por ser mi amigo, por escuchar mis penas y alegrías, aconsejarme todas las veces en las que me encontré derrotada y siempre sacarme una sonrisa y darme apoyo en especial en la culminación de mi profesión.*

*Mis amigos por creer en mí, por todas las risas y desvelos que vivimos a lo largo de la etapa universitaria sin ustedes no hubiera sido lo mismo.*

*Anabel Velasco B.*

## ***Dedicatoria***

### ***A Dios***

*Por haberme permitido llegar hasta este punto y por  
Haberme iluminarme mi camino para lograr mis sueños.*

### ***A mis padres Luis y Carmita***

*Por ser mi guía en mi camino, sus sabios consejos  
me han permitido crecer como ser humano,  
su incondicional apoyo me ayuda a cumplir mis sueños.*

### ***A mis hermanos Mauricio y Bryan***

*Por ser mis consejeros, confidentes, por siempre  
creer en mí y confiar que podía lograr mis sueños.*

*Ana Karen Apelo P.*



## ***Dedicatoria***

*A:*

*Dios Por haberme permitido llegar hasta este punto y haberme dado salud e inteligencia para lograr mis objetivos, además de su infinita bondad y amor.*

*Mis padres Esthela y Luis Alfonso por ser el pilar fundamental en todo lo que soy, en toda mi educación, tanto académica, como de la vida, por su incondicional apoyo perfectamente mantenido a través del tiempo, por sus consejos, por sus ejemplos de perseverancia y constancia que los caracterizan y que me ha infundado siempre y por su amor.*

*Mi hija Danna Alejandra por ser mi fuente inspiración para superarme, mi fortaleza día a día, por brindarme su amor, es por ti mi deseo de ser mejor persona y profesional para darte el ejemplo a seguir te dedicó no solo este título sino también los próximos.*

*Mi hermana Joselyn, por quererme y apoyarme siempre, por estar siempre a mi lado y ser el motor de superación en mi vida, esto también te lo debo a ti.*

*Daniel si bien es cierto poco es el tiempo que nos conocemos, pero también es cierto que me diste muchas alegrías, y mi felicidad más grande, por ser mi amigo y compañero fuente de calma y consejo en todo momento.*

*Anabel Velasco B.*

*Los grandes sueños son de los locos geniales, lo ejecutan los  
luchadores tenaces, lo disfrutan los felices mortales y lo critican  
los inútiles*

**Anónimo**

*Sólo porque algo no haga lo que era previsto no quiere decir que  
sea inútil el esfuerzo.*

**Thomas Alva Edison**

## *Prólogo*

El ser humano, siempre busca desarrollar nuevas ideas, las cuales le simplifiquen la vida, en las tareas tediosas, los robots fueron diseñados para simplificar la vida del ser humano. En la actualidad la robótica ha avanzado significativamente permitiendo a los robots tener inteligencia, creando nuevas divisiones de la robótica, así nació el término de robótica cooperativa la cual representa a un grupo de robots interactuando entre sí para lograr objetivos comunes.

Diversos son los problemas planteados en la vida diaria, muchas veces son monótonos de realizar, y el uso de la robótica para facilitar la vida del ser humano se vuelve costosa para ser implementada fácilmente, la robótica cooperativa pretende solucionar varios de estos problemas (mover objetos, limpiar áreas, ...) disminuyendo el tiempo de ejecución y el costo de implementación, a través de la asociación de agentes, es decir el uso de robots sencillos, trabajando en conjunto para lograr una meta específica.

Una de las interrogantes planteadas es: ¿Cómo los seres humanos mueven una caja pesada?, en la mayoría de los casos piden ayuda a otra persona, coordinan habilidades, realizan la tarea de mover la caja pesada al sitio asignado. Un robot también puede mover una caja, pero ¿Si la caja es demasiado pesada para el robot?, podría ser movida por un grupo de robots, se sincronizaran para realizar la tarea asignada, a estos robots se los denomina como robots cooperativos.

El presente proyecto describe una de las interrogantes de la robótica cooperativa, mover una caja con varios robots, para lo cual, se pretende sentar las bases para futuras investigaciones, se desarrolló e implementó un sistema de software el mismo simula la cooperación entre 2 agentes con un objetivo, este proyecto se divide en 2 etapas

La primera mover una caja en una trayectoria lineal, y la segunda en una trayectoria circular, para lo cual primero se establece el manejo de la plataforma, las comunicaciones entre los agentes y se define la autonomía de los mismos.

La arquitectura implementada es distribuida también se evaluó el nivel de cooperación alcanzado entre los agentes y el desempeño en un grupo, a continuación se detallará el desarrollo en los capítulos correspondientes.

## ÍNDICE DE CONTENIDO

<b>CAPÍTULO I INTRODUCCIÓN</b> .....	<b>1</b>
1.1 ANTECEDENTES.....	1
1.2 JUSTIFICACIÓN E IMPORTANCIA.....	3
1.3 ALCANCE .....	4
1.4 OBJETIVOS .....	5
1.4.1 General.....	5
1.4.2 Específicos .....	5
1.5 DESCRIPCIÓN GENERAL DEL PROYECTO .....	5
<b>CAPÍTULO II MARCO TEÓRICO.</b> .....	<b>7</b>
2.1 ROBÓTICA COOPERATIVA.....	7
2.1.1 Historia.....	7
2.1.2 Origen.....	8
2.1.3 Desarrollo .....	9
2.1.4 Definición.....	9
2.1.5 Clasificación por nivel de comunicación de los sistemas de Robótica Cooperativa .....	10
2.1.5.1 Muy acoplados(o colectivos) .....	10
2.1.5.2 Pocos acoplados (o manadas).....	11
2.1.6 Ventajas y Desventajas.....	11
2.1.7 Aplicaciones.....	12
2.2 CONTROL EN ROBÓTICA COOPERATIVA .....	13
2.2.1 Hardware .....	13
2.2.1.1 Arquitectura Centralizada .....	13

2.2.1.2	Arquitectura Distribuida .....	14
2.2.2	Software .....	15
2.2.2.1	Agente.....	15
2.2.2.1.1	Características de los agentes .....	16
2.2.2.1.2	Clasificación de los Agentes .....	17
2.2.2.1.2.1	Clasificación de los agentes por el grado de inteligencia.....	17
2.2.2.1.2.1.1	Agente Inteligente.....	18
2.2.2.1.2.1.2	Características de un agente inteligente .....	19
2.2.2.1.2.2	Clasificación de los agentes por el tipo de programación .....	20
2.2.2.1.2.2.1	Agentes Basado en Metas.....	20
2.2.2.1.2.2.2	Característica de un Agentes Basado en Metas .....	22
2.2.2.2	Sistema Multi Agentes (SMA).....	22
2.3	ESTADO DEL ARTE .....	25

## **CAPÍTULO III DISEÑO DE HARDWARE. ....27**

3.1	SISTEMA MULTI – ROBOT HARDWARE .....	27
3.1.1	Arquitectura De Control Distribuido.....	28
3.1.1.1	Introducción.....	28
3.1.2	Arquitectura implementada .....	31
3.1.2.1	Requerimientos del sistema Multi-robot .....	32
3.1.2.2.	Selección de la plataforma utilizada en el proyecto de Robótica Cooperativa.....	33
3.1.2.2.1	Controlador del Sistema Multi - Robot.....	34
3.1.2.2.1.1	Entradas Universales: Análogas .....	36
3.1.2.2.1.1.1	Entrada Análogas en modo 5[kΩ] .....	36
3.1.2.2.1.1.2	Entrada Análogas en modo 10[V].....	37

3.1.2.2.1.2	Entradas Digitales.....	38
3.1.2.2.1.3	Entrada a Contadores .....	39
3.1.2.2.2	Sensores .....	40
3.1.2.2.3	Comunicaciones .....	46
3.1.1.2.4	Actuadores .....	46
3.1.3	Prototipos para el Sistema Multi - Robot .....	47
3.1.3.1	“TANQUE” versión 1.0.....	48
3.1.3.2	“TANQUE” versión 1.1.....	51

## **CAPÍTULO IV DISEÑO DE SOFTWARE.....54**

4.1	SISTEMAS MULTI – ROBOT SOFTWARE.....	54
4.1.1	Algoritmos .....	55
4.1.1.1	Introducción .....	55
4.1.1.2	Algoritmo Implementado .....	55
4.1.1.2.1	Consideraciones particulares de la solución.....	55
4.1.1.2.1.1	Descripción del Diagrama de Flujo .....	60
4.1.1.2.2	Aplicaciones para la demostrar robótica cooperativa.....	62
4.1.1.2.2.1	Aplicación 1 .....	62
4.1.1.2.2.1.1	Etapa 1.....	64
4.1.1.2.2.1.2	Etapa 2.....	69
4.1.1.2.2.2	Aplicación 2 .....	72
4.1.1.3	Diagrama de Flujo.....	74
4.1.2	Programación .....	75
4.1.2.1	Introducción .....	75
4.1.2.2	Modos de programación.....	76

4.1.2.2.1	Modo en línea .....	76
4.1.2.2.1.1	Programación basada en Visual C++ .....	76
4.1.2.2.1.2	Requerimientos del Computador para C++ .....	77
4.1.2.2.2	Modo en ejecución .....	77
4.1.2.2.2.1	Programación en C .....	78
4.1.2.2.2.2	Requerimientos del Computador para C .....	80
4.1.2.2.2.3	Instalación .....	80
4.1.2.3	Diagrama Estructurado del Programa .....	81
4.1.2.3.1	Variables Utilizadas en el Programa .....	84
4.1.2.4	Funciones de la Librería ftMscLib .....	86
4.1.2.4.1	Funciones Principales .....	86

## **CAPÍTULO V PRUEBA Y RESULTADOS.....92**

5.1	SISTEMAS MULTI – ROBÓTICOS PRUEBAS Y RESULTADOS .....	92
5.1.1	Pruebas del Sistema de Robótica Cooperativa .....	92
5.1.1.1	Prueba 1: Movimiento lineal de una caja .....	93
5.1.1.1.1	Primera Etapa .....	93
5.1.1.1.2	Segunda Etapa .....	98
5.1.1.1.2.1	Fase 1: Determino el peso de la caja .....	99
5.1.1.2	Prueba 2 Mover la caja con trayectoria circular .....	102
5.1.1.2.1	Etapa 1 .....	102
5.1.1.2.2	Etapa 2 .....	107
5.1.1.3	Nivel de Cooperación alcanzado .....	107
5.1.1.4	Sistema Multi – Agentes .....	109
5.1.2	Análisis de resultados finales .....	109



5.1.2.1	Problemas encontrados en el proyecto .....	110
5.1.2.2	Análisis Mecánico de “TANQUE” .....	111

**CAPÍTULO VI .RESULTADOS DEL PROYECTO.....114**

6.1	RESULTADOS DEL PROYECTO .....	114
6.1.1	Introducción .....	115
6.1.2	Consideraciones para desarrollar proyectos de robótica cooperativa. ....	115
6.1.2.1	Mover cajas pesadas con un grupo de robots .....	116
6.1.2.2	Jugar fútbol con un grupo de robots.....	117
6.1.3	Consideraciones para implementar proyectos de robótica cooperativa.....	117
6.1.4	Implementación de una de las interrogantes de la robótica cooperativa (mover cajas pesadas) .....	120
6.1.5	Consideraciones de una plataforma, para desarrollar proyectos de robótica cooperativa. ....	122
6.1.5.1	Robots Móviles: Robótica Cooperativa aplicable a futuro (nivel industrial).....	122
6.1.5.2	Robots Humanoides: Robótica Cooperativa semejanza con el comportamiento del ser humano .....	124

**CAPÍTULO VII CONCLUSIONES.....125**

7.1	CONCLUSIONES .....	125
7.2	RECOMENDACIONES .....	127

**REFERENCIAS BIBLIOGRÁFICAS .....128**

**ANEXOS ..... ¡ERROR! MARCADOR NO DEFINIDO.**

**ANEXO 1 DATASHEET ..... ¡Error! Marcador no definido.**

**ANEXO 2 PC PROGRAMMING ..... ¡Error! Marcador no definido.**

ANEXO 3 C COMPILER FOR ROBOTICS PROGRAMS.....	¡Error! Marcador no definido.
ANEXO 4 INVESTIGACIONES DEL PROYECTO .....	¡Error! Marcador no definido.
Investigaciones del Proyecto .....	¡Error! Marcador no definido.
Investigaciones de Robótica Cooperativa Realizadas a nivel Mundial .....	¡Error! Marcador no definido.
Cuadro de Investigaciones realizadas a nivel mundial .....	¡Error! Marcador no definido.
ANEXO 5 ROBOTS HUMANOIDES .....	¡Error! Marcador no definido.
Robots Humanoides .....	¡Error! Marcador no definido.
Características Principales para Adquisición del Robot .....	¡Error! Marcador no definido.
Características técnicas de Robot Humanoides .....	¡Error! Marcador no definido.
ANEXO 6 PROFORMAS.....	¡Error! Marcador no definido.
Proformas .....	¡Error! Marcador no definido.
ANEXO 7 LIMITANTES EN ADQUISICIÓN .....	¡Error! Marcador no definido.
Limitantes de Adquisición .....	¡Error! Marcador no definido.
Limitantes en la adquisición de la plataforma humanoide tipo Robonova .....	¡Error! Marcador no definido.
ANEXO 8 ROBOTS MÓVILES.....	¡Error! Marcador no definido.
Robots Móviles.....	¡Error! Marcador no definido.
Características técnicas de los robots móviles de la ESPE.....	¡Error! Marcador no definido.
Características de Selección para robots móviles.....	¡Error! Marcador no definido.
Selección del Robot móvil de acuerdo a los pesos obtenidos.....	¡Error! Marcador no definido.
<b>ÍNDICE DE FIGURAS .....</b>	<b>I</b>
<b>ÍNDICE DE FIGURAS DE ANEXOS.....</b>	<b>¡ERROR! MARCADOR NO DEFINIDO.</b>
<b>ÍNDICE DE TABLAS .....</b>	<b>V</b>

**ÍNDICE DE TABLAS DE ANEXOS.....¡ERROR! MARCADOR NO DEFINIDO.**

## CAPÍTULO I .

### INTRODUCCIÓN



Figura I-1: Estructura del Capítulo I

#### 1.1 ANTECEDENTES

El desarrollo tecnológico permite en la actualidad el diseño y la implementación de sistemas cada día más complejos y a la vez más versátiles, esto motiva al desarrollo de nuevos esquemas, elementos y sistemas, el cual es un campo en desarrollo constante tanto a nivel aplicativo como de investigación, actualmente es la robótica colaborativa y colectiva, básicamente se pretende recoger la información de diferentes elementos robóticos y formar una sola base de conocimiento.

Han surgido nuevas tecnologías en la rama de la robótica, una de ellas es *“La robótica cooperativa se interesa, de manera general, en el diseño y el control de sistemas multi-robot o multi-agente, grupos de robots o agentes trabajando de manera conjunta para lograr objetivos comunes. Estos sistemas son útiles en tareas en las cuales un solo*

*individuo es incapaz de realizarla, o bien en tareas en las cuales se obtiene una ganancia”*  
[León Yazid, 2007].

La robótica cooperativa está inspirada en los comportamientos abstraídos de la naturaleza, por ejemplo miles de abejas trabajando sin mando construyen una colmena, millones de neuronas comunicándose configuran un cerebro. ¿Qué puede hacer un enjambre de robots? de aquí nace el concepto de robótica cooperativa.

La robótica cooperativa busca diseñar sistemas compuestos de varios robots capaces de resolver problemas conjuntamente. Los robots forman parte de un sistema multi-robot o multi-agentes son simples en términos de diseño y control, menos costosos, a diferencia de los sistemas de un sólo robot especializado. Los sistemas multi-robot están orientados a resolver problemas, en los cuales la participación de un solo robot no es suficiente o resulta ser muy costosa, en términos de diseño y de tiempo, por ejemplo el transporte de objetos voluminosos, el traslado de material peligroso, la exploración y cobertura de un terreno. Si tuviésemos un solo robot, este vendría a encargarse de realizar todas las tareas: reconocer el medio, enviar la información, tomar decisiones, controlar su movimiento. Al dividir las tareas entre varios equipos, el trabajo se realiza de una manera más rápida y eficiente.

Un fallo en el funcionamiento de un robot, puede ser subsanado por el resto del equipo. Por ejemplo, un sistema donde varios robots buscan una pelota en un recinto. Sí un robot pasa al lado de la pelota y no la detecta, puede un compañero suyo detectarla y resuelve el error.

Se puede entonces apreciar la complejidad al desarrollar un sistema de robótica cooperativa y las múltiples líneas de investigación requeridas al desarrollo del mismo.

Una aplicación de esta tecnología, se logra al apreciar a los robots trabajando en celdas de manufactura flexible, aplicaciones en áreas didácticas y de investigación como son los robots futbolistas.

En la actualidad la Robótica Cooperativa utiliza en su mayoría plataformas robóticas móviles, en las cuales se desempeñan diferentes tipos de aplicaciones como el transporte de objetos mediante distintas estrategias, desde el empuje, la sujeción y arrastre de los objetos. “Robo Cup” y las ligas de robot han incentivado al desarrollo de sistemas

cooperativos, al dividir sus torneos de fútbol en categorías, permitiendo avanzar en la robótica cooperativa distribuida.

El grupo de investigación en robótica móvil autónoma “ROMA” de Colombia está desarrollando e implementando una plataforma robótica cooperativa aplicada al transporte de objetos largos a través de un entorno estructurado utilizan las herramientas provenientes del kit Lego Mindstorms RCX 2.0.

La Universidad Católica de Chile está investigando en la detección y diagnóstico de “Fallas en Robots Móviles Cooperativos”, en la investigación la Universidad de Chile utilizó tres robots móviles diseñados por la universidad.

En la Universidad Jadavpur en Kolkata, India utilizan robots humanoides en su investigación del Comportamiento de robótica cooperativa, el objetivo del proyecto es lograr el desenvolvimiento de los robots Bioloid, mediante la locomoción de los mismos utilizando un control descentralizado. La plataforma es un grupo de tres robots Bioloid humanoides equipados con sensores de distancia y una cámara de color. Los robots se mueven hacia la meta manteniendo una formación triangular.

La Escuela Politécnica del Ejército en pro de estar a la vanguardia de nuevos avances a nivel de investigación, y considerando los avances realizados a nivel mundial, ha decidido apoyar los proyectos de investigación como es el caso del presente proyecto de Robótica Cooperativa, el cual pretende demostrar los nuevos conceptos y tendencias tecnológicas que circulan en el mundo.

## **1.2 JUSTIFICACIÓN E IMPORTANCIA**

Actualmente, en el área de la robótica, se está produciendo un gran desarrollo en el campo de los robots cooperativos. Algunos problemas son demasiado difíciles de resolver para un único robot: (empujar una caja, explorar un campo). La robótica se considera cooperativa cuando dos o más robots se encuentran desarrollando una tarea conjunta.

Este proyecto consiste en la definición de un modelo para el soporte de inteligencia cooperativa, en una comunidad conformada por miembros homogéneos, con las mismas capacidades y habilidades. Los miembros de esta comunidad son conocidos, en cuanto a sus destrezas de comunicación, etc.

El trabajar en robótica cooperativa tiene algunos problemas a resolver, entre ellos podemos citar los siguientes:

Interferencia entre robots, según el tipo de programación y de control de los distintos robots, la interferencia entre ellos puede llegar a ser un problema, golpes, choques, distracciones. Por ejemplo, si se tiene un equipo de robot y el objetivo es seguir una pelota, unos robots estorbarán el movimiento de los otros robots, o si los sensores no son muy “certeros”, pueden confundir otro robot con la pelota.

Coste de la comunicación, los robots deben mantener una comunicación fluida ya sea entre ellos, o con una unidad central, complicando así su diseño, aumenta el peso, el consumo de energía, y provoca interferencia con otros sistemas de comunicación o con los sensores.

Incertidumbre respecto a los otros robots, el sistema pierde eficacia cuando un robot realiza un trabajo ya elaborado por un compañero suyo, o al contrario, si deja de realizar una tarea pensando que esta ya fue realizada. Por ejemplo, un grupo de robots explorando un terreno, en el cual pueden pasar varias veces por un mismo sitio, o dejar alguna zona sin explorar.

Detrás de la definición se esconde un problema más complejo: ¿Cómo se organiza cada robot y su función para completar la tarea global?, se deberá decidir si la arquitectura es centralizada o distribuida, tomando en cuenta las ventajas y desventajas de cada tipo.

Se puede entonces entender a la robótica cooperativa como un campo en desarrollo, y sus aplicaciones pueden ir desde rutinas de trabajo diario en un hogar, o complejos sistemas industriales.

En la actualidad existen muy pocos experimentos implementados con plataformas robóticas reales.

### **1.3 ALCANCE**

El presente proyecto propone implementar un sistema de robótica cooperativa, basado en dos robots, equipados con sensores ultrasónicos, sensor de pistas, interruptor, y de rotación de motor “encoders”, para cubrir un espacio de trabajo máximo de 3[mts] cuadrados.

Este proyecto además consiste en la definición de un modelo adecuado para la implementación y el estudio de la robótica cooperativa, para el soporte de inteligencia cooperativa, en una comunidad conformada por miembros homogéneos, con similares capacidades y habilidades.

Los miembros de la comunidad son conocidos, en cuanto a sus habilidades, destrezas, medios de comunicación, etc.

Se desarrollará un sistema de software para la robótica cooperativa basado en librerías de software libre, y programación en C.

Los dos robots serán de características similares para poder combinar sus habilidades y se usará comunicación vía Bluetooth.

La aplicación para estas plataformas será simular el principio básico del fútbol este es transportar un balón desde un punto inicial hasta un final, a través de movimientos de los miembros de la comunidad (robots).

## **1.4 OBJETIVOS**

### **1.4.1 General**

Implementar un sistema de robótica cooperativa, basado en dos robots de características similares.

### **1.4.2 Específicos**

1. Acondicionar la plataforma robótica para comunicarse entre ellos, a través de controladores.
2. Desarrollar un sistema para la interacción de dos elementos robóticos.
3. Desarrollar un sistema capaz de implementar los algoritmos de cooperación, entre dos robots la con un objetivo en común.
4. Determinar el nivel de cooperación alcanzado, entre los dos robots.

## **1.5 DESCRIPCIÓN GENERAL DEL PROYECTO**

El presente proyecto pretende desarrollar un sistema de software capaz de implementar la cooperación entre dos robots, las cuales tendrán un objetivo en común,



además de sentar las bases para futuras investigaciones, al resolver problemas, como es mover una caja, o jugar fútbol.

El proyecto se desarrolló en 2 fases:

La primera fase es mover una caja una distancia  $l$  en una trayectoria lineal, observar y resolver los problemas de sincronización de los equipos, comunicación, y movilidad para lo cual se divide esta fase en 3 etapas a continuación se describe cada una:

Primera Etapa: Manipulación de las Librerías de C para manejar el controlador, pruebas de sensibilidad de los sensores y actuadores, construcción del modelo robótico y pruebas de funcionamiento

Segunda Etapa: Determinación de características operativas y sincronización de los robots para obtener movimientos lineales.

Tercera Etapa: Evaluación del desempeño del software para robótica cooperativa.

La segunda fase es la aplicación de la robótica cooperativa, para el proyecto se demostrará la cooperación de dos elementos robóticos, con un objetivo en común, mover un balón desde una posición cualquiera en el área de trabajo, hasta el final de la misma.

## CAPÍTULO II .

### MARCO TEÓRICO

#### INTRODUCCIÓN

En este capítulo se presenta una breve descripción de la historia, origen, ideas e influencias de la robótica cooperativa para el diseño de los sistemas multi-robots, se definen algunos conceptos, se presenta la clasificación de la robótica cooperativa. Además, se revisó las investigaciones y proyectos más representativos.

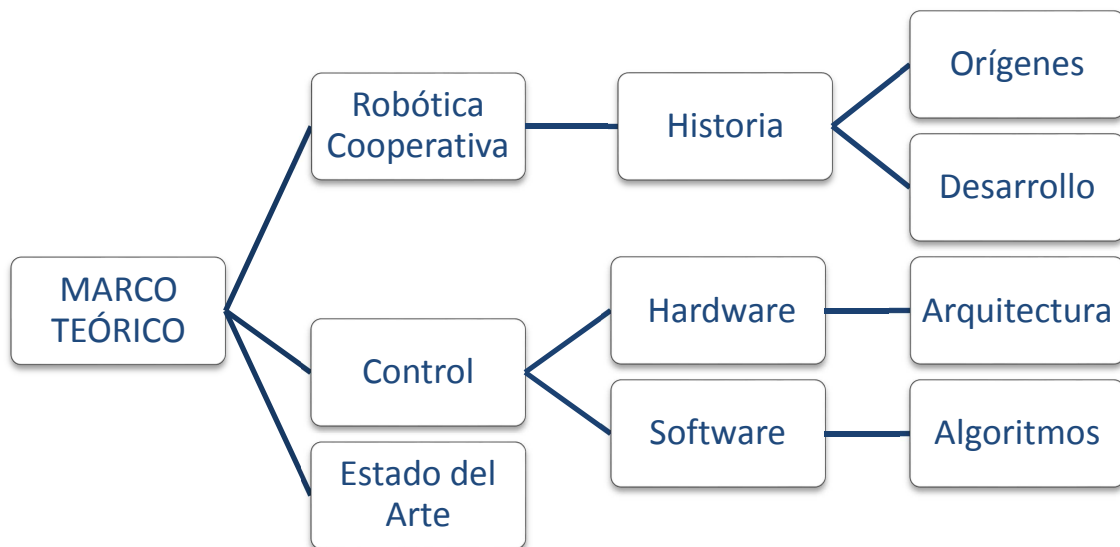


Figura II-1: Estructura del Capítulo II

### 2.1 ROBÓTICA COOPERATIVA

#### 2.1.1 Historia

En los últimos años, la tecnología ha ido creciendo y ampliando su interés en imitar comportamientos desarrollados en la naturaleza, es así como nace la robótica cooperativa o

cooperativa cuyo objetivo es diseñar y controlar sistemas capaces de resolver un problema conjuntamente, donde la intervención de un solo robot no es suficiente o resulta muy costosa. Los robots, elementos de estos sistemas cooperativos son en general simples de diseñar y controlar, generalmente homogéneos permitiendo de esta forma una reducción en tiempo y costos, por ejemplo en el transporte de objetos voluminosos, la manipulación de material peligroso, etc.

### **2.1.2 Origen**

Los etólogos distinguen entre sociedades y agrupaciones de animales. Aunque ambos términos refieren a un grupo de individuos de la misma especie, el término sociedad se refiere a individuos organizados, los cuales presentan conductas cooperativas, mientras el término agrupación define a individuos reunidos espacialmente, los cuales no necesariamente están organizados o involucrados en conductas cooperativas (Jiménez Builes, Ovelle Carranza, & Branch Bedoya, 2008).

Los animales forman sociedades y agrupaciones para trascender sus capacidades individuales. Las agrupaciones en particular, proveen protección contra depredadores o aumentan la eficiencia de la depredación, (Camazine, Deneuborg, Sneyd, & Bonabeau, 2001). Es decir al agruparse individuos de la misma especie pueden aumentar la eficiencia y ahorrar energía gracias a la sincronización de sus movimientos individuales.

De allí nace el concepto de robótica cooperativa con el esquema de control robótico basado en el comportamiento, este modelo es un instructivo, examina las características sociales de los insectos y animales y aplica estos hallazgos al diseño de sistemas de multi-robots.

La aplicación más común de este conocimiento es en el uso de reglas simples de control local de varias sociedades biológicas, -especialmente las hormigas, abejas y pájaros desarrolló comportamientos similares en sistemas de robots cooperativos.

El trabajo en este punto ha demostrado la habilidad de grupos de multi-robots para acudir en masa, dispersarse, agregarse, buscar y seguir huellas (Palacios García, 2009).

### 2.1.3 Desarrollo

Marvin Minsky propuso, en su libro “La Sociedad de la Mente” escrito en el año 1985, *“La mente es una sociedad constituida por la interacción de infinidad de diminutos agentes, los cuales carecen, a su vez, de toda actividad mental”*.

Después de los primeros estudios en los años 80 de la robótica y posteriormente de la robótica industrial, el ser humano se ve con la necesidad de crear máquinas capaces de realizar tareas tediosas o repetitivas, como mover cajas, explorar campos, etcétera, con el objetivo de facilitar dichos trabajos, buscando formas para reducir costos, comenzaron por encontrar un modelo en el cual más elementos trabajan para un mismo fin, he aquí donde se desarrolla la robótica cooperativa o colectiva.

### 2.1.4 Definición

De las diferentes investigaciones del concepto de robótica cooperativa, se obtiene como resultado la siguiente definición:

La robótica cooperativa es un grupo de robots interactuando entre sí, no necesariamente se deben colaborar explícitamente para realizar una tarea. En robótica cooperativa, la solución de un problema está ligada a un concepto conocido como inteligencia cooperativa, es decir la forma de solucionar el problema surge del comportamiento como grupo. Esta inteligencia cooperativa hace al sistema más robusto, al existir varios individuos, estos se pueden repartir las tareas, y en caso de fallar uno de ellos el otro puede tomar su lugar. Por lo tanto se pueden tener individuos no tan confiables realizando una tarea con una gran confiabilidad.

El concepto descrito anteriormente se lo utilizará para la demostración de robótica cooperativa.

En robótica cooperativa, cada individuo tiene su propia inteligencia pero además tienen la capacidad de comunicación y por lo tanto de cooperación y coordinación, de esta manera se tiene la inteligencia cooperativa (integración de la inteligencia de cada individuo más su interacción).

La robótica cooperativa, puede tener dos tipos de comportamiento, el de un grupo de robots sin inteligencia y sin comunicación entre sí, pero realizando una tarea específica, y el otro grupo de agentes cooperativos, los cuales poseen comunicación, dependiendo del

hardware pueden lograr tener una inteligencia artificial avanzada. Ambos tipos de comportamientos son basados en modelos de la vida real. (Romero Ternero, 2009)

## 2.1.5 Clasificación por nivel de comunicación de los sistemas de Robótica

### Cooperativa

Dependiendo al nivel de comunicación e interacción entre los distintos robots de un sistema se los puede clasificar en:

- Muy Acoplados
- Poco Acoplados

#### 2.1.5.1 Muy acoplados(o colectivos)

Todos los individuos tienen su propia inteligencia además de la capacidad de comunicación, por lo tanto, cooperación y coordinación entre sí, se puede decir a la inteligencia cooperativa como la suma de las inteligencias de cada uno de los individuos. Cada robot debe tener la capacidad de transferir información a otro compañero, y además tener la suficiente inteligencia para pedir ayuda, y coordinar sus movimientos con los otros individuos.

Por ejemplo, cuando dos robots intentan mover una caja empujándola de uno de sus lados manteniéndola en línea recta como se observa en la Figura II 2: Sistema Multi – Robot Muy Acoplado. Los dos robots deben sincronizar sus velocidades, evitando de este modo el movimiento inadecuado de un lado a otro de la caja. Son sistemas poco robustos, si falla un robot, falla todo el sistema.

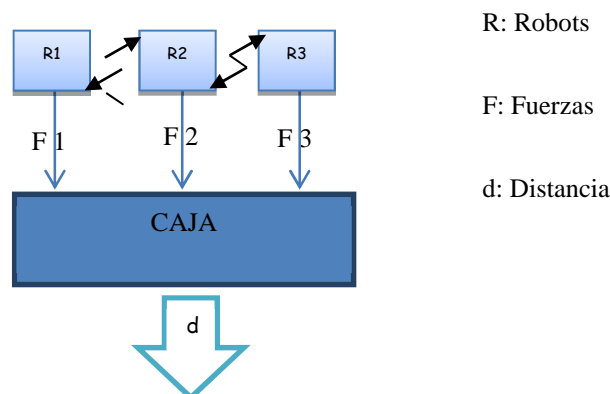


Figura II-2: Sistema Multi – Robot Muy Acoplado

### 2.1.5.2 Pocos acoplados (o manadas)

Los robots en este tipo de sistema funcionan independientemente el uno del otro, quiere decir: no se toma en cuenta los movimientos o decisiones de los demás. No hay una inteligencia central, al contrario todo el control es distribuido, cada elemento simplemente sigue unas reglas.

Este tipo de comportamiento trata de imitar sistemas biológicos como el de los insectos sociales. Como se observa en la Figura II 3: Ejemplo de Sistema Multi –Robot Poco Acoplado, se tiene un grupo de robots, para recoger las pelotas de un campo de tenis una vez acabados los entrenamientos, las lleven a una esquina del campo. Cada robot buscará una pelota, la detectará, recogerá, y trasladará al lugar adecuado. Para éste trabajo no es necesario la comunicación entre los robots. (Jiménez Builes, Ovelle Carranza, & Branch Bedoya, 2008)

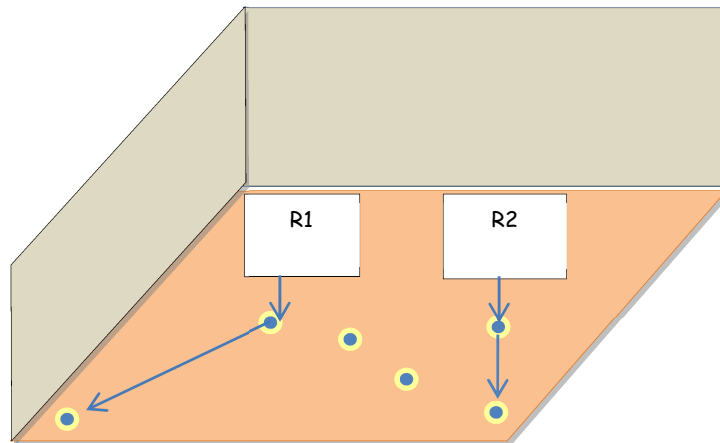


Figura II-3: Ejemplo de Sistema Multi –Robot Poco Acoplado

### 2.1.6 Ventajas y Desventajas

En la tabla II-1 se muestra un resumen de las ventajas y desventajas de la robótica cooperativa, recopiladas de las numerosas investigaciones encontradas respecto al tema:

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>• Mejora el rendimiento del grupo de trabajo gracias al ahorro de recursos así como de energía.</li> <li>• Al trabajar con varios robots el sistema se vuelve más productivo y se puede realizar cosas que un solo robot no lo haría.</li> <li>• Los multi-robot permiten un sensamiento distribuido.</li> <li>• Este sistema permite distribuir tareas y reducir costos en la operación.</li> <li>• Es un sistema más robusto por esta razón se tiene tolerancia a fallas, si un solo robot no cuenta con las capacidades para concluir su tarea, otro robot del grupo puede reemplazar.</li> <li>• Al ser un sistema multi-robot, no se busca crear un solo robot con la capacidad de realizar una tarea, en la cual se consume más energía, mayor procesamiento de datos, etc. Con un sistema multi robot se puede ahorrar recursos.</li> </ul>	<ul style="list-style-type: none"> <li>• Al trabajar con varios robots puede volverse complejo debido a la necesidad de sincronizar los robots, y establecer una comunicación confiable para evitar fallos en el sistema.</li> <li>• En el área de trabajo, los equipos pueden convertirse en obstáculos o confundirse y en lugar de cooperar existiría una competencia.</li> <li>• Establecer una buena comunicación en algunos casos se vuelve costoso, aumentando así el precio final del proyecto.</li> <li>• El sistema deja de ser eficiente si varios robots realizan la misma tarea varias veces, por ejemplo si se explora un campo y varios robots examinan el mismo tramo o a su vez algún tramo quede sin explorar, al confiarse que el otro robot lo ha hecho.</li> </ul>

**Tabla II-1: Ventajas y Desventajas Robótica Cooperativa**

### 2.1.7 Aplicaciones

En la actualidad el tema de la robótica cooperativa se ha extendido, una de las áreas donde más se estudia este tema es en la educación, como por ejemplo el tema de este proyecto, donde su objetivo es formar un precedente en la universidad para futuras investigaciones, a continuación se presentan algunos proyectos en los cuales se está trabajando actualmente, para una referencia más amplia diríjase al Anexo 4:

- En la universidad Pontificia Universitaria Javeriana de Colombia se está desarrollando una plataforma multi – robot en tecnología Quemes constituida por tres componentes: un conjunto de robots móviles, un entorno de programación gráfica y un esquema educativo progresivo de robótica cooperativa.

- El ingeniero Enrique González generó conocimientos y herramientas para el desarrollo de aplicaciones basadas en agentes, con las cuales construyó el equipo de robots futbolistas para participar en las competencias internacionales (Robo Cup).

## **2.2 CONTROL EN ROBÓTICA COOPERATIVA**

El control de un sistema de robótica cooperativa se lo puede realizar en base a 2 tipos:

- Control en base a hardware
- Control en base a Software

### **2.2.1 Hardware**

Se utilizan diferentes tipos de arquitecturas para realizar el control de robótica cooperativa mediante hardware, las más utilizadas son:

- Arquitectura Centralizada
- Arquitectura Distribuida

La selección de la arquitectura adecuada para el proyecto depende directamente de las necesidades de los objetivos planteados a continuación se hará una breve descripción de las arquitecturas, este tema se lo profundizará en el Capítulo 3.

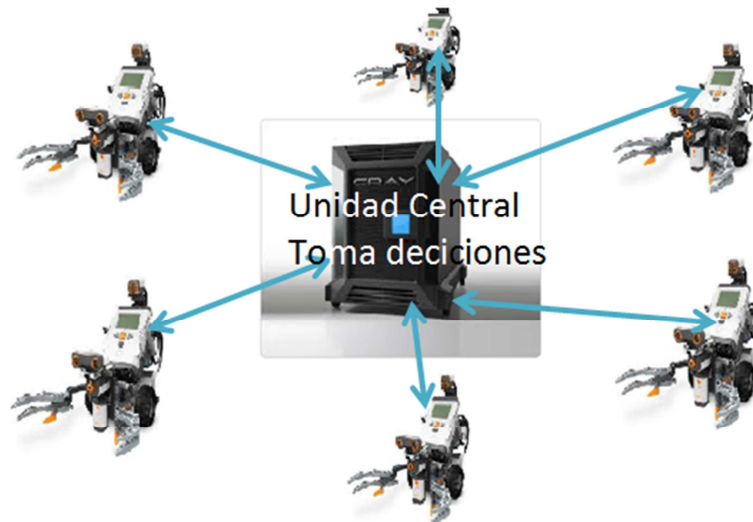
#### **2.2.1.1 Arquitectura Centralizada**

Está constituida de varios robots dependiendo de una unidad central la cual envía el objetivo a realizar. Los mismos adquieren datos a través de sus sensores (por ejemplo al reconocen un terreno mediante sensores ultrasónicos, persiguen una pelota mediante cámaras), y toda la información se la envían a la unidad central, la encargada de tomar las decisiones. La principal ventaja de éste sistema, es su unidad central la misma, almacena mucha información, permitiendo ser más preciso el sistema al momento de tomar una decisión, también es eficiente a la hora de obtener soluciones. Como desventajas. La unidad central estará recibiendo continuamente datos de los otros robots, pudiéndose provocar una congestión al recibir gran cantidad de datos, y ralentizarse la toma de decisiones. Además, si la unidad central falla, o deja de funcionar, el sistema también deja de funcionar, al no ser capaz los robots de tomar decisiones solos.



Por la tanto la unidad central es el cuello de botella del sistema, debido a la recepción, tratamiento y envío de la información, el tratamiento de datos en el sistema depende de la velocidad de respuesta del sistema, la recepción de información esta limita al tamaño del equipo y al rango de acción.

A continuación se muestra un ejemplo donde los robots lego depende de la unidad central para moverse en este caso es una computadora al tener un procesador rápido en la adquisición de datos, permitiendo un sistema más eficiente, pero existen sistemas centralizados con un robot central como líder el mismo analiza y toma las decisiones a realizar por el resto de la manada como se observa en la naturaleza, en la Figura II 4: Arquitectura Centralizada con Robots Lego, se observa una arquitectura centralizada con robots legos los mismos organizan objetos según les orden la unidad central.



**Figura II-4: Arquitectura Centralizada con Robots Lego**

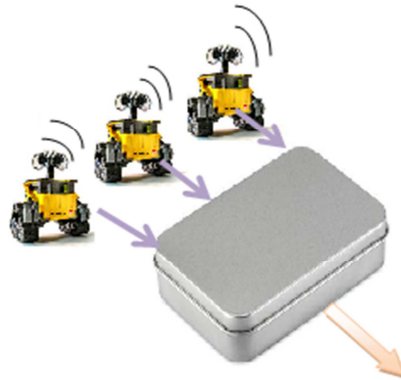
### 2.2.1.2 Arquitectura Distribuida

Está constituida con varios robots, cada robot decide cómo actuar mediante sus percepciones del ambiente, sin recibir órdenes de ninguna entidad central. Con éste tipo de arquitectura el sistema es más dinámico y rápido, no es necesario el envío, y el almacenamiento de información en ninguna entidad. Su principal ventaja es:

Sí deja de funcionar un robot, todos los demás siguen funcionando independientemente. Y su desventaja es la dificultad para diseñar los algoritmos de programación, aunque son muy simples, deben de coordinar las acciones a realizar en los robots sin equivocaciones en el robot a realizar la acción esperada.

Con una arquitectura distribuida se evitan varios problemas obtenidos con la anterior arquitectura mencionada, los robots son autónomos se evita el cuello de botella, el tamaño de los equipos teóricamente son ilimitados y escalables, de esta forma también se soluciona el problema del rango de acción, el sistema es más robusto, cada robot sabe la tarea a realizar si uno falla el sistema no falla, se puede enviar a otro robot para realizar la tarea. La autonomía de los robots es una ventaja indispensable para evitar fallos.

A continuación se describe un ejemplo de una arquitectura distribuida, en donde cada robot decide su acción a realizar de acuerdo a su entorno, en este sistema la comunicación juega un papel muy importante, y su debilidad es si la comunicación falla, al no tener la misma percepción del ambiente, se observa en la Figura II 5: Arquitectura Distribuida, implementada con robots móviles, su objetivo es mover una caja en línea recta.



**Figura II-5: Arquitectura Distribuida**

## **2.2.2 Software**

El control en robótica cooperativa se lo realizar mediante software utilizando diferentes tipos de algoritmos, basados en sistemas multi agentes, de acuerdo a las necesidades de los objetivos planteados a continuación se hará una breve descripción de agentes por ser la base del algoritmo empleado, este tema se lo profundizará en el Capítulo 4.

### **2.2.2.1 Agente**

Agente, del latín agere, significa “el que hace”, son capaces de percibir los cambios en el ambiente, procesa las percepciones a través de sensores y responde de manera racional mediante los actuadores, reaccionan a un estímulo realizando una acción (Russell & Norvig, 2004).

En general un agente simple se describe en la Figura II 6: Agente Simple, el agente procesa por medio de sensores los cambios en el ambiente y responde a través de acciones.

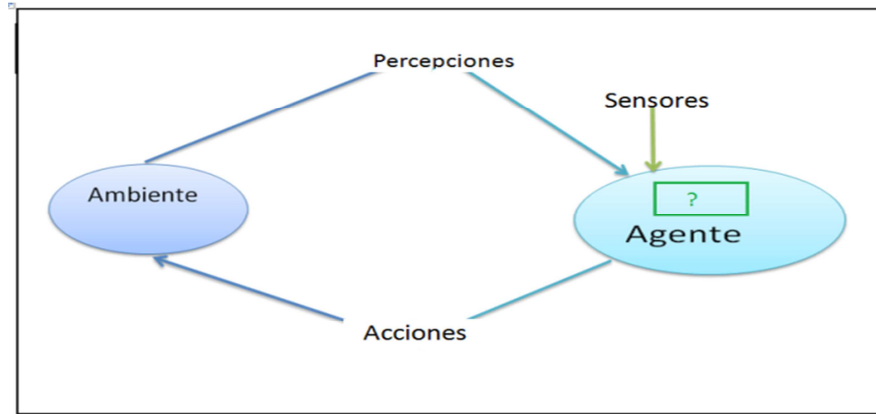


Figura II-6: Agente Simple

### 2.2.2.1.1 Características de los agentes

En la Tabla II 2: Características de los Agentes, se detalla cada una de ellas.

Característica	Descripción
Autonomía	Los agentes actúan sin intervención humana directa o de otros agentes y tienen alguna clase de control sobre sus acciones y estado interno. El software tradicional se ejecuta en entornos interactivos, donde responde a órdenes directas del usuario.
Reactividad	Percibe el entorno en el cual está inmerso y responde de manera oportuna a los cambios ocurridos en él lugar (para actuar adecuadamente un agente debe poder conocer en todo momento el mundo alrededor de él).
Iniciativa (pro actividad)	Debe tener un carácter emprendedor y tomar la iniciativa para actuar guiado por los objetivos a satisfacer. En cada momento el agente decide la acción a llevar a cabo. No sólo actúa en función a los estímulos percibidos sino también realiza acciones como resultado de sus decisiones.
Sociabilidad	Capacidad de interactuar con otros agentes (incluso humanos) utilizando alguna clase de lenguaje de comunicación de agentes. Los agentes colaboran entre sí para la ejecución de tareas (MAS).

Tabla II-2: Características de los Agentes

### 2.2.2.1.2 Clasificación de los Agentes

Los agentes se pueden clasificar de dos tipos:

- Grado de Inteligencia
- Tipo de programación

#### 2.2.2.1.2.1 Clasificación de los agentes por el grado de inteligencia

Esta clasificación está dada por el nivel de inteligencia de los agentes la cual permite la interacción con el medio ambiente se detalla en la Tabla II 3: Clasificación de los agentes por su inteligencia.

Clasificación	Características
Agente Inteligente	Capaz de Interactuar en un entorno de forma razonada, la toma de decisiones depende de su conocimiento del entorno
Agente Simple	Sus decisiones depende de las percepciones del medio ambiente, si las condiciones cambian no sabría qué decisión tomar
Agente Reactivos	Responde a estímulos sobre el entorno que le rodea.
Agentes de Interfaz	Son útiles para ayudar a los usuarios a conocer una aplicaciones en particulares, el usuario no conoce los procesos que lleva acabo solo los resultados.

**Tabla II-3: Clasificación de los agentes por su inteligencia**

A continuación se describe los agentes inteligentes al ser los más relevantes de la clasificación descrita anteriormente.

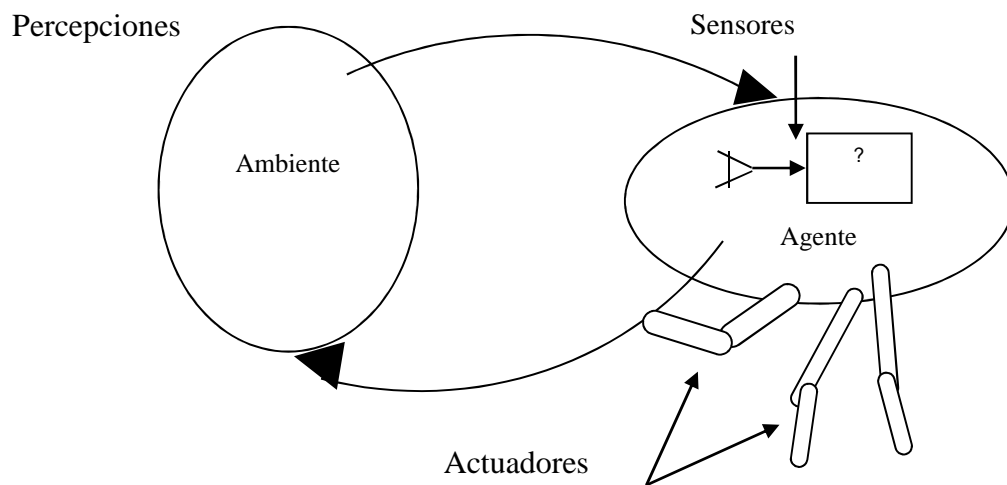
### 2.2.2.1.2.1.1 Agente Inteligente

El agente inteligente se puede definir como una entidad capaz de interactúa en un entorno, de forma razonada e independiente, actuando correctamente y tendiendo a maximizar el resultado, se describe en la Figura II 7: Agente Inteligente, teniendo actualmente muchas aplicaciones como robótica, correos electrónicos, programas informáticos,..., Es por eso qué dar una definición exacta es imposible a continuación se enuncian algunas :

(Hípola & Vargas Quesada , 2013) *“Entidad software que, basándose en su propio conocimiento, realiza un conjunto de operaciones destinadas a satisfacer las necesidades de un usuario o de otro programa, bien por iniciativa propia o porque alguno de éstos lo requiere”.*

(Mikelecý, Lisková, & Cech, 2009) *“Programas de ordenador capaces de efectuar una tarea o actividad sin la manipulación directa de un usuario humano.”*

(Jennings & Wooldrigde, 2002) *“Es un sistema situado en un determinado entorno capaz de actuar de forma autónoma y razonada, en dicho entorno para llevar a cabo sus objetivos”*



**Figura II-7: Agente Inteligente**

### 2.2.2.1.2.1.2 Características de un agente inteligente

Permiten entender la interacción de los agentes con el ambiente y con otros miembros de los agentes, las agentes tienen características internas y externas, Tabla II 4: Características de un agente inteligente se las describe a continuación. (Larios, 2004)

<b>Características Internas</b>	
Razonamiento	Permite interpretar acciones, resolver problemas, realizar inferencias y determinar soluciones.
Decisión	Mediante una secuencia de percepciones recibidas hasta ese momento a través de sensores, el agente decide su acción para cumplir su meta.
Control	Envían acciones para cumplir con la meta, interactuando en el ambiente a través de actuadores.
<b>Características Externas</b>	
Percepción	A través de sensores (infrarrojo, color, ultrasónico,...) puede conocer las condiciones en el ambiente en cualquier instante.
Acción	A través de actuadores (motor, foco,...) modifican su interacción con el ambiente
Comunicación	Permite interactuar con otros agentes, y conocer el entorno y los problemas de los demás agentes.

**Tabla II-4: Características de un agente inteligente**

### 2.2.2.1.2.2 Clasificación de los agentes por el tipo de programación

Esta clasificación nos permitirá diferenciar las opciones de programación de un agente, las cuales permite el mapeo para pasar de las percepciones a acciones, en la Tabla II 5: Clasificación de los agentes por tipo de programación se muestra las diferentes características.

Clasificación	Características
Agente Reactivo Simple	Es el agente más sencillo, seleccionan sus acciones en base a las percepciones, ignora las percepciones anteriores
Agente Basado en Modelos	Este agente guarda sus percepciones anteriores para la toma de decisiones.
Agente Basado en Metas	Necesita tener información sobre su meta, y el estado del medio ambiente.
Agente Basado en Utilidades	Utilizan un modelo del entorno, una función de utilidad en el entorno, para seleccionar la acción que le lleve a generar el mayor grado de utilidad.

**Tabla II-5: Clasificación de los agentes por tipo de programación**

Se describe a continuación los agentes basados en metas al ser estos los utilizados en el siguiente proyecto.

#### 2.2.2.1.2.2.1 Agentes Basado en Metas

Los agentes basados en metas, permiten alcanzar objetivos, mediante la reestructuración de su planificación, cuando las condiciones en el medio ambiente cambian. La búsqueda y la planificación, son los sub campos de la inteligencia artificial dedicados a encontrar las secuencias de acciones para alcanzar las metas de un agente. (Roa, Guitierrez, & Stegmayer, 2008)

Para un agente basado en metas, conocer el estado del medio ambiente no es suficientes para decidir la acción a tomar, necesita también saber cuál es su objetivo a

realizar, un ejemplo práctico, un robot detecta un objeto delante de él, tiene varias posibilidades de acciones: esquivarlo, chocarse con él, o detenerse un centímetros antes, la acción correcta a realizar debe estar ligada con la meta a cumplir, si un robot está diseñado para esquivar objetos, el mismo no debe chocar con el objeto. Este tipo de agentes es eficiente cuando cumpla sus metas, puede resultar sencillo cuando los objetivos se resuelven en una sola acción, pero será difícil cuando los agentes consideren secuencias largas para llegar a su objetivo, dependen mucho de la planificación de sus metas para obtener la eficiencia deseada.

En la Figura II-8: Agente Basado en Metas se describe el funcionamiento de un agente basado en metas, el ambiente cumple un papel importante al tomar las decisiones a ejecutar.

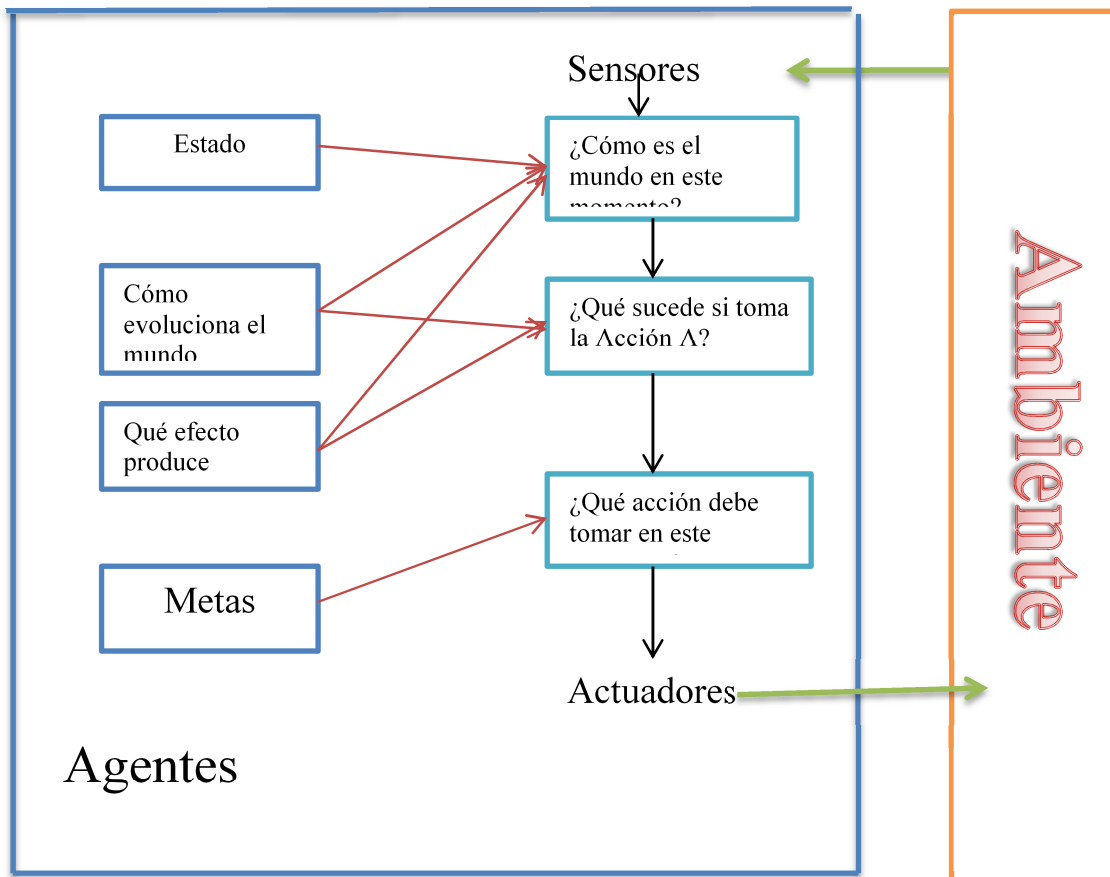


Figura II-8: Agente Basado en Metas



### 2.2.2.1.2.2.2 Característica de un Agentes Basado en Metas

En la Tabla II 6: Características de los Agentes Basados en Metas, descrita a continuación se enuncia las principales características.

Característica	Función
Racionalidad	Un agente actúa en función de lograr sus metas
Veracidad	La información de su entorno siempre debe es verdadera
Habilidad Social	Interacción con otros agentes

**Tabla II-6: Características de los Agentes Basados en Metas**

### 2.2.2.2 Sistema Multi Agentes (SMA)

Un SMA es un conjunto organizado de agentes interactuando de forma cooperativa para lograr de manera colaborativa un objetivo global. (Sayde Alcántara, Fernández , & Ruiz Rodríguez, 2008)

Este sistema permite a los agentes con características especiales (como los agentes basados en metas), interactúen con otros agentes de similares características, siempre y cuando el agente pueda comunicarse con los demás agentes. Se debe considerar los cambios de la evolución del medio ambiente como los muestra Bussmann y Demazeau.

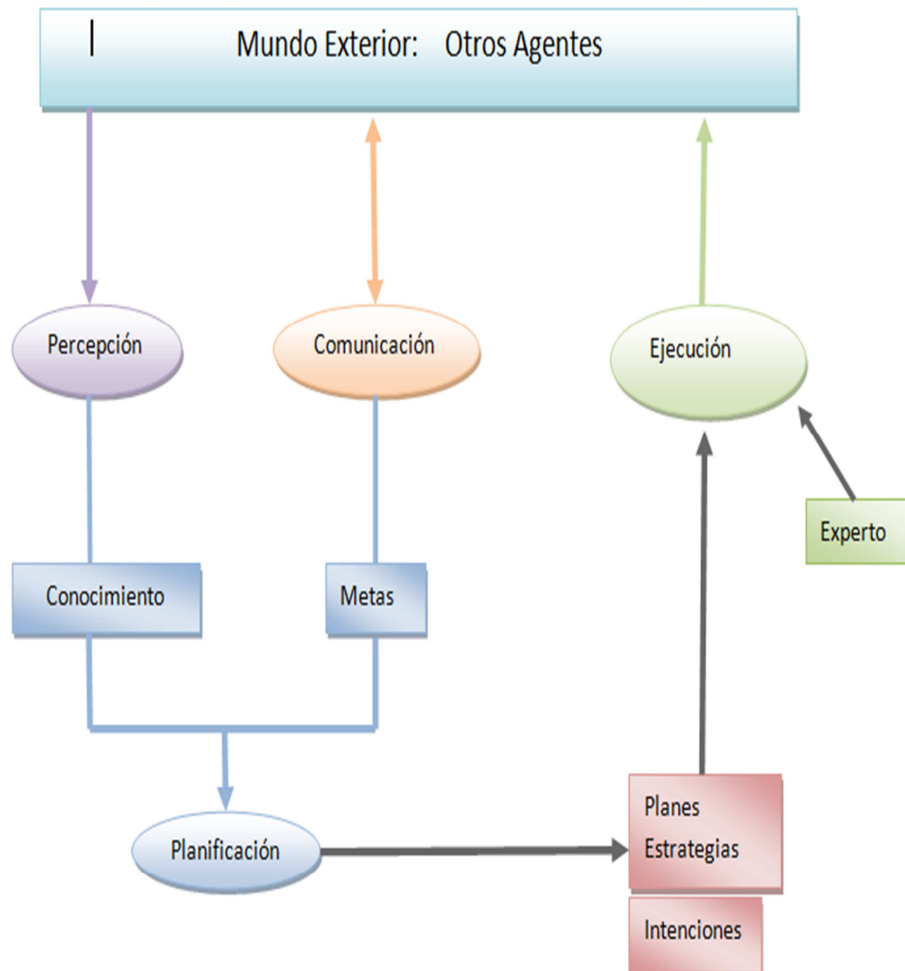
Un agente debe ser capaz de manipular explícitamente su conocimiento sobre el ambiente.

La percepción y la comunicación debe ser lo suficientemente buena como para mantener una visión coherente del estado del sistema.

La toma de decisiones y ejecución de decisiones debe realizar se en el tiempo.

Identificación de eventos imprevistos debe ser seguida por una evaluación de los cambios internos.

En la Figura II 9: Funciones de un Sistema Multi agentes (SMA), se observa a un sistema de agentes interactuando con el medio ambiente, mediante la percepción y la comunicación se puede establecer las metas a lograr por el grupo de agentes.



**Figura II-9: Funciones de un Sistema Multi agentes (SMA)**

➤ **Planificación**

Son las secuencias de acciones para alcanzar un determinado objetivo.

El proceso de planeación involucra el ambiente en su estado inicial y las acciones a ejecutar en el mismo, la planeación solo es efectiva en entornos predecibles. (Julián & Botti, 2010)

➤ **Colaboración**

Los SMA colaborativos ahorran en tiempo de ejecución de tareas y recursos, al utilizar los agentes disponibles en el sistema, obteniendo un sistema confiable y preciso,

para definir de forma clara la colaboración en un SMA se describen los siguientes: repartición de tareas y compartir resultados.

➤ **Repartición de tareas.**

Se basa en la descomposición de una tarea a cargo del agente, cuando este se ve sobrecargado por la misma y repartir la tarea a otros agentes con habilidades necesarias para resolver el problema. Cuando los agentes terminen su tarea el resultado es devuelto al agente encargado de la descomposición, el mismo encargado de unir, y enviar el resultado final solicitado, la asignación de tareas se las puede realizar con agentes homogéneos o heterogéneos de la siguiente manera:

Los SMA creados con agentes homogéneos, la asignación de tareas es muy sencilla, se reparte las tareas aleatoriamente a los agentes inactivos del SMA y disponibles.

Los SMA creados con agentes heterogéneos, la asignación de tareas será compleja, al ser necesario primero evaluar a los agentes, para determinar quién tiene las habilidades físicas para cumplir con la meta y se reparte la tarea de acuerdo a la información del sistema sobre las habilidades físicas de los agentes, para encontrar el agente idóneo para realizar la tarea.

➤ **Compartir resultados.**

Los agentes deben tener el conocimiento y representación del entorno, ejecutan una misma tarea varios agentes, se comparten los resultados parciales, con el objetivo de poder encontrar la solución global a través de la cooperación de los diferentes agentes.

➤ **Coordinación**

Un SMA debe ponerse de acuerdo con otros agentes para cumplir sus objetivos mediante la coordinación, obteniendo mayor precisión en la resolución de tareas, se tiene tres tipos de coordinación a continuación se describen.

**Coordinación Global.** El SMA determina y planifica globalmente las acciones de los diferentes agentes.

**Coordinación Individual.** El SMA tiene completa autonomía, cada agente decide qué hacer y resuelve localmente los conflictos incluso los detectados con otros agentes.

**Coordinación orientada por la Cooperación.** El SMA se coordina en el momento de ejecutar acciones

### 2.3 ESTADO DEL ARTE

La robótica cooperativa, con los avances tecnológicos está generando nuevas investigaciones, así muchos grupos de investigación y universidades como la ESPE han comenzado a desarrollar proyectos sobre el tema, con diferentes tipos de plataformas.

A nivel mundial se encuentra desarrollando diferentes proyectos de investigación de la robótica cooperativa, en el año 2007, la Universidad Autónoma de Barcelona, España crea un modelo de un grupo de robots colaboradores cazadores, la problemática de la investigación es inmovilizar una pelota a través de un sistema de robots cooperados, usan robots Educativos Bae Bot de Parrallax (J.García, 2007), las investigaciones avanza y en el año 2010 la Universidad Santo Tomas Colombia, desarrolló un estructura entre 2 robots los mismos cuentan con sistemas sensoriales propios, navegación de forma autónoma y cooperativa, el objetivo del grupo es encontrar objetos en un área de trabajo, una vez concluida la tarea entran en estado de reposo, usan robots de su propia creación (A.Correa, Sanabria, & Sandoval, 2010), en la Universidad Nacional de la Plata, Argentina desarrollaron un modelo de ciclo de vida para el aprendizaje basado en compartición de conocimientos en sistemas autónomos de robots, el sistema autónomo robótico aprende de la naturaleza y es capaz de colaborar con otros robots autónomos (Salvador, 2010), en la actualidad se está realizando un proyecto en el Centro de Investigación Robótica Móvil (ROMA) Colombia, desarrollan un algoritmo para la coordinación de un sistema multi robots cooperativos para tareas de búsqueda de fuentes de calor en entornos dinámicos, la plataforma está siendo diseñada con el fin de salvar vidas humanas en el menor tiempo posible durante un siniestro (explosiones, ...) su principal tarea sería búsqueda y rescate, a medida del avance del proyecto se están presentando las siguientes limitaciones: la dificultad de acceder a lugares reducidos, mover escombros y la falta de visibilidad cuando existe mucho polvo el proyecto se encuentra en la fase de implementación (Beltrán, 2012).

Desde el comienzo de la robótica cooperativa se ha ido buscando formas, métodos y algoritmos para alcanzar un mejor desarrollo de las plataformas es así como se implementa la programación en base a agentes, en el año 2006, en la Universidad de Paul Sabater, Francia se crea una simulación robótica, donde se desconoce su tarea principal y el medio ambiente, el problema a resolver la búsqueda, colocación de cajas y la colaboración en el

área determinada, el método usado es ADELFE es un modelo de auto organización adaptativa (Picard, 2006), en el Instituto Nacional de Astrofísica Óptica y Electrónica (INAOE), México 2007 se investigó sobre los requisitos del Movimiento Colectivo Coordinado en un Sistema Multi-Agente enfocándose en las sociedades de animales, cada agente identifica su situación por sus propios medios, y determina en consecuencia la tarea a realizar como en un enjambre de abejas teniendo en cuenta los líderes y objetivos, se pretende la interacción en una área de trabajo con varios robots permaneciendo en una formación determinada y sean capaces de eludir objetos, y luego regresen a su formación inicial, en el mismo año en la Universidad de Fribourg, Suiza se estaba investigando Agentes Autónomos y Robótica Cooperativa colocando el problema más común a resolver en un área de trabajo se busca objetos y se los apila en un mismo sitio, la solución de la misma fue innovadora se pretendió dar a los agentes autonomía el poder de decidir qué acción tomar sobre sus propias percepciones convirtiéndolo en un sistema flexible, tolerante a fallos usaron robots Kephers obteniendo resultados útiles a los objetivos planteados (Chantemargue, Courant, & Hirsbrunner, 2007), la robótica cooperativa y la teoría de agentes avanza, es así como en el año 2011 en España, la Universidad de Alicante implementa una arquitectura de agentes para robótica de enjambres utilizando JADE, es un sistema completamente descentralizado a través de multi agentes en varios niveles su implementación es un sistema completamente descentralizado a través de un sistema multi agentes en varios niveles donde el primer nivel son agentes básicos, luego tienen agentes intermedios de alto nivel encargados de controlar el enjambre y dirigir a los objetivos deciden la tarea a realizar, tiene una alta tolerancia a fallos con sistemas de control permitiendo la agregación de nuevos agentes ante el falló de uno de ellos (Sempere, Aznar, Pujol, & Rizo, 2011).

Estando a la vanguardia de las investigaciones a nivel mundial, la Escuela Politécnica del Ejército apoya en la investigación sobre el tema de robótica cooperativa, así el presente proyecto pretende implementar un sistema multi-robot, en el cual los elementos robóticos constaran de similares características en hardware, y en el software se utilizará programación basada en agentes por metas, con el objetivo de realizar una aplicación de robótica cooperativa.

## CAPÍTULO III .

### DISEÑO DEL HARDWARE

En este capítulo se presenta la arquitectura del robot necesarias para resolver la problemática descrita en el capítulo 1, se presenta un sistema distribuido este permite demostrar la cooperación entre los 2 elementos robóticos, a su vez se describe los diferentes dispositivos utilizados en los robots, el controlador, los sensores, comunicaciones, actuadores.

#### 3.1 SISTEMA MULTI – ROBOT HARDWARE

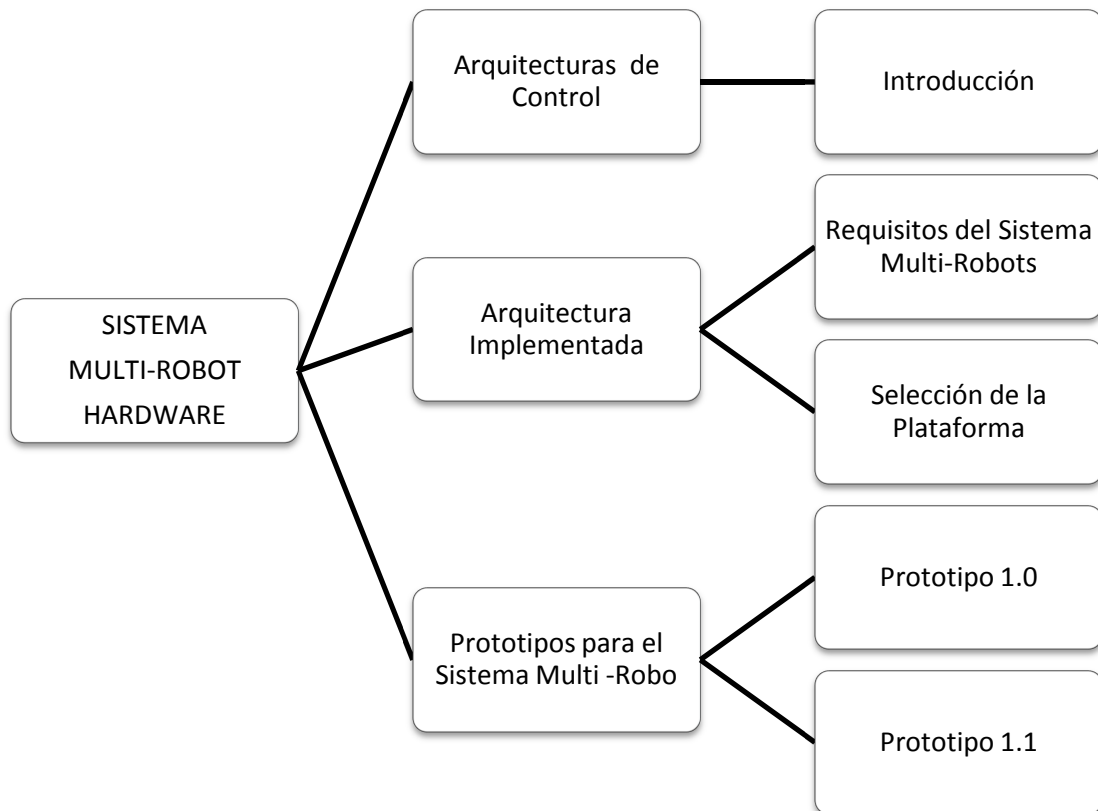


Figura III-1: Estructura del Capítulo III

### 3.1.1 Arquitectura De Control Distribuido

#### 3.1.1.1 Introducción

Las arquitecturas de control, dan un comportamiento inteligentes a los agentes físicos, la arquitectura de control distribuido consta de varios robots, los mismos deciden la tarea a realizar dependiendo de su programación y el medio que los rodea, esta arquitectura es dinámica y tolerante a fallos, si uno robots no puede cumplir con la tarea especificada, se llama a otro, de esta forma el sistema se vuelve más robusto y los elementos del mismo son independientes, para poder interactuar con inteligencia robótica. De las distintas investigaciones en este campo llegaron a la siguiente conclusión: los robots físicos deben estar dotados de tres primitivas básicas, son consideradas las capacidades de un agente físico, a continuación se describen:

- **Capacidad De Percibir:** Debe ser capaz de obtener información del entorno donde se encuentra, sin importar la presión con la cual esta fue adquirida.
- **Capacidad De Razonar:** Un agente robótico debe ser capaz de determinar la acción a realizar, con la información previamente adquirida.
- **Capacidad De Acción:** Un agente físico debe ser capaz de ejecutar la acción previamente seleccionada de forma precisa.

En la Figura III-2 : Disposición Básica de las Primitivas de Control, se muestra el esquema de las primitivas básicas de control.



**Figura III-2 : Disposición Básica de las Primitivas de Control**

Las primitivas de control se les pueden considerar como las fases del proceso de razonamiento, es posible construir un modelo, permitiendo a un agente físico interactuar en mayor o menor medida con el medio donde se encuentra dependiendo de dos factores descritos a continuación:

- La relación establecida entre las entradas y salidas de las primitivas de control, esta relación se refiere al orden y la forma con la cual se ejecutan las primitivas (secuencial, paralelo, basada en eventos, etc.).
- Mecanismos de procesamiento y distribución de información sensorial a través del sistema de control. En los paradigmas, la información sensorial se utiliza de forma específica, en la única función del agente, y en otros para la construcción de modelos virtuales del entorno y de forma general por el sistema de control para tomar decisiones.

La aplicación de estos factores da como resultado modelos de razonamiento, los mismos se dividen en dos grupos:

El primero grupo se basa en la disposición de las fases de razonamiento de manera vertical, solo una de las capas tiene acceso a los datos obtenidos a través de los sistemas de percepción del robot y por lo tanto cada una de las capas es responsable de forma individual de la salida y entrada de la capa siguiente.

En el otro grupo se disponen las capas de razonamiento de manera horizontal de esta forma todas las capas tienen acceso a la información sensorial obtenida por el agente. En la

Figura III-3: Modelos de Razonamiento según la Disposición de las Primitivas de Control, se presentan las estructuras básicas de estos dos modelos.

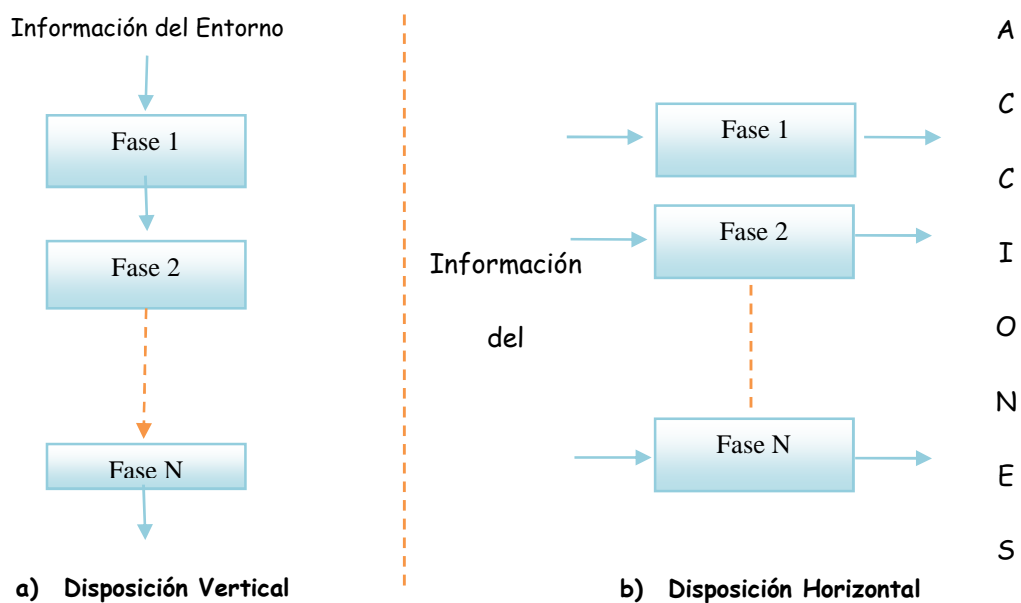


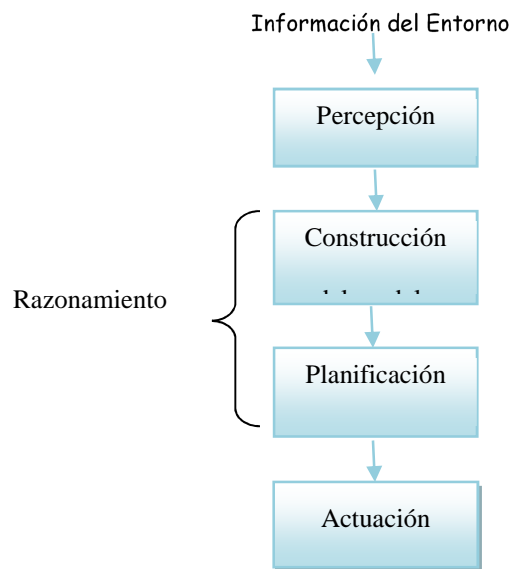
Figura III-3: Modelos de Razonamiento según la Disposición de las Primitivas de Control



La aplicación de estos modelos de razonamiento da como resultado tres paradigmas (Deliberativo, Reactivo, Híbrido) para el proyecto de grado se analizó el paradigma Deliberativo este se ajusta al sistema tomando en cuenta las limitaciones en hardware se describen en el presente capítulo y las limitaciones en software se describirán en el capítulo 4.

**Paradigma Deliberativo:** Son sistemas de resolución general de problemas (GPS, General Problem Solver) creados por Herbert Simon y Allen Newell en 1957 (Cairo & Olarte), este paradigma está basado en la premisa, un sistema es capaz de utilizar una representación simbólica del entorno que lo rodea, entonces podrán generar acciones, las mismas pueden considerarse inteligentes.

En este paradigma la función de razonamiento es la más importante, en la cual la información obtenida a partir de los sensores es modelada, construyéndose una representación simbólica del estado del entorno y aplicando técnicas, normalmente de planificación automática, para la elaboración de un plan secuencial de acciones, cuya aplicación supone la obtención de los objetivos de la tarea o tareas a realizar. Se observa este paradigma en la Figura III-4: Estructura del Paradigma Deliberativo.



**Figura III-4: Estructura del Paradigma Deliberativo.**

Este paradigma da buenos resultados en entornos reducidos y específicos, por este motivo se lo seleccionó, las desventajas de este tipo de modelo es la escalabilidad en entornos volátiles, debido a los algoritmos utilizados para la resolución de los problemas de planificación, en los cuales se asume la propiedad de racionalidad calculada, definida por Russell en 1993 (Marínez, 2011), consiste en lo siguiente:

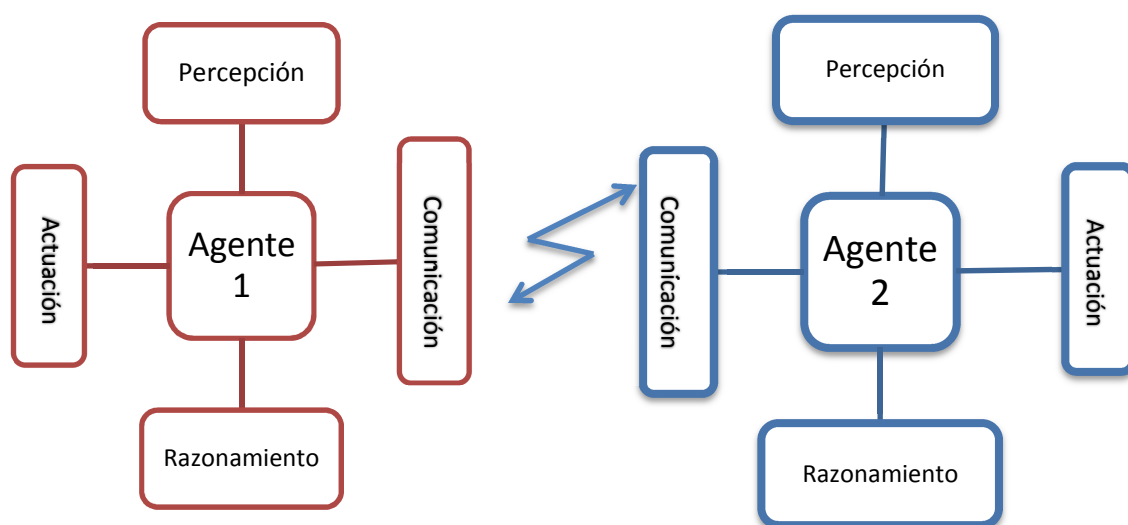
*"Dado un conjunto de información obtenida del mundo en un instante de tiempo  $t$ , la acción obtenida tras el proceso de planificación realizado utilizando la información previamente obtenida, debe ser aplicada en el instante de tiempo  $t$  en el cual comenzó el proceso de razonamiento para poder ser óptima."*

Esto quiere decir: Los cambios en el entorno no pueden producirse durante el proceso de deliberación, esta propiedad no se cumple en entornos volátiles o donde el tiempo de deliberación es muy elevado, del cual se obtiene un conjunto de acciones, el mismo no toma en cuenta nuevos aspectos del entorno.

Para el presente proyecto el entorno no puede variar, si esto ocurre el algoritmo implementado, no funciona y por ende las metas no logran ser alcanzadas por el sistema.

### 3.1.2 Arquitectura implementada

En el presente proyecto en la Figura III-5: Arquitectura Implementada para el Desarrollo del Proyecto, se muestra la implementación de una arquitectura distribuida permitiendo tener agentes independencia el uno del otro, donde las decisiones dependen de la capacidad de razonar de cada uno, ante un estímulo, para el sistema multi robots se utiliza el paradigma deliberativo para cada miembro del grupo, los cuales razonan de acuerdo a las situaciones del entorno que los rodea, mediante sus sensores generan una representación simbólica del área de trabajo encontrada, donde a cualquier cambio el algoritmo se vuelve ineficiente.



**Figura III-5: Arquitectura Implementada para el Desarrollo del Proyecto**

En la Figura III-2 : Disposición Básica de las Primitivas de Control, se determina las necesidades de un agente para actuar en el mundo que lo rodea, las cuales son:

- **Capacidad de percibir**, el agente logra percibir mediante los sensores, distancia, velocidad, trayectoria, y obstáculos.
- **Capacidad de razonar**, el agente consigue interactuar con el medio ambiente y con otros agentes mediante un controlador el cual le permitirá comunicarse y procesar los datos percibidos.
- **Capacidad de actuar**, el agente con la información procesada interactúa con el medio que lo rodea mediante acciones realizadas por sus actuadores.

### 3.1.2.1 Requerimientos del sistema Multi-robot

Con la arquitectura diseñada en el inciso anterior, se determina los dispositivos a utilizar, para la solución del problema descrito en el capítulo 1.

El sistema va a percibir mediante los siguientes sensores:

- Sensor Ultrasónico
- Sensor de Contacto
- Sensor de Pista IR
- Sensor de Rotación de motor

El sistema va a razonar a través del controlador proporcionado por la plataforma.

El sistema va a actuar mediante los siguientes actuadores:

- Motores
- Focos

El sistema se va a comunicar mediante el estándar de comunicación Bluetooth, el cual debería estar incorporado en el controlador.

### 3.1.2.2. Selección de la plataforma utilizada en el proyecto de Robótica

#### Cooperativa

Con los requerimientos establecidos anteriormente se selecciona a Fischertechnik “TANQUE” como el robot a utilizar en el proyecto de robótica cooperativa de acuerdo al anexo 8, donde se muestran las ventajas frente a otros robots.

Para resolver el problema planteado en el capítulo uno, respecto a la robótica cooperativa se realizó una arquitectura distribuida, con dos robots de similares características en hardware, a continuación se muestra la Tabla III-1: Descripción de los dispositivos del Sistema Multi – Robots.

Nombre del Dispositivo	Función del Elemento
Controlador	El controlador va permitir la interacción entre los diferentes sensores y actuadores.
Sensor Ultrasónico	Permite ubicarnos en el área de trabajo y evitar choques con las paredes.
Sensor de Pistas	Ayuda en la ubicación adecuado del “TANQUE” en la elemento a moverse.
Sensor de Contacto	Detecta cuando el obstáculo este cerca.
Sensor de Rotación de Motor (Encoder)	Controla la velocidad con la cual se mueven los motores, permitiendo de esta forma al “TANQUE” ir casi en línea recta.
Comunicaciones	Vía Bluetooth entre robots permite la coordinación.

**Tabla III-1: Descripción de los dispositivos del Sistema Multi- Robot**

### 3.1.2.2.1 Controlador del Sistema Multi - Robot

El controlador Tx tiene 12 puertos de entrada y 8 puertos de salida PWM, las 12 entradas pueden ser configuradas como entradas digitales, 8 entradas pueden ser configuradas como entradas analógicas decodificadas a través de un ADC, 4 entradas pueden ser configuradas como contadores rápidos, 8 salidas son compatibles con diversas cargas inductivas, semiconductores accionados por Mosfets proporcionan al menos 600 [mA] cada uno (para un total de 2,5 [A] Máx) a 9[V].

La unidad central de procesamiento es un Atmel AT91SAM9260, la velocidad del reloj es de 200 [MHz]. Este CPU cuenta con la mayoría del hardware requerido, por ende no se necesita colocar muchos elementos adicionales a la tarjeta de circuito impreso.

En la parte inferior hay un SDRAM de 8 [Mbyte] y una memoria flash de 2 [Mbyte]. Cada uno de los circuitos integrados del controlador dispone de ocho puertos Mosfets para un total de 16 Mosfets, dos para cada salida.

En la mitad superior tiene un codificador de línea y un conmutador analógico (IC), 8 pines son multiplexados para el desplazamiento de bits a través de ocho entradas analógicas.

Los puertos analógicos son conectados al mismo convertidor ADC multiplexados al procesador. Aunque los puertos son llamados "universales" las entradas pueden ser configuradas como señales digitales, en el modo digital esos puertos es una emulación de software.

Las entradas para los contadores rápidos están directamente conectadas a cada uno de los pines E/S del procesador, controlado por un interruptor bilateral para la activación y desactivación del software.

El módulo Bluetooth en la parte inferior derecha y los inductores para la fuente de alimentación en la parte inferior izquierda, en la Tabla III 2: Características Técnicas del Controlador, se observan los datos más relevantes.

<b>Característica</b>	<b>Dato Técnico</b>			
Dimensiones	90[mm]	90[mm]	15[mm]	
Pesos	90[gr]			
Memoria	RAM	8 MB		
	Flash	2 MB		
Procesador	ARM (9 -32)[bit]			
Reloj	200[MHz]			
Programable	ROBO Pro			
	Compilador- C			
Alimentación	Acu Set	8,4[V]	Corriente	1500[mA]
	Power Set	9[V]	Corriente	1000[mA]
Comunicación	USB	2.0	Máx.	12[Mbit]
	Bluetooth	2,4[GHz ]	Alcance	10[m]
	RS 485			
Entradas	Análogas	0-9[V]		
		0-5[kHz]		
	Digitales	0-9[V]		
		0-5[kHz]		
Salidas	Motor	9[V]	Corriente	500[mA]
Display	128 X 64			
	Monocromático			

**Tabla III-2: Características Técnicas del Controlador**

Como se mencionó anteriormente el controlador cuenta con diferentes tipos de entradas: universales las cuales se pueden configurar como análogas y digitales, contador las cuales se enuncian en la Figura III 6: Descripción Física del Controlador Tx

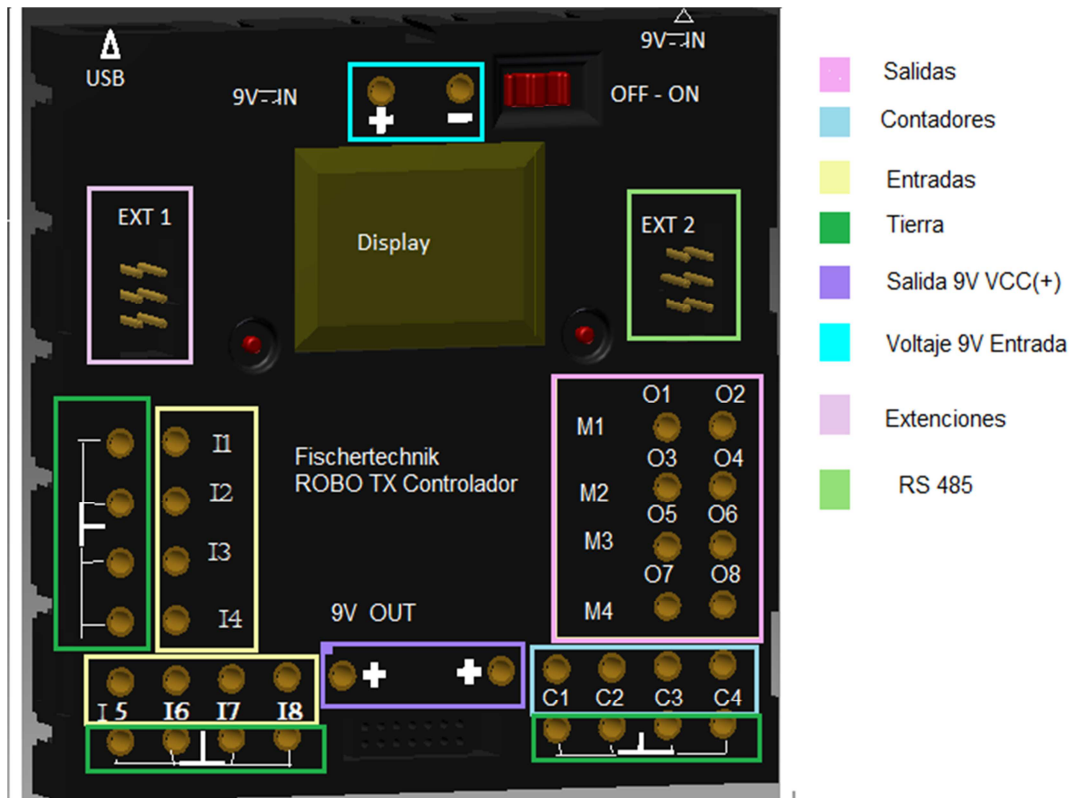


Figura III-6: Descripción Física del Controlador Tx

### 3.1.2.2.1.1 Entradas Universales: Análogas

En el controlador las entradas análogas figura III-5 pueden trabajar en dos modos de 10[v] y 5 [kΩ], la diferencia entre los dos modos es:

- En el primero acepta valores positivos máximos de 10[v]
- Mientras el segundo modo una carga resistiva conectada entre la entrada y la tierra, para el funcionamiento de estos modos se necesita un firmware el cual proporciona un voltaje en mili voltios o un número entero en la resistencia en Kilo ohmios.

#### 3.1.2.2.1.1.1 Entrada Análogas en modo 5[kΩ]

La entrada análoga en modo 5[kΩ] utiliza un divisor de tensión construido con la tierra, la fuente de alimentación Vcc y dos resistencias: una interna y la otra resistencia es el propio sensor, como se observa en la Figura III-7: Divisor de Tensión en Entrada Análoga 5[kΩ]. El sensor se conecta entre la entrada y la tierra. La resistencia interna suministra aproximadamente 2[V] a la entrada. Dependiendo del valor de resistencia del

sensor, el conversor analógico / digital (ADC), mide el valor de tensión de acuerdo con la ley de Ohm. El ADC no puede medir la resistencia directamente, sino indirectamente a través de la caída de tensión. El firmware convierte la lectura de tensión en un valor resistivo y escribe estos valores en un registro. Las resistencias internas se eligieron para proteger al sensor de cualquier corriente.

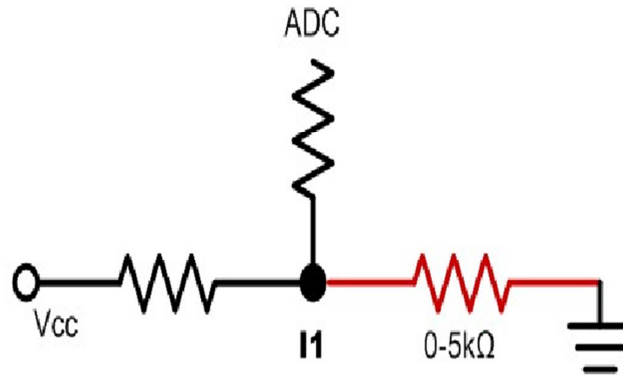


Figura III-7: Divisor de Tensión en Entrada Análoga 5[kΩ]

#### 3.1.2.2.1.1.2 Entrada Análogas en modo 10[V].

En el modo de 10[V], la configuración de hardware es muy similar. Se envía a la entrada por tierra y de esta forma no flota. Esto se hace con una resistencia de valor suficientemente alto, como se ve en la Figura III-8: Modo de Funcionamiento Entrada Análoga 10[V]. Cuando se conecta una fuente de tensión, las muestras de ADC dan directamente el valor, se escribe en un registro el cual se lee en la aplicación donde se ejecutando las sentencias, las entradas están protegidas para no ser altamente sensibles. Una tensión de al menos 75-100 [mV] necesita la entrada.

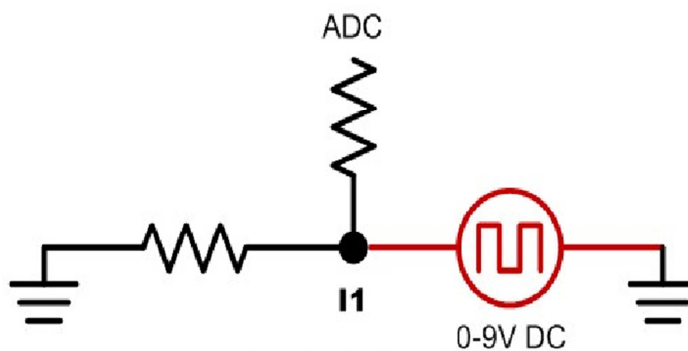


Figura III-8: Modo de Funcionamiento Entrada Análoga 10[V]



En la siguiente Tabla III 3: Comparación de las Entradas Análogas modo 10[v] y 5[kΩ], en la cual se describe las principales características de cada una de ellas.

Modo	10 [V]	5 [k]
V salida	0 [V]	2.2 [V]
A máx.	10 [u A]	60 [uA]
Unidades	[mV]	[Ohm]
Min	75	
Máx.	10000	5000
Resolución	100 [mV]	1 [Ohm]
Precisión	> 1[V] = 5%	100= 50%
		500= 20%
		>1[K] = 5%
Tipo Variable	Entera	Entera

Tabla III-3: Comparación de las Entradas Análogas modo 10[v] y 5[kΩ]

### 3.1.2.2.1.2 Entradas Digitales

Estas entradas figura III-5 poseen los dos modos descritos anteriormente y la conexión es la misma, la diferencia radica en los valores de histéresis, si las entrada dispara un valor menor a 500 [Ω] se lee 1 lógico, y superior a esta resistencia se lee 0 lógico.

Para explicar mejor el funcionamiento de estas entradas se realizó un experimento, se colocó un actuador un ventilador, junto a un sensor de calor NTC, creando una situación donde el ventilador y el sensor envían valores constantes alrededor de los 500[Ω], sin embargo este comportamiento no es deseado por lo tanto se usa un circuito llamado Schmitt Trigger entre el trabajo analógico y digital. Un disparo Schmitt introduce histéresis

en el ejemplo, los sensores NTC se modifican a  $450[\Omega]$  a 1 lógico, pero sólo libera a  $550[\Omega]$  a 0 lógico. (Fischertechnik, Eight MOSFET driven Power Outputs, 2012)

Modo	10[V]	5[k]
V salida	0[V]	2.2[V]
A Máx.	10 [uA]	60 [uA]
Unit	[mV]	[Ohm]
Min	0	43
Máx.	10000	5000
0	540	723
1	840	439
Histéresis	300	284
Tipo Variable	Bool	Bool

**Tabla III-4: Comparación de las Entradas Digitales modo 10[v] y 5[k $\Omega$ ]**

Como se puede ver en la Tabla III-4: Comparación de las Entradas Digitales modo 10[v] y 5[k $\Omega$ ], los valores de umbral en el modo de 10[V] son aproximadamente 0,5 [V] para 0 lógico y 0,9 [V] para 1 lógico. En el modo de 5[k $\Omega$ ] estos son 430[ $\Omega$ ] 1 lógico y 750[ $\Omega$ ] para 0 lógico. En el modo de 10 [V] el umbral inferior crea un 0 lógico, con las tensiones inferiores a 0,5[V], mientras en el modo de 5[k $\Omega$ ] una mayor resistencia crea un 0 lógico y por lo tanto 750[ $\Omega$ ] y superior.

### 3.1.2.2.1.3 Entrada a Contadores

Las entradas de contador son las entradas digitales conectadas a los pines digitales E/S en el procesador, son entradas generalmente utilizadas para conectar los sensores de rotación de los motores (encoders).

Los umbrales son diferentes de las entradas universales en modo digital. Las entradas de contador son digitales y espera una señal digital en alto o bajo, pero también puede funcionar con una carga resistiva, porque internamente está dando 5[V]. Los umbrales son aproximadamente 1[V] para 1 lógico y 2[V] para 0 lógico y funciona con niveles TTL o CMOS. Los umbrales resistivos son 620[Ω] para 1 lógico y 1.6 [kΩ] para la 0 lógico. En ambos casos, se aplica la lógica negativa. (Interfaces universales usan la lógica negativa en el modo de 5[kΩ] pero positiva en el modo de 10[V].)

Los valores son descritos en la Tabla III-5: Comparación del Contador modo 10[v] y 5[kΩ], para las entradas digitales universales, estas son características del software en el capítulo 4 se amplía el tema.

Modo	10[V]	5[kΩ]
V salida	5.5[V]	5.5[V]
A Max	1.8 [mA]	1.8 [mA]
Unit	mV	Ohm
Min	0	0
Max	5	5
0	1990	992
1	630	1520
Histéresis	998	890
Tipo Variable	Bool	Entera
Resolución	Reset en 1	Reset en 1

Tabla III-5: Comparación del Contador modo 10[v] y 5[kΩ]

### 3.1.2.2.2 Sensores

En esta parte de la investigación se procedió a utilizar el método ensayo y error determinando así los sensores adecuados, para proporcionar al sistema la información

sensorial necesaria, con la misma se puede cumplir con las metas especificadas, el “TANQUE” cuenta con una serie de sensores entre ellos sensor ultrasónico, sensor de pistas IR, sensor de contacto, sensor de rotación de motor “encoders”, resistor NTC, foto resistor, sensor óptico de colores, se menciona las pruebas realizadas para determinar los sensores seleccionados y se describen sus funcionamientos.

Las pruebas realizadas en el laboratorio para determinar la precisión de los sensores dieron como resultado, los dispositivos adecuados en el presente proyecto son: sensor ultrasónico, sensor de pistas IR y sensor de contacto como se menciona al principio del capítulo el entorno no puede sufrir cambios volátiles por ende estos sensores se acoplan adecuadamente al diseño para determinar el entorno.

### ➤ Sensor Ultrasónico

El sensor permite al robot determinar la distancia a la cual se encuentra un objeto y de esta forma determinar la acción a seguir, en este caso el robot con las distancias obtenidas puede conocer si el objeto se encuentra frente a él es la pared del área de trabajo o la caja a moverse.

Está constituido por un emisor y un receptor, el mismo envía un sonido por encima del oído humano, un umbral inferior de los 16[kHz] y un umbral superior de 1.6 [GHz], este sensor detecta los objetos a partir del eco como se observa en la Figura III-9: Principio Básico de Funcionamiento de un Sensor Ultrasónico, mide el tiempo desde la transmisión hasta la recepción, esta operación la transforma en la distancia entre el objeto y el sensor mediante la técnica de pesca.

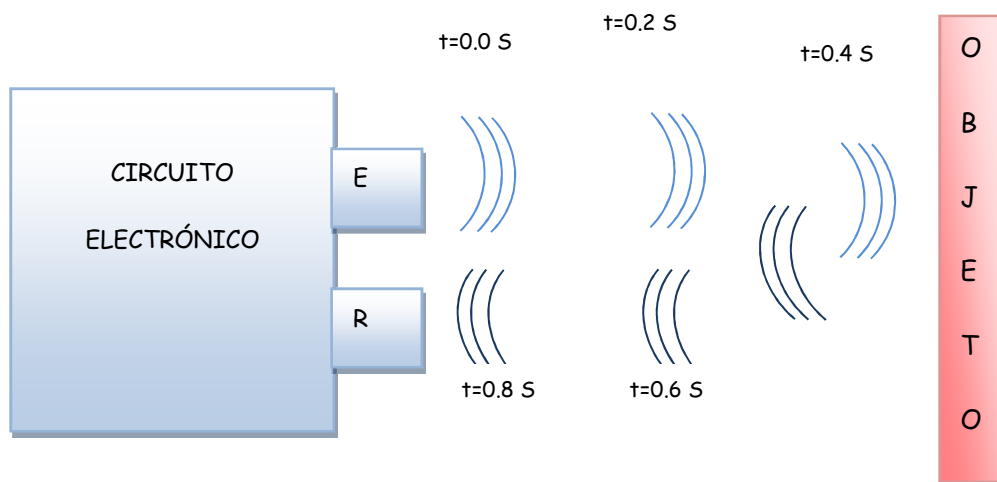


Figura III-9: Principio Básico de Funcionamiento de un Sensor Ultrasónico

Las características del sensor ultrasónico son (Fischertecchnik, 2010):

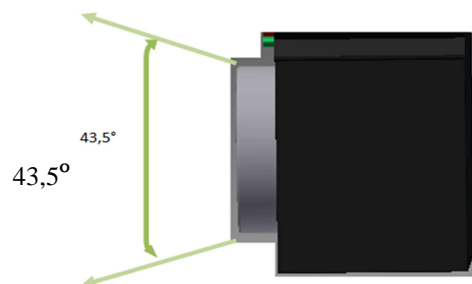
- Rango de medición de 2 a 400 cm
- Devuelve la unidad de distancia en cm

☞ Se observa en la Figura III-10: Cobertura Horizontal del Sensor Ultrasónico, el ángulo es solo de  $65^\circ$ .



**Figura III-10: Cobertura Horizontal del Sensor Ultrasónico**

☞ En la Figura III-11: Cobertura Vertical del Sensor Ultrasónico se puede observar el ángulo es de  $43,5^\circ$ .



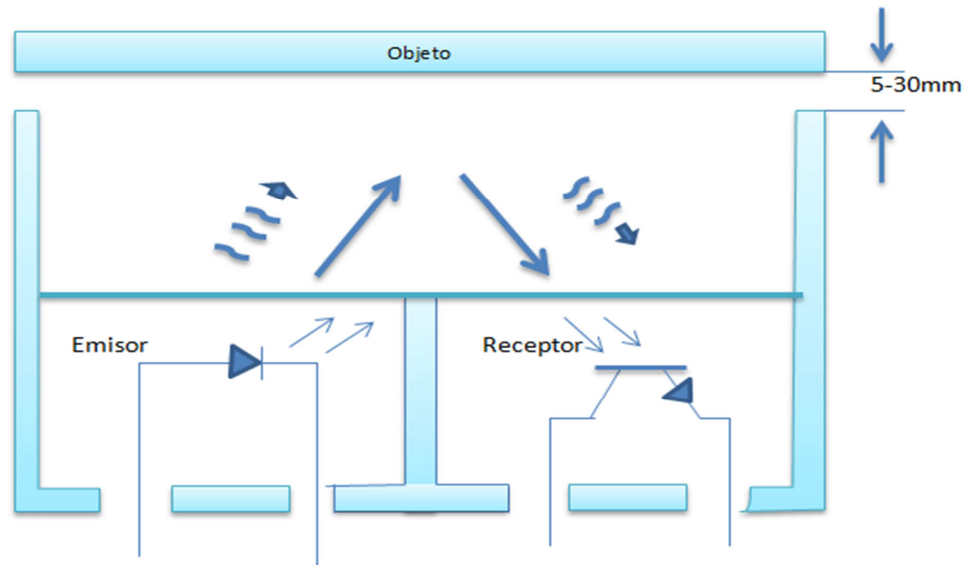
**Figura III-11: Cobertura Vertical del Sensor Ultrasónico**

☞ Dimensiones: (4.5 x 1.7 x 3) cm

#### ➤ **Sensor de Pistas IR**

Este sensor va a permitir al robot seguir un camino, de esta forma el robot impacta a la caja en un mismo punto varias veces determinando de manera más precisa el peso de la misma, es decir si necesita o no ayuda para moverla en la primera tarea a realizar demostrando robótica cooperativa, en la segunda meta este sensor ayuda a los robots para mantenerse en el círculo.





**Figura III-13: Principio de Funcionamiento del Sensor IR**

### ➤ Sensor de Contacto

Los sensores de contacto son los dispositivos más simples de todos los sensores, son interruptores se activan o desactivan si se encuentran en contacto con un objeto, de esta manera se reconoce la presencia de un objeto en un determinado lugar. Su simplicidad de construcción añadido a su robustez, los hacen muy empleados en la construcción de micro robots.

Este dispositivo va a permitir conocer al sistema el peso la caja, el sensor de contacto utilizado tiene 3 pines, como se observa en la Figura III-14: Sensor de Contacto.



**Figura III-14: Sensor de Contacto**

### ➤ Sensor de Rotación del Motor (Encoder)

Los Encoder generan señales digitales en respuesta al movimiento. Están disponibles en dos tipos, uno es de rotación, y el otro al movimiento lineal, el Encoder del “TANQUE” es incremental lo cual significa, generar pulsos mientras se mueven, se utilizan para medir la velocidad, o la trayectoria de posición.

La tecnología en estos Encoder es magnética, se utilizan frecuentemente en aplicaciones de trabajo pesado como en laminadoras de papel y acero, proveen buena resolución, altas velocidades de operación, y máxima resistencia al polvo, humedad, y golpe térmico y mecánico.

La tecnología magnética es muy resistente al polvo, grasa, humedad, y a otros contaminantes comunes en los ambientes industriales, así como a los golpes y vibraciones, utiliza un imán permanente y un efecto de Hall o dispositivo magneto resistivo para producir un cambio en cualquier voltaje o resistencia eléctrica en presencia de material ferromagnético, se puede observar en la Figura III-15: Principio de Funcionamiento del Sensor Magnético. (Fischertechnik, The TX brick accepts a wide variety of inputs, 2011)

Los engranes utilizados en los motores para el proyecto tienen una relación de transmisión de 25 a 1, entonces una vuelta de árbol proveniente del engrane es equivalente a 75 impulsos en el Encoder, estos dispositivos permiten al sistema un control de velocidad de sus motores, con el fin de lograr una sincronización en la tarea designada, las características técnicas se encuentran el anexo 1 Datasheet Encoder.

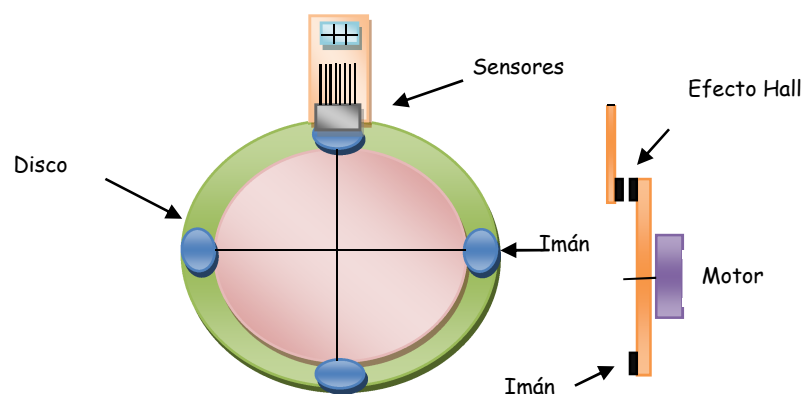


Figura III-15: Principio de Funcionamiento del Sensor Magnético



### 3.1.2.2.3 Comunicaciones

La comunicación es parte fundamental en el presente proyecto se utiliza Bluetooth, los controladores cuentan internamente con los dispositivos instalados, como se observa en la Figura III-16: Comunicación entre los Dos Robots.

Bluetooth es una especificación industrial para redes inalámbricas de área personal (WPAN) posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 [GHz]. Los principales objetivos de utilizar esta norma son:

- Facilitar las comunicaciones entre equipos móviles y fijos.
- Eliminar los cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

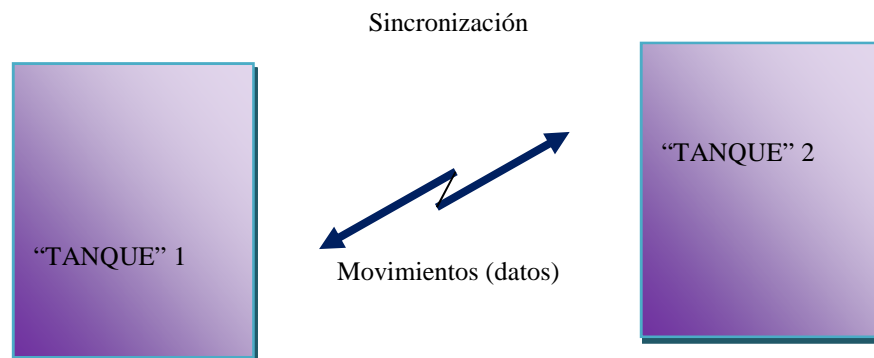


Figura III-16: Comunicación entre los Dos Robots

### 3.1.1.2.4 Actuadores

Los actuadores son elementos ante un estímulo proceden a realizar una acción para este caso los estímulos son la información sensorial y los actuadores en el proyecto son motores los mismos permiten el movimiento de los "TANQUES" y luces, se activan en determinados momentos indicando los eventos realizados por los robots.

## ➤ Motores

Como accionamiento para los robots se emplea dos motores por “TANQUE”, son motores eléctricos, están dimensionados para una tensión de 9 [V] y un consumo de corriente de máximo 0.5 [A], más información anexo 2.

Los motores tienen una caja reductora, se muestra en la Figura III-17: El Principio de una Caja de Reducción, con una relación de transmisión de 25 a 1, una vuelta de árbol proveniente del engranaje, son 75 impulsos del codificador.

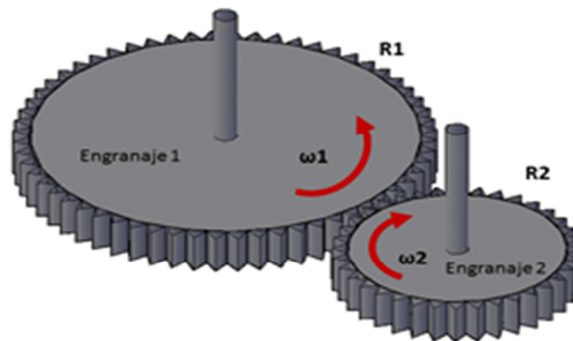


Figura III-17: El Principio de una Caja de Reducción

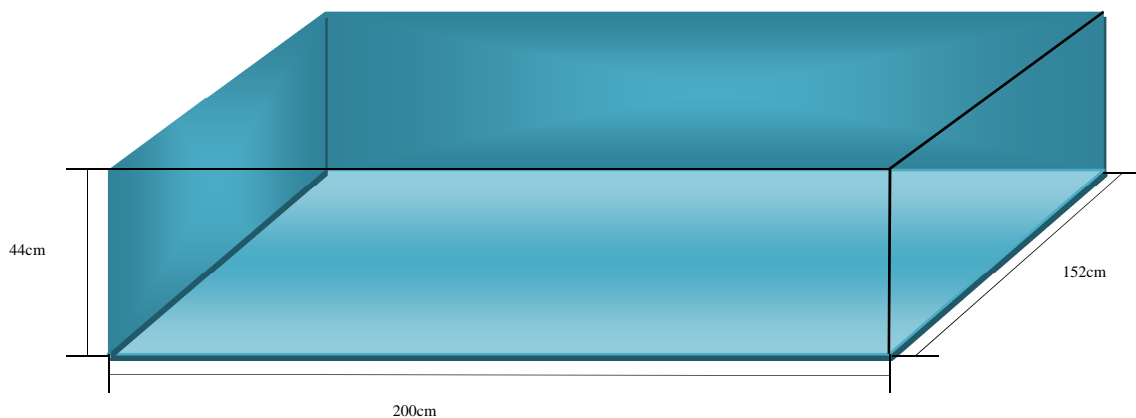
### 3.1.3 Prototipos para el Sistema Multi - Robot

Para el presente proyecto se creó un área de trabajo de (200 x 152 x 44) [cm] se observa en la Figura III-18: Dimensiones del Área de Trabajo, él cual es un ambiente controlado se dimensionó de esta forma por las características técnicas de los sensores, teniendo mayor influencia el sensor ultrasónico con las dimensiones descritas anteriormente, el sensor tiene un campo de visión de todos los elementos sobre el área de trabajo, no se confunde con los objetos externos en la misma.

Se crean dos prototipos, El prototipo 1.0 se pensó en darle el mayor grado de inteligencia posible, sin ocupar al computador como procesador principal, si no los procesadores internos de los equipos, en la primeras pruebas se consiguió dar inteligencia al robot más no se logró tener un funcionamiento continuo, volviéndolo al sistema obsoleto, se lo ocupaba un número determinado de veces, luego del cual sus actuadores dejaban de tener el rendimiento necesario para el proyecto.

El prototipo 1.1 tiene mayores ventajas al prototipo anterior sin embrago se pierde inteligencia e independencia.

A continuación se explica detalladamente los cambios efectuados del prototipo 1.0 al 1.1 y el porqué de los mismos.



**Figura III-18: Dimensiones del Área de Trabajo**

### 3.1.3.1 “TANQUE” versión 1.0

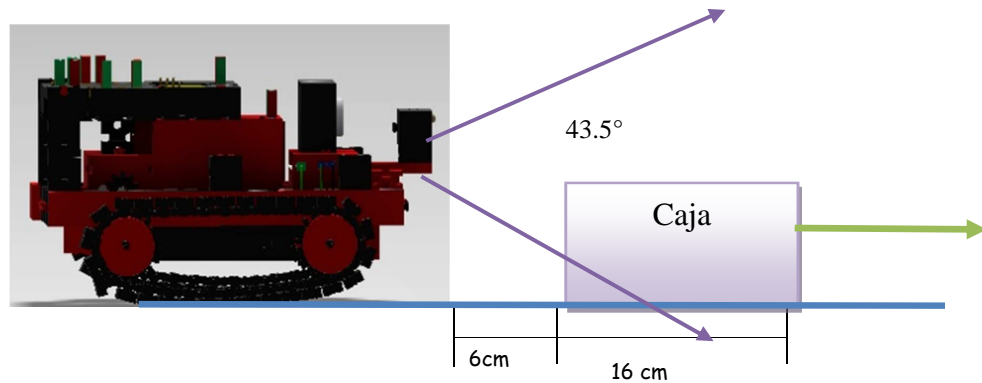
El diseño inicial se realizó en base al manual de ensamble del “TANQUE” Figura III-19: Prototipo 1.0, en el mismo se utiliza una batería accuset de 9[v] y 1,5 [mA], permitiéndole tener independencia, es decir, no se necesita cables de ningún tipo, sin embargo el consumo de la corriente es alto en el sistema, debido a la utilización de los diferentes sensores: de color, ultrasónico y actuadores, dando como resultado la descarga rápida de la batería reduciendo de este modo el tiempo de trabajo con los robots.



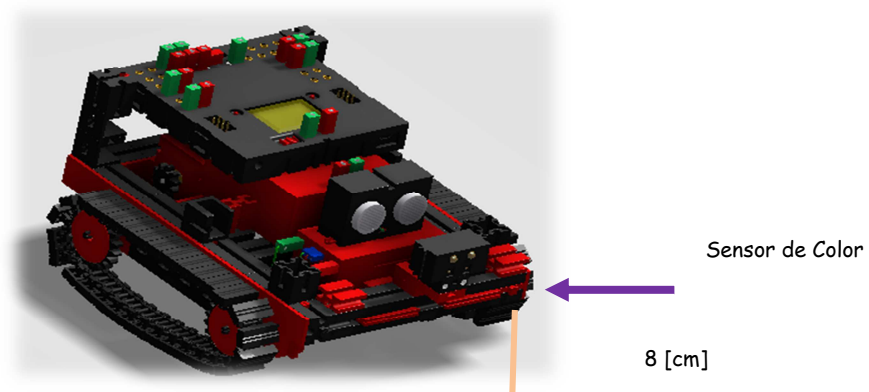
**Figura III-19: Prototipo 1.0**

En primera instancia el sensor ultrasónico se encontraba del piso 10 [cm] hacia arriba, y el sensor de color a 8 [cm] hacia arriba, el efecto obtenido fue: el sensor

ultrasónico detectaba rápidamente la caja y la confundía con la pared como se lo muestra en la Figura III-20: Prototipo 1.0, Rango de Detección del Sensor Ultrasónico, causando inconvenientes en el desarrollo del algoritmo, en la Figura III-21: Ubicación del Sensor de Color en la Infraestructura para tener un mejor ángulo de visión y de esta forma no interfiera mucho con la luz y por ende los cambios no sean tan abruptos sin embargo por las características técnicas del mismo no se logró adaptarlo a las necesidades del sistema.



**Figura III-20: Prototipo 1.0, Rango de Detección del Sensor Ultrasónico**



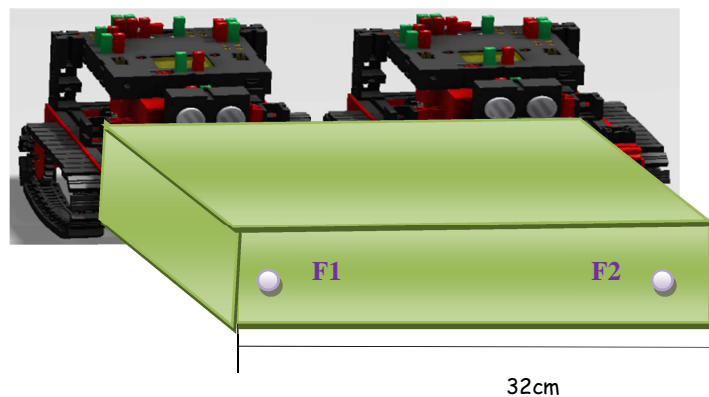
**Figura III-21: Ubicación del Sensor de Color en la Infraestructura**

La robótica cooperativa en la presente investigación implica a dos elementos, interactúan entre sí, en la investigación el problema planteado es mover una caja en línea recta con la sincronización de dos elementos robóticos, la caja utilizada en el diseño inicial tiene las dimensiones (32 x 16 x 5) [cm], y un peso de 553 [gr], los problemas suscitados se presentan a continuación:

- El peso del elemento a mover (caja) por los dos robots, era reducido de la capacidad de los mismos, en una primera instancia se tomó ese peso debido a la posición de los sensores y el infraestructura del robot, con la batería la estructura del robot se volvía más pesada y por ende perdía su fuerza en mover su estructura y no movía una caja acorde a sus características.

- La altura de la caja, y el sensor ultrasónico colocado sobre la batería, no favorecía para tener una visión del ambiente y de donde se encuentra la caja como se lo ve la Figura III-20: Prototipo 1.0, Rango de Detección del Sensor Ultrasónico.

- El largo de la caja no era suficiente como se muestra en la Figura III-22: Caja Pequeña para los Puntos de Fuerza, para colocar a los robots en los puntos de fuerza necesarios para ejercer el movimiento lineal con el cual se pretende demostrar la robótica cooperativa, los puntos de fuerza en esta caja se encuentran a los extremos.



**Figura III-22: Caja Pequeña para los Puntos de Fuerza**

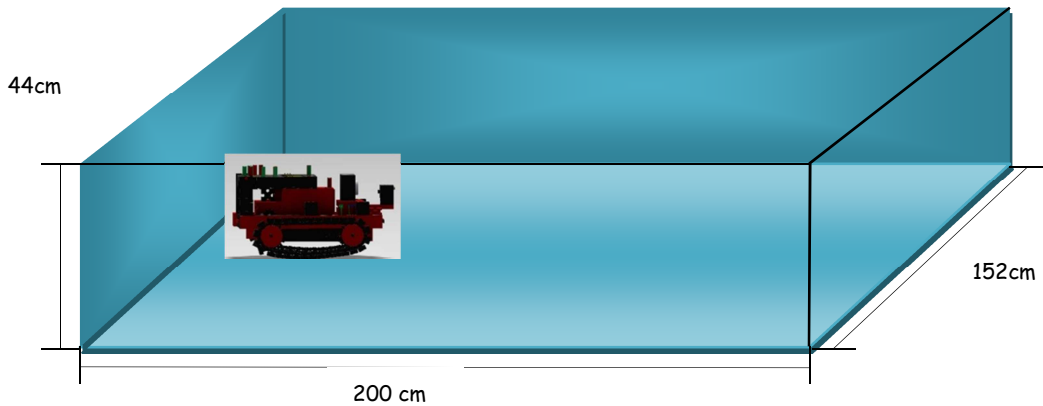
- No se utiliza un sensor de pistas IR, pretendiendo dar mayor inteligencia al sistema, sin éxito los actuadores con el uso continuo, comenzaron a perder fuerza, como resultado: los robots dejan de ir en línea recta y no mueven la caja.

- En una primera instancia se pintó las caras de la caja de diferentes colores con el fin de conocer los colores con los cuales se podría trabajar, la idea era pintar la caja liviana de un color y la pesada de otro.

- El sensor de color variaba demasiado con los cambios en la intensidad de luz ambiental, los rangos de color eran demasiado grandes como para hacer una barrido, al

depender de un factor ambiental difícil de predecir y controlar, por este motivo se decidió cambiar por un sensor de contacto.

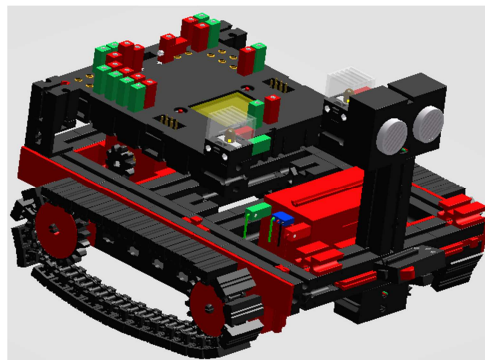
Se puede observar en la Figura III-23: Prototipo 1.0 y su Área de Trabajo.



**Figura III-23: Prototipo 1.0 y su Área de Trabajo**

### 3.1.3.2 “TANQUE” versión 1.1

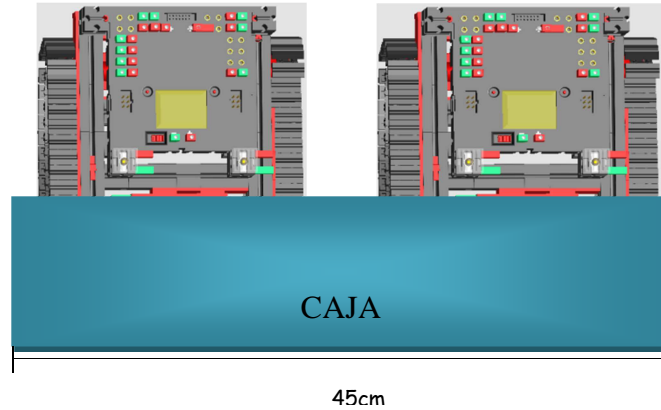
En el prototipo 1.1 se cambió la posición del sensor ultrasónico de 10 [cm] a 20[cm] hacia arriba y 4[cm] hacia afuera, se suprimió el sensor de color por un sensor de contacto y se agregó un sensor de pistas IR, la infraestructura del robot con este diseño perdió un poco de inteligencia, al solo seguir un camino predeterminado, para impactar a la caja en un mismo punto, en el desplazamiento del objeto se sincronizan los controladores para comenzar a empujar, se eliminó la batería y el diseño de la infraestructura cambio como se lo ve en la Figura III-24: Prototipo 1.1.



**Figura III-24: Prototipo 1.1**

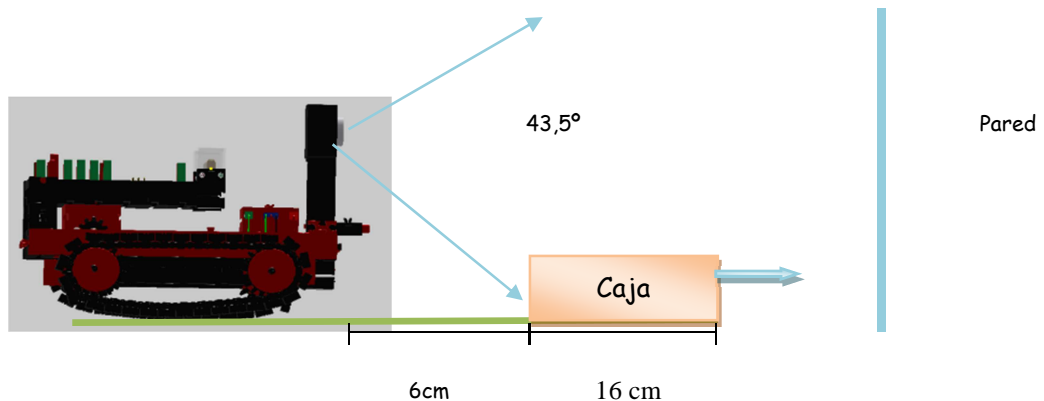
La caja fue redimensionada en largo 45 [cm] y la altura a 5,7[cm], con la infraestructura mejorada y la caja redimensionada los resultados fueron:

- Los 2 robots tiene una distancia de seguridad para mover la caja y evitar choques entre los robots, con las nuevas dimensiones de la caja los puntos de fuerza fueron distribuidos de manera de poder empujar un objeto (caja), como se muestra en la Figura III-25: Posición Puntos de Fuerza en la Caja.



**Figura III-25: Posición Puntos de Fuerza en la Caja**

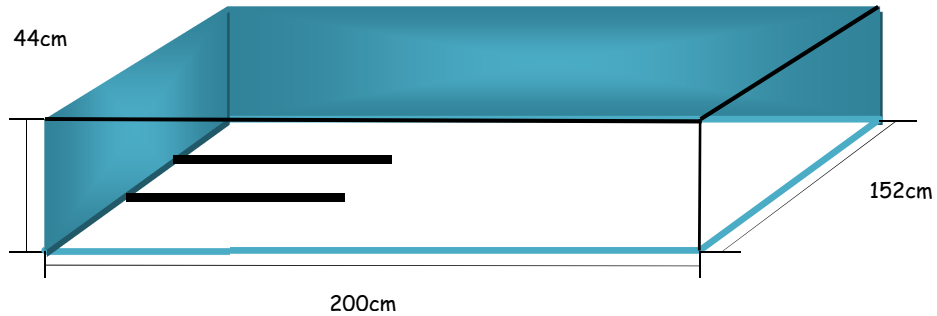
El ángulo de visión del sensor ultrasónico permitirá al robot no confundir la caja con la pared como se ve en la Figura III-26: Distancia Mínima entre el Robot la Caja y la Pared, Sí el robot confunde la caja con la pared se chocara con la misma y de este modo se forzará al intentar moverla, en conclusión las actuadores sufrirían.



**Figura III-26: Distancia Mínima entre el Robot la Caja y la Pared**

- El sensor de color fue reemplazado por un sensor de contacto, el mismo permite saber el peso de la caja, de acuerdo el algoritmo utilizado en el capítulo 4 se explica con mayor detalle.
- Los “TANQUES” al no llevar la batería pueden mover juntos un peso mayor 603[gr].
- El sensor de pistas IR, ayuda al sistema a impactar en un mismo punto varias veces, permitiéndole al robot no confundirse con el peso de caja, con el prototipo anterior

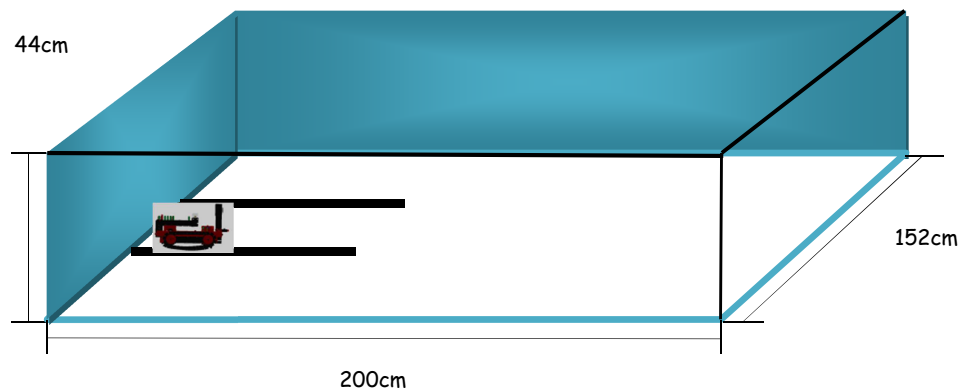
al instante del impacto, Sí la caja se movía unos pocos centímetros, el “TANQUE” se confundía, y asumía un peso erróneo de la caja, la Figura III-27: Área de Trabajo Modificada para Sensor IR, muestra los cambios realizados en el área de trabajo para cumplir con el objetivo antes mencionado.



**Figura III-27: Área de Trabajo Modificada para Sensor IR**

- Con la versión 1.1 se logra probar varias veces el algoritmo sin presentar mayores inconvenientes con los actuadores, el inconveniente de este prototipo es la necesidad del cargador, limitando la distancia de movilidad, la interferencia en el desenvolvimiento de los robots en el área de trabajo.

Se observa en la Figura III-28: Prototipo 1.1 y Área de Trabajo



**Figura III-28: Prototipo 1.1 y Área de Trabajo**



## CAPÍTULO IV .

### DISEÑO DEL SOFTWARE

En este presente capítulo se hará una descripción de los algoritmos utilizados así como también del análisis matemático necesario para resolver los problemas planteados en el capítulo 1, tomando en cuenta las limitaciones de hardware. Se definen el lenguaje de programación y las herramientas de compilación.

#### 4.1 SISTEMAS MULTI – ROBOT SOFTWARE

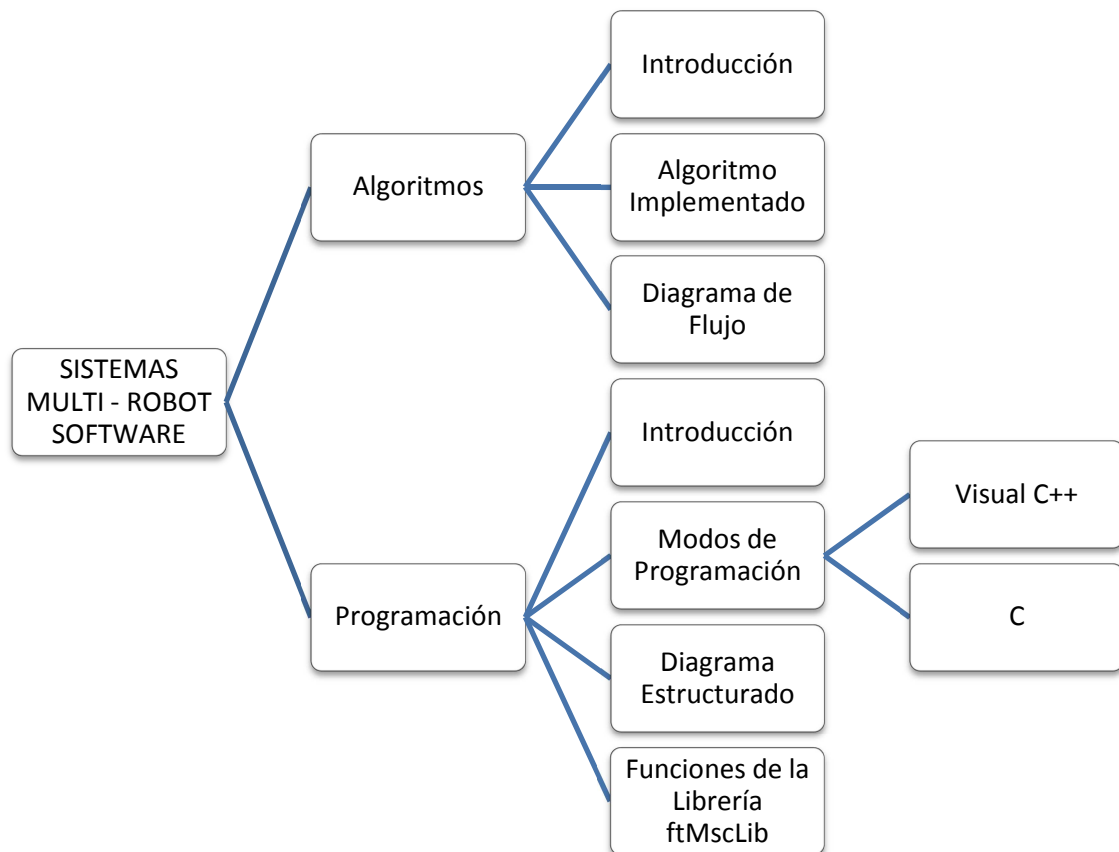


Figura IV-1: Estructura del Capítulo IV

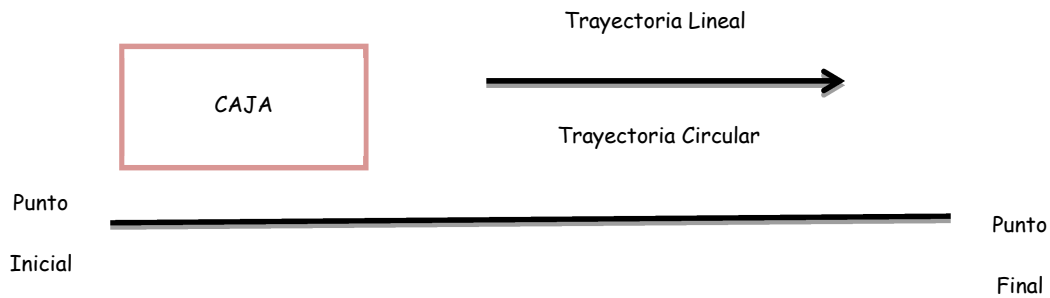
## 4.1.1 Algoritmos

### 4.1.1.1 Introducción

*“Un algoritmo es un conjunto finito de pasos definidos, estructurados en el tiempo y formulados con base a un conjunto finito de reglas no ambiguas, que proveen un procedimiento para dar la solución o indicar la falta de esta a un problema en un tiempo determinado.” [Rodolfo Quispe-Otazu, 2004].*

### 4.1.1.2 Algoritmo Implementado

**Problema:** Se necesita mover una caja desde un punto inicial a un punto final, el movimiento tiene dos trayectorias: lineal y circular, véase Figura IV-2: Descripción Gráfica del Problema



**Figura IV-2: Descripción Gráfica del Problema**

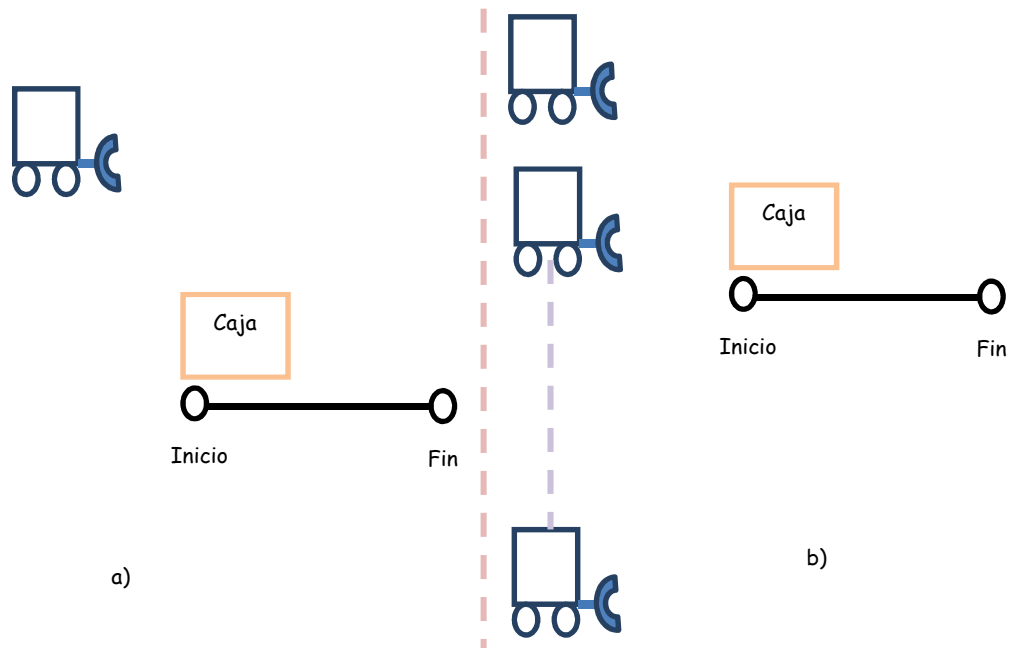
Sí se necesita mover una caja desde un punto inicial a uno final. Su solución será usar uno o más robots, para lo cual el algoritmos determinará cuántos robots son necesarios para mover la caja, y la trayectoria (lineal y circular) a realizar.

#### 4.1.1.2.1 Consideraciones particulares de la solución

- La forma de la caja debe ser un ortoedro
- La masa de la caja es uniforme
- La caja debe ser sólida
- El rozamiento entre la caja y superficie debe ser mínimo.
- Dimensiones de los robots.

- Dimensiones de la caja.
- Peso mínimo y máximo de la caja.

### Solución



**Figura IV-3: Solución Usando Robots**

En la solución planteada en la Figura IV – 3: Solución Usando Robots, se determina la necesidad de utilizar agentes basados en metas con la función específica de mover cajas, en el desarrollo del proyecto se los denominará agente movedor de cajas, los cuales para resolver el problema descrito anteriormente deben formar un grupo de robots en la Figura IV-4: Sistema Multi Robots, se muestra la interacción de un grupo de 3 agentes, comunicándose e interactuando entre sí para lograr un objetivo en común, se podría tener n agentes realizando la tarea.

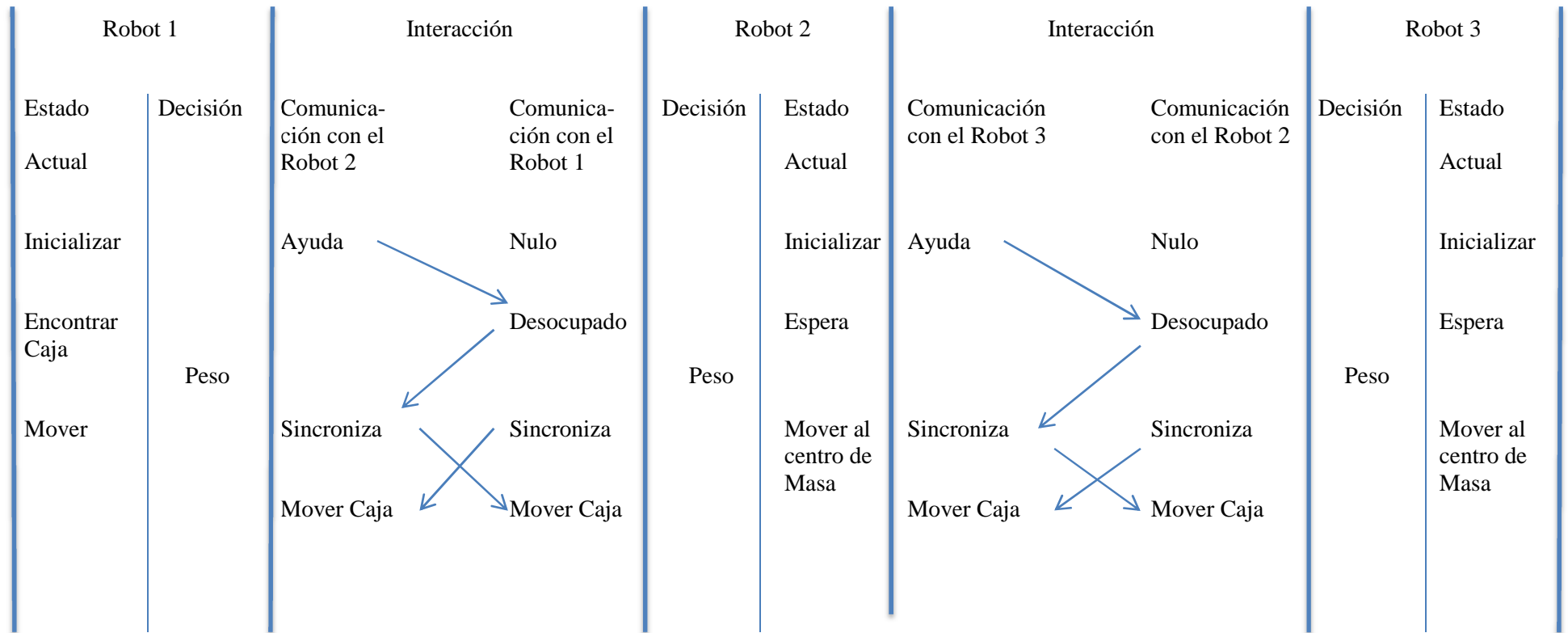


Figura IV-4: Sistema Multi Robots

El sistema de la Figura IV-4: Sistema Multi Robots o multi agentes están conformados por agentes basados en metas, en la Tabla IV-1: Agente Mover de Caja, se muestra las percepciones y actuaciones con el medio que los rodea.

Tipo de Agente	Percepciones	Acciones	Metas	Ambiente
Movedor de cajas	<ul style="list-style-type: none"> <li>• Velocidad del Motor</li> <li>• Pulso</li> <li>• Distancia a un objeto</li> <li>• Píxeles Blancos o negros</li> </ul>	<ul style="list-style-type: none"> <li>• Los motores caminan distancias predeterminadas</li> <li>• Sincronización de los motores</li> <li>• Prender Focos</li> </ul>	<ul style="list-style-type: none"> <li>• Seguir trayectorias</li> <li>• Determinación del número de robots</li> <li>• Mover cajas</li> </ul>	<ul style="list-style-type: none"> <li>• Pista negra sobre blanca.</li> <li>• Cajas de diferentes pesos</li> </ul>

Tabla IV-1: Agente Mover de Cajas

En la Tabla IV-1: Agente Mover de Caja, se describe las metas a cumplir por estos agentes a continuación se enuncia la programación orientada a agentes basado en metas así el algoritmo general se basa en 3 metas:

- Tabla IV -2: La Meta a Seguir una Trayectoria Lineal
- Tabla IV 3: La Meta a Determinar el Número de Robots
- Tabla IV 4: La Meta es Mover Cajas

Entrada	Píxeles Blancos o Negros	
Estado	Si Pixel es negro	Seguir a la misma velocidad
	No pixel negro	Gira un motor
Meta	Seguir Trayectorias lineal	
Plan	Seguir una línea recta una distancia de 40 [cm]	
Mientras (Plan no se termine)		
Acción	Giro del motor izq. y aumentar la velocidad motor derecho Giro del motor der. Y aumentar la velocidad motor izquierdo. Mantener la misma velocidad	
Salida	Acción	

Tabla IV-2: La Meta a Seguir una Trayectoria Lineal

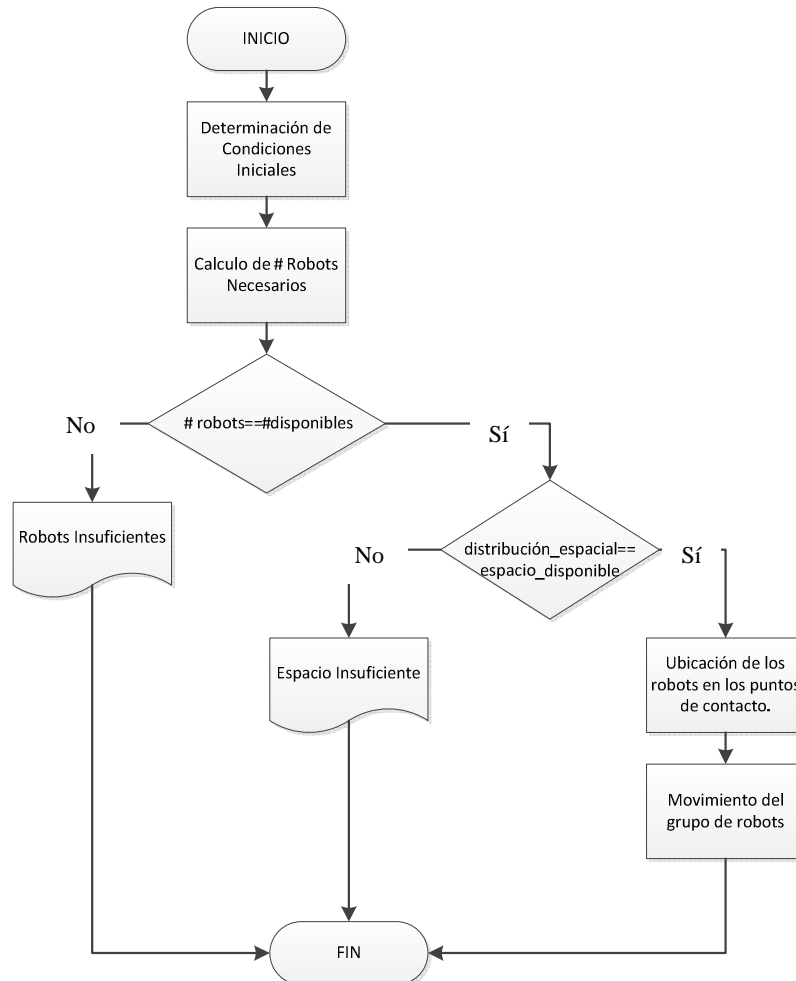
Entrada	1 Pulso Número de vueltas del Encoder	
Estado	Número de vueltas del Encoder menor que 2	Caja pesada y necesita 2 robots
	Número de vueltas del Encoder mayor que 2	Tiene una caja liviana y 1 solo robot
Meta	Determinar el número de robots	
Plan	Probar si un robot puede mover la caja	
Mientras (Plan no se termine)		
Acción	Mover el robot por un lapso pequeño Pedir ayuda si el robot no avanza	
Salida	Acción	

**Tabla IV-3: La Meta a Determinar el Número de Robots**

Entrada	Comunicación Bluetooth Pulso	
Estado	Pedir ayuda	Necesita 2 robots
	No pidió ayuda	1 solo robot
Meta	Mover Cajas	
Plan	Mover una caja una distancia de 50 [cm]	
Mientras (Plan no se termine)		
Acción	Posicionarse en los centro de masa de acuerdo al número de robots Sincronizar las velocidades de los motores	
Salida	Acción	

**Tabla IV-4: La Meta es Mover Cajas**

Una vez conocido las metas se procede a realizar en la Figura IV-5: Diagrama de Flujo General del Problema, el cual describe la programación a realizar.



**Figura IV-5: Diagrama de Flujo General del Problema**

#### 4.1.1.2.1.1 Descripción del Diagrama de Flujo

1. Determinar la cantidad de robot, que entran en la caja, mediante la relación geométrica entre robot y caja.

Lr: Largo del robot

n: número de robots

Lc: Largo de la caja

$$Lr * n = Lc$$

$$n = \frac{Lc}{Lr}$$

2. El robot 1 encuentra la caja de forma ortoedro y verifique Sí la puede mover, caso contrario llama a otro agente y vuelve a probar así sucesivamente hasta determinar la cantidad de robot necesarios para mover la caja o hasta que el paso 1 se cumpla.
3. Calcular la cantidad de robots necesario mediante la relación fuerza y el torque del motor

F: Fuerza necesaria para mover la caja.

tr: Torque del motor

$$F = tr$$

4. Determinar sí un solo robot puede realizar el movimiento lineal o circular, caso contrario ir al paso 5.
5. Se comunican los robots entre ellos y se determina la disponibilidad de cada uno, caso contrario se informa el inconveniente presentado.
6. Sí el número de robots necesarios se encuentran disponibles se determina los puntos de contacto, de acuerdo al número de robots necesarios para lo cual se utiliza la teoría de centro de masas, si se necesita un robot el punto de contacto es el medio pero si la caja es más pesada y se necesitan 2 robots el centro de masas se divide en 2, si necesita otro robot el centro de masas se divide en 3 y así sucesivamente de acuerdo al número de robot, como se observa en la Figura IV-6: Puntos de Contacto Dependiendo del Número de Robots.

CM: Centro de Masas

d = diagonal

h = altura de la caja

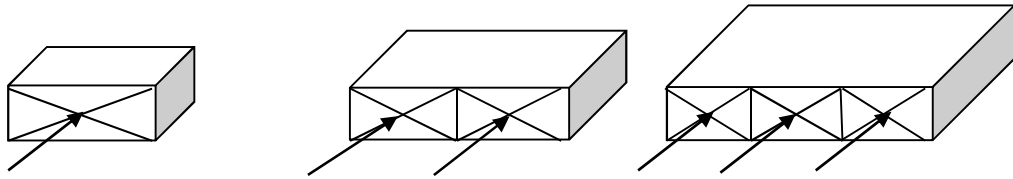
l = largo de la caja

n = número de robots

$$d = \frac{l}{n}$$



$$CM = \sqrt{h^2 + d^2}$$



**Figura IV-6: Puntos de Contacto Dependiendo del Número de Robots**

7. Los robots se ubican en los puntos de contacto.
8. Verificar la trayectoria (lineal o circular) para mover la caja y determinar las velocidades de los robots.
9. Sincronizar las salidas de todos los robots y realizar el movimiento.

Con el algoritmo antes mencionado, se procedió a realizar dos aplicaciones las cuales permiten saber sí el algoritmo funciona correctamente:

#### **4.1.1.2.2 Aplicaciones para la demostrar robótica cooperativa**

##### **4.1.1.2.2.1 Aplicación 1**

*Mover linealmente una caja, dependiendo del peso de la caja se define el número de robot a utilizar*

#### **Solución:**

Para dar una solución a la problemática plantea, primero se toma en cuenta los elementos disponibles de los robots, se mencionó en el capítulo tres el limitado número de elementos, en este caso se utilizaron los siguientes elementos:

- Sensor Ultrasónico
- Sensor de Contacto
- Sensor de Pistas IR
- Sensor de Rotación del Motor (Encoder)
- Actuadores: Motor, Luces

- Solo se pueden ensamblar 2 robots con estos elementos.

Con los elementos mencionados se dio una solución al problema, a continuación se describe:

1. El robot 1 y el robot 2 se encuentran en una posición específica en el área de trabajo y desde este punto inicia el algoritmo.
2. El robot 1 debe conocer la proximidad con la pared, así evita posibles impactos con la misma, se utiliza el sensor ultrasónico, el cual envía el valor sentido al robot, si está a 5 [cm] de la pared se detiene, es un mecanismo de protección de choques.
3. El robot 1 se mueve en forma lineal, dentro del área de trabajo hasta encontrar la caja a mover, una vez encontrado el robot 1 se detiene para enviar una señal al momento de chocar el sensor de contacto con la caja. (el algoritmo general sugiere utilizar un sensor de fuerza, sin embargo en la aplicación no se cuenta con uno, "TANQUE" fue acoplado con un sensor de contacto).
4. Con la señal enviada al robot 1 realiza un análisis de si la caja es pesada o no:
  - Enceran los sensores de rotación de movimiento (Encoder).
  - Se activan los Encoder y envía una señal a los motores por un lapso de un segundo.
  - Si el valor del Encoder de vuelve como resultado un número mayor a 2, se identifica a la caja como un objeto liviano.
  - Si la caja es liviano se enciende la luz derecha y el robot la mueve linealmente.
  - Si la caja es pesada envía una señal vía Bluetooth al robot 2 para mover en conjunto la caja.
5. Si la caja es liviana el robot 1, mueve la caja 50[cm] en línea recta.
6. Si la caja es pesada, después del envío de la señal Bluetooth, el robot 1 se posiciona en un extremo de la caja, y espera al robot 2.
7. El robot 2 llega a su ubicación en el otro extremo de la caja y envía una señal indicando encontrarse en posición para mover la caja, se sincronizan.

8. Cuando se han sincronizado el robot 1 y el robot 2, empiezan a mover la caja en forma lineal una distancia de 50[cm], una vez terminada esa acción terminan su trabajo.

*¿Cómo saber cuál es el punto adecuado donde el robot debe chocar para mover la caja?*

Para responder esta pregunta realizaremos un experimento y un estudio profundo acerca de la fuerza y el punto de aplicación el cual influyen en el movimiento del cuerpo, para existir el movimiento de un cuerpo en una determinada dirección, la fuerza que lo empuja debe ser capaz de vencer la fuerza de rozamiento, para lo cual se debe encontrar el centro de masa, este término se refiere al lugar o punto donde está concentrada toda la masa de un cuerpo.

#### 4.1.1.2.1.1 Etapa 1

##### ➤ *Análisis Matemático Caja Liviana:*

El análisis matemático se procede a realizar con una caja liviana, Figura IV-7: Análisis General de la Caja Liviana, donde se puede visualizar el centro de caja donde debe topar el robot.

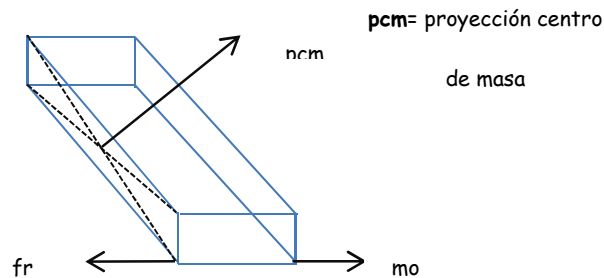
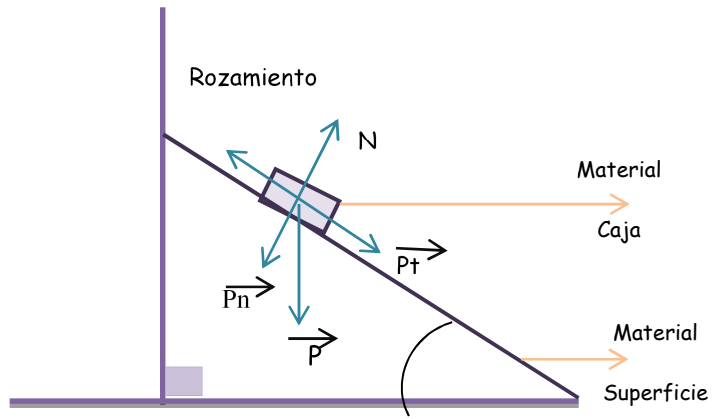


Figura IV-7: Análisis General de la Caja Liviana

##### ➤ **Fuerza de Rozamiento:**

La fuerza de rozamiento, se produce cuando dos cuerpos están en contacto, para calcular la fuerza de rozamiento en el presente proyecto, se realizó el siguiente experimento:

En un plano inclinado, para encontrar el coeficiente de rozamiento, véase Figura IV-8: Análisis del Coeficiente de Rozamiento



**Figura IV-8: Análisis del Coeficiente de Rozamiento**

(1)

(2)

(3)

(4)

(5)

Como el coeficiente de rozamiento aparece cuando no hay movimiento  $F_r = 0$

Por lo tanto, en el eje de las "X"

(7)

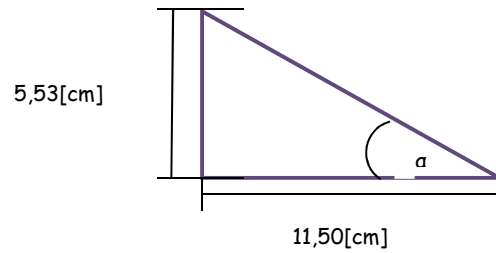
(8)

Para el proyecto el triángulo formado fue, Figura IV-9: Análisis del Coeficiente de Rozamiento de Material Tabla Prensada.

—

(9)

(10)



**Figura IV-9: Análisis del Coeficiente de Rozamiento de Material Tabla Prensada**

Una vez encontrado el coeficiente de rozamiento, se procedió a encontrar la fuerza de rozamiento:

$$Fr = \mu \times N \quad (11)$$

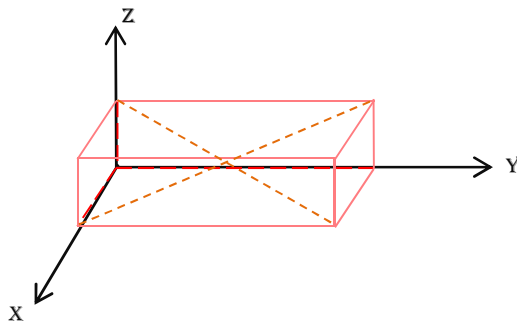
$$Fr = \mu \times m \times g \quad (12)$$

$$Fr = 0,522 \times 0,454kg \times 9,81 \frac{m}{s^2} \quad (13)$$

$$Fr = 2,32[N] \quad (14)$$

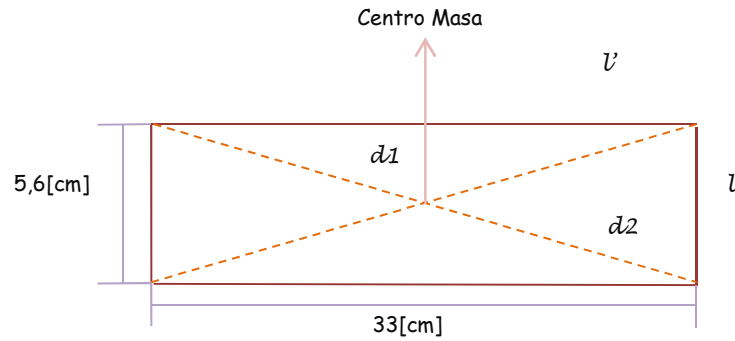
#### ➤ Centro de Masa

En el proyecto en particular la caja a mover, es homogénea es decir su densidad es uniforme, por consecuente el centro de masa con el centro geométrico son el mismo véase Figura IV-10: Análisis del Centro de Masa Caja Liviana para calcular se realizó el siguiente procedimiento:



**Figura IV-10: Análisis del Centro de Masa Caja Liviana**

*Cara de la caja donde choca el robot:*



**Figura IV-11: Cara A de la Caja Liviana**

$$\text{-----} \quad (15)$$

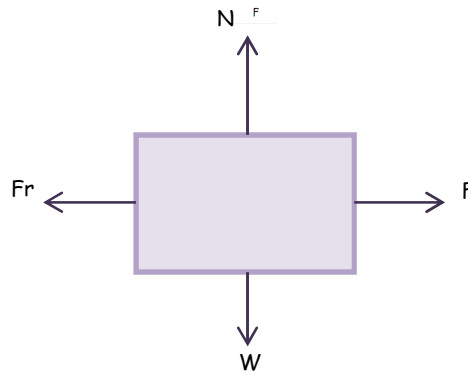
$$\text{-----} \quad (16)$$

$$(17)$$

El cruce de las diagonales Figura IV -11: Cara A de la Caja Liviana, da como resultado 16,74[cm] este valor es el centro del lado de la cara donde va a chocar ese punto tiene la misma dirección del centro de masa lo cual indica para este caso en particular si el punto posee la misma dirección del centro de la masa no se tiene problemas al momento de mover la caja.

➤ **Fuerza necesaria para vencer la fuerza de rozamiento**

Para romper la fuerza de rozamiento se toma en cuenta la siguiente Figura IV-12: Diagrama de Cuerpo Libre.



**Figura IV-12: Diagrama de Cuerpo Libre**

$$\Sigma F = m \times a \quad (18)$$

$$F - Fr = m \times a \quad (19)$$

$$F = m \times a + Fr \quad (20)$$

Se analiza el movimiento en este caso es uniformemente variado inicia con una velocidad constante y al momento de chocar con la caja su velocidad cambia, experimentalmente se determinaron los siguientes datos:

	Tiempo[s]	Distancia[cm]
	4,35	61
	4,38	61
	4,50	61
	4,32	61
	4,36	61
	4,36	61
	4,35	61
<b>Promedio</b>	4,37	61

Tabla IV-5: Datos Experimentales

Por consecuente la velocidad es 0,139 [m/s].

Con los datos obtenidos en la Tabla IV 5: Datos Experimentales se procede a utilizar la siguiente fórmula permitiendo encontrar la aceleración.

$$d = V_0 t - \frac{at^2}{2} \quad (21)$$

$$d = \frac{2V_0 t - at^2}{2} \quad (22)$$

$$2d - 2V_0 t = -at^2 \quad (23)$$

$$a = \frac{2V_0 t - 2d}{t^2} \quad (24)$$

$$a = 0,002 \frac{m}{s^2} \quad (25)$$

Con el valor de la aceleración se puede calcular la fuerza necesaria para romper la fricción con la fórmula 21

$$F = 0,454 \times 0,002 + 2,32 \quad (26)$$

$$F = 2,32[N] \quad (27)$$

#### 4.1.1.2.2.1.2 Etapa 2

##### ➤ Análisis Matemático Caja Pesada:

Para mover la caja en este caso se necesita dos robots, por lo tanto se buscaron dos centros de masa, véase Figura IV-13: Análisis General de la Caja Pesada

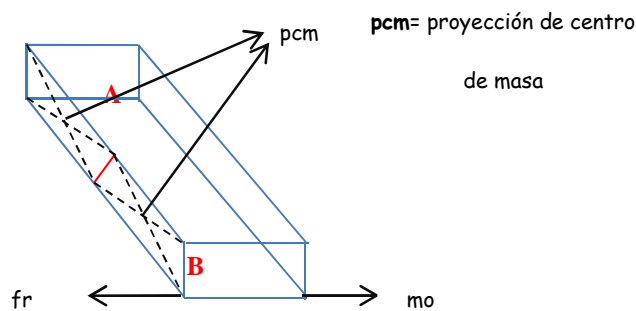


Figura IV-13: Análisis General de la Caja Pesada

##### ➤ Fuerza de Rozamiento

El coeficiente de fricción es el mismo determinado en la fórmula 11, el material utilizado es el mismo, por consecuente solo se calculó la fuerza de rozamiento.

$$Fr = \mu \times N \quad (28)$$

$$Fr = \mu \times m \times g \quad (29)$$

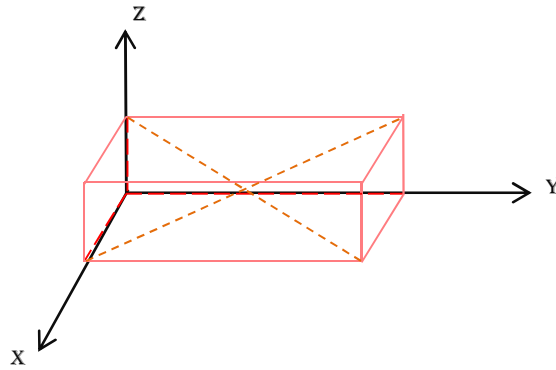
$$Fr = 0,522 \times 0,604kg \times 9,81 \frac{m}{s^2} \quad (30)$$

$$Fr = 3,09[N] \quad (31)$$



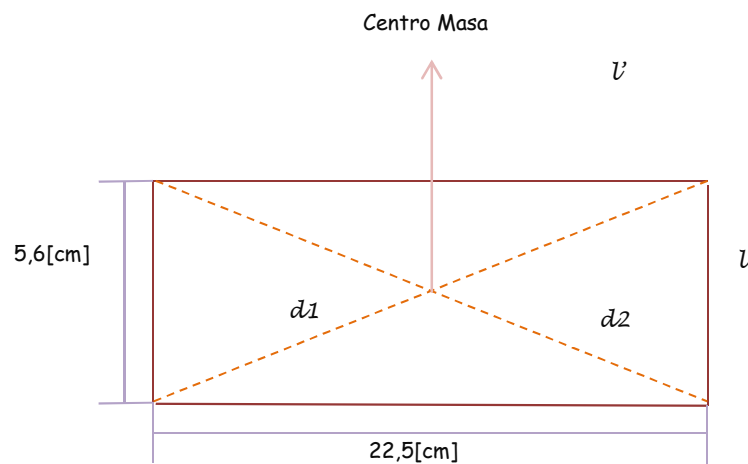
### ❖ Centro de Masa

Para encontrar los centros de masa en la caja se procedió a dividir la caja en dos partes con iguales dimensiones, véase Figura IV-14: Análisis del Centro de Masa Pesada, las dimensiones totales de la caja son largo 45[cm], ancho 16[cm] y profundidad 5,6[cm].



**Figura IV-14: Análisis del Centro de Masa Pesada**

Cara de la caja donde choca el robot, véase Figura IV-15: Cara A de la Caja Pesada



**Figura IV-15: Cara A de la Caja Pesada**

$$d1 = d2 = \sqrt{l^2 + l'^2} \quad (32)$$

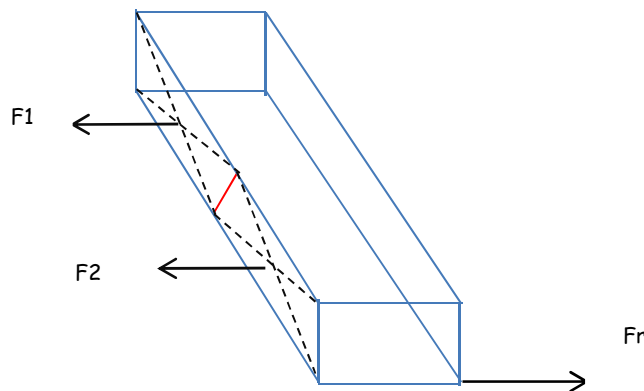
$$d1 = d2 = \sqrt{5,6^2 + 22,5^2} \quad (33)$$

$$d1 = d2 = 23,09 [cm] \quad (34)$$

El cruce de las diagonales da como resultado 11,55 [cm] y este es el centro del lado de la cara donde va a chocar ese punto tiene la misma dirección del centro de masa, se debe tomar en cuenta, inicialmente se dividió el cuerpo a la mitad por lo tanto es el mismo valor para el lado A como para el lado B lo cual indica en este caso en particular si el punto posee la misma dirección del centro de la masa no se tiene problemas al momento de mover la caja.

➤ **Fuerza necesaria para romper la fuerza de rozamiento**

Con la fórmula 25 se calculó la aceleración dando como resultado  $0,002 \frac{m}{s^2}$  y con la fórmula 21 se procede a calcular la fuerza tomando en cuenta la división de la caja, Figura IV-16: Fuerza Necesaria para Mover la Caja:



**Figura IV-16: Fuerza Necesaria para Mover la Caja**

Con la fórmula 29 da como resultado la fuerza de fricción total es 3,09 [N], en el primer experimento se llega a definir la fuerza máxima para un solo robot está es 2,32[N] por ende se necesita de dos robots para romper la fuerza de fricción, en la figura IV-15 se observa los puntos donde se ejercerá la fuerza.

$$\sum F > Fr \quad (35)$$

$$F1 + F2 > Fr \quad (36)$$

$$2,32[N] + 2,32[N] > 3,09[N] \quad (37)$$

$$4,64[N] > 3,09[N] \quad (38)$$

Con la sumatoria de fuerzas se demostrará a los dos robots, los cuales juntos rompen la fuerza de fricción y por ende se puede realizar el movimiento deseado, se procedió a programar y realizar las pruebas en el capítulo 5 en el cual analizan los resultados obtenidos.

#### **4.1.1.2.2.2 Aplicación 2**

##### ***Mover una caja con 2 elementos robóticos en forma circular***

Se pretende mover la caja en forma circular con los dos miembros del sistema multi – robot sin importar el peso de la misma, para realizar la tarea se necesita de coordinación entre ambos elementos, demostrando robótica cooperativa.

##### **Solución**

Dando una solución al problema planteado se toma en cuenta los dispositivos disponibles por los robots:

- Sensor de Contacto
- Sensor de Pistas IR
- Sensor de Rotación del Motor (Encoder)
- Actuadores: Motor, Luces

Con estos elementos se realizó un algoritmo el mismo se describe a continuación:

1. Los robots se los coloca en posición inicial, frente a la caja la acción de movimiento se realizará en el momento los pulsadores envíen la señal, la misma coloca al sistema operativo.
2. Con la señal proporcionada por los pulsadores se envía vía Bluetooth y comienza la sincronización de los robots.
3. Una vez sincronizado se enceran los sensores de rotación de movimiento (Encoder), y los equipos comienzan el movimiento circular.
4. Después de recorrer aproximadamente la cuarta parte del círculo, los robots se detienen y empieza la verificación, determinando de esta forma si las velocidades deben cambiar o no.

5. A la mitad del recorrido se realiza el paso anterior y así cada cuarta parte del círculo recorrido.

*¿Cómo saber cuáles son las velocidades adecuadas para realizar el recorrido circular de los equipos sincronizados?*

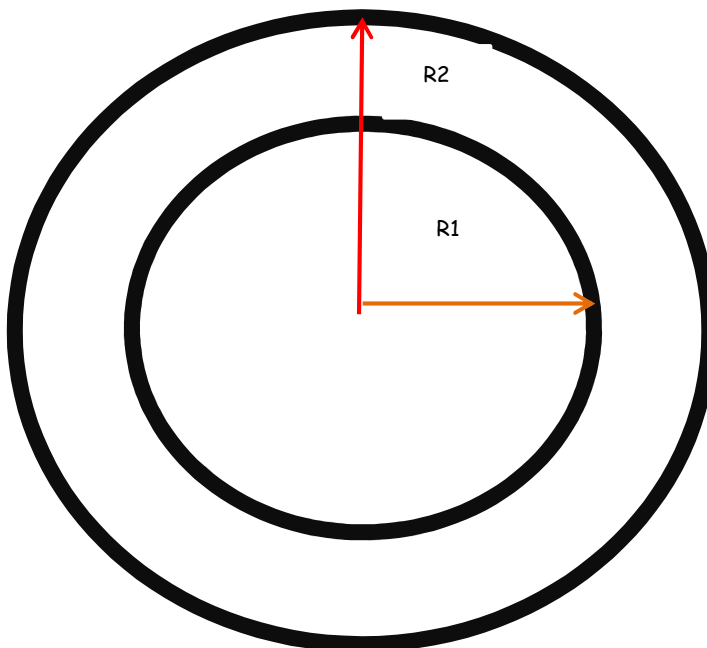
Respondiendo a esa pregunta se realizó el análisis matemático determinado de esta forma las velocidades y las fuerzas involucradas en este experimento como son la fuerza de fricción y fuerza centrífuga a continuación se describe el análisis.

***Análisis Matemático velocidades centrípetas:***

La centrífuga es una fuerza ficticia, tiende alejar a los objetos del eje de rotación.

Por consecuente la fuerza de rozamiento y la fuerza centrífuga deben ser iguales para mover al objeto en este caso la caja dentro del círculo.

El área de trabajo cuenta con dos círculos de diferentes diámetros, el primer círculo su diámetro es 92 [cm], y el segundo círculo cuenta con un diámetro de 54[cm].



**Figura IV-17: Área de Trabajo para el Experimento 2**

Como se menciona la  $F_r$  deben ser igual  $F_{cf}$  las fórmulas son las siguientes:

$$F_r = F_{cf} \quad (39)$$

$$\mu \times m \times g = m \times \omega^2 \times R \quad \mu \times g = \frac{v^2}{R^2} \times R \quad (40)$$

$$\mu \times g = \frac{v^2}{R} \quad (41)$$

$$v = \sqrt{\mu \times g \times R} \quad (42)$$

$$v = 0,94 \frac{m}{s} \quad (43)$$

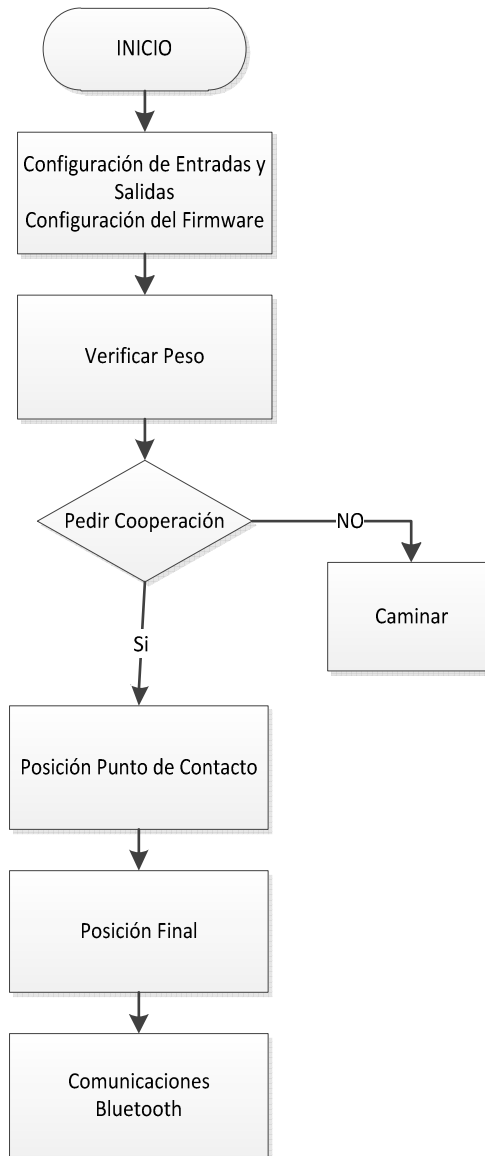
Con la fórmula 43, se puede determinar la velocidad con la cual el “TANQUE” A debe ir para calcular la velocidad a la cual debe ir el “TANQUE” B, se utiliza la fórmula 43 con la variación del radio.

$$v = 0,72 \frac{m}{s} \quad (44)$$

Una vez determinadas las velocidades se procedió a programar y realizar las pruebas en el capítulo 5 en el cual se analizan los resultados obtenidos.

#### 4.1.1.3 Diagrama de Flujo

A continuación se presentan el diagrama de flujo figura IV-18: Diagrama de Flujo del Sistema Multi - Robot



**Figura IV-18: Diagrama de Flujo del Sistema Multi – Robot**

## 4.1.2 Programación

### 4.1.2.1 Introducción

La programación es el proceso de diseñar, codificar, depurar y mantener el código fuente en un lenguaje de programación, cuyo propósito es crear programas los mismos exhibe un comportamiento deseado.

El controlador del robot “TANQUE”, es versátil en lenguajes de programación, el mismo puede ser programado en Visual C++, C, C#, Java, Labview, mediante librerías creadas en su mayoría por Knobloch, para Windows.

#### **4.1.2.2 Modos de programación**

El controlador de “TANQUE”, usa diferentes librerías de programación para distintos usos, dependiendo de las necesidades del usuario se puede ejecutar la programación en 2 modos:

- Modo en línea
- Modo autónomo (Down load).

##### **4.1.2.2.1 Modo en línea**

El programa es ejecutado en un computador conectado a la interfaz de programación mediante el puerto serie con protocolo RS-232. El programa envía y recibe datos y realiza las operaciones programadas en los lenguajes enunciados anteriormente.

Su ventaja es el acceso a la memoria del computador, los algoritmos de programación implementados en este pueden ser complejos y su capacidad de respuesta depende únicamente del computador, su desventaja es la dependencia total de una computadora, si se desconecta el controlador de la computadora este no sabrá las acciones a ejecutarse, para este modo se utilizó el lenguaje de programación Visual C++.

##### **4.1.2.2.1.1 Programación basada en Visual C++**

El controlador de “TANQUE” se lo puede programar mediante software de alto nivel como visual C++ 2010, para esto se utiliza la librería ftMscLib.dll (Se lo puede encontrar en el anexo [2]), esta librería permite establecer la transferencia de datos del computador al controlador, esta librería soporta solo comunicación en línea con el controlador.

La comunicación es por cable USB o Bluetooth, cualquiera de los 2 medios genera un COM virtual, mediante el cual se enlaza el firmware (es el software del controlador, las funciones del sistema operativo, como el programa de arranque del controlador) y la computadora.

El controlador cuenta con el sistema operativo 4NetOS ® mayor información anexo [2], el firmware se ejecuta constituye la base del sistema multitarea, tiempo real del sistema operativo. Las diferentes tareas del sistema:

- Lectura de las entradas y salidas.
- Sincronización de datos de área de transferencia con el PC o con las tarjetas esclavas

- Manejo de las interfaces en serie (incluido el USB y Bluetooth)
- Gestión del sistema de archivos
- Lectura interruptores de pulsador y la visualización de la salida
- La ejecución de un programa local de la robótica, si está presente y se inicia, en la división de tiempo operación multiplexado con otras tareas
- Otras tareas internas

#### **4.1.2.2.1.2 Requerimientos del Computador para C++**

- Sistema Operativo Windows Xp o Superior
- Tipo de Sistema 32 bits o 64 bits
- Memoria RAM de 2 GB

Como antes se lo menciona no es factible crear un programa en visual C++, por la dependencia del computador por lo tanto para el presente proyecto se utiliza el modo en ejecución.

#### **4.1.2.2.2 Modo autónomo**

El programa requiere ser transferido y almacenado en la interfaz de programación para su posterior ejecución, sin la necesidad de estar permanentemente conectada. Este modo es especialmente útil para la elaboración de prototipos móviles.

Este modo permite independencia de la computadora, su principal desventaja es la limitación en memoria (Flash 2 MB), los algoritmos implementados en este modo son simples por sus características de procesamiento, su ventaja es la independencia del controlador, es decir decide los pasos a seguir en base a las reglas previamente cargadas a su controlador sin la necesidad de pedir permiso a computador, en este modo se utiliza el lenguaje de programación C



#### 4.1.2.2.2.1 Programación en C

Un programa creado en C, se lo puede cargar al controlador de “TANQUE”, utilizando yagarto el cual contienen las herramientas ARM – GCC.

Yagarto es un conjunto de herramientas de GNU para arquitecturas de sistemas embebidos ARM para Windows. ARM, es una arquitectura RISC (Ordenador con Conjunto de Instrucciones Reducidas), se llamó Advanced RISC Machine es el conjunto de instrucciones reducidas de 32 bits más ampliamente utilizado en microprocesadores, microcontroladores pequeños de bajo consumo.

GCC es una Colección de Compiladores de GNU, es un conjunto de compiladores creador por el proyecto GNU, estos se los considera estándar para los sistemas derivados de UNIX de código abierto y también de propietarios.

GNU es un sistema operativo completamente libre, pero en la actualidad no se encuentra implementado completamente, parte de este proyecto es GCC fue escrito para la compilación del sistema operativo GNU desarrollado como un software libre, GCC en la actualidad es un compilador libre para C y C++, este es compilador portable y necesita de un editor de C.

Las Herramientas GNU ARM es una solución completa para el desarrollo de procesadores ARM en compilador GCC para programas escritos en lenguaje C.

GCC es un compilador portable, en la actualidad se ejecuta en la mayoría de plataformas disponibles, y puede producir ejecutables para diferentes tipos de procesadores, micro controladores, DSP's, no es solo un compilador nativo de C, también puede compilar cualquier programa, produce ficheros ejecutables, a sí se puede compilar software para sistemas embebidos, GCC está escrito en C enfocado a la portabilidad y con la facilidad de compilar a sí mismo, tiene un diseño modular le permite soportar nuevos lenguajes y arquitecturas (GNUPro, 2010)

GCC es software libre distribuido bajo GNU, con Licencia Pública General, teniendo la libertad para modificar GCC como son el software de GNU, permitiendo crear una nueva funcionalidad de manera personalizada.

Las GNU Binary Utilities, o binutil, son herramientas de programación para la manipulación de código objeto, el ELF ejecutado y enlazador de código objeto, el mismo soporta compilaciones cruzadas, y el encabezado la arquitectura utilizada generan

herramientas descritas en la Tabla IV-6: Herramientas de GNU para generar código binario leído por el controlador.

Descripción		Arquitectura	
		ARM	Descripción
GCC	Compilador	arm-elf-gcc	
GAS	Ensamblador	arm-elf-as	
GLD	Enlazador	arm-elf-ld	
Binarios		arm-elf-ar	crea modifica y extra desde archivos
		arm-elf-objcopy	copia archivos objetos
		arm-elf-objdump	Vuelca información sobre archivos objetos
		arm-elf-ranlib	genera índice para archivos
		arm-elf-readelf	muestra el contenido del archivo ELF
		arm-elf-size	lista el tamaño y secciones
		arm-elf-strings	lista los caracteres imprimibles
		arm-elf-strip	elimina símbolos del archivo objeto
GBD	Depurador	arm-elf-gdb	

**Tabla IV-6: Herramientas de GNU**

Compilar un archivo de C mediante GCC y obtener un ejecutable en formato Bin.

Se editó el archivo en formato .C en visual C 2010, y mediante la herramienta ARM ELF GCC, se compila y se verifica la existencia de errores, luego el comando ARM ELF LD se enlaza y se obtiene un archivo objeto, luego mediante el comando ARM ELF OBJCOPY se obtiene el ejecutable .Bin será cargado al controlador mediante el software ROBOTEXTtest

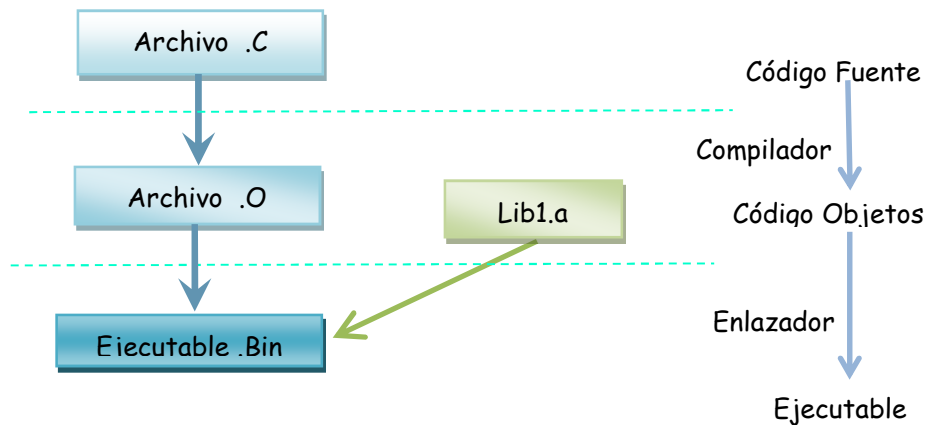


Figura IV-19: Fases para la Compilación y Ejecutable en Código Nativo

#### 4.1.2.2.2.2 Requerimientos del Computador para C

- Yagarto
- Editor de C (Visual C 2010)
- RoboTxTest.exe (Se lo puede encontrar en [34])
- Sistema Operativo Microsoft Windows Xp

#### 4.1.2.2.2.3 Instalación

Se instala el yagarto en el disco C y se descarga el archivo ftMscCComp, se lo guarda en la disco C, en esta carpeta se encuentran 3 archivos básicos necesarios para crear cualquier ejemplo nuevo,

Clean.bat es un archivo creado por el fabricante de tipo Make File, permite eliminar los archivos de enlazador, ensamblador, compilador.

Make.bat es un archivo creado por el fabricante de tipo Make File, permitiendo crear ejecutables, este archivo permite enlazar el editor de C, el compilador GCC y las herramientas binutil, devolviendo un archivo ejecutable en el controlador con la extensión .Bin

Param.mk Es un archivo este lleva los parámetros a crear, debe incluirse el nombre del archivo con estación en el formato .C y el archivo de salida con extensión .O, (GCC compila archivos .C y genera ejecutables .O).

### 4.1.2.3 Diagrama Estructurado del Programa

El diagrama estructurado permite tener una visión global de la programación, a continuación se enuncian los diagramas utilizados:

- Tabla IV 7: Diagrama Estructurado Inicialización
- Tabla IV 8: Diagrama Estructurado de Verificación de Peso
- Tabla IV 9: Diagrama Estructurado de Posición Punto de Contacto
- Tabla IV 10: Diagrama Estructurado de Posición Final de Robot Antes de Mover Una caja Pesada
- Tabla IV 11: Diagrama Estructurado de Comunicaciones Bluetooth
- Tabla IV-12: Diagrama Estructurado de Caminar

Inicio
Variable Inicio de Comunicación Controlador – Programa
Configuración de Salidas
Configuración de Entradas Universales
Configuración de Entradas Contador
Configuración del Firmware
Variable Sensor _ultrasónico ← 0
Variables Sensor_encoders ← 0
Variable Sensor_contacto ← 0
Variable Mensaje ← 0
ESTADO ← inicio

**Tabla IV-7: Diagrama Estructurado Inicialización**

Verificación de Peso					
ESTADO ← inicio					
Evaluar(ESTADO)					
Caso <b>Inicio</b>		Caso <b>Tope</b>	Caso <b>to_probrar</b>	Caso <b>To_verificacion</b>	
Energizar Motores		Parar Motores	Energizar Motores (1s)	Valor sensor_encoder <b>V</b> >2 <b>F</b>	
Dezplazar al frente			Tomar valor Sensor Encoder		
Variable sensor_ultrasonico <b>V</b> >=7 <b>F</b>		Encerarrar Variable de Sensor Encoder			
Seguir Linea	Parar Motores				
valor sensor_contacto <b>V</b> ==1 <b>F</b>		Encerar variables	Salta <b>to_probrar</b>	Salta <b>to_verificación</b>	Salta <b>Caminar</b> Salta <b>Retroceder</b>
Saltar <b>Tope</b>	Seguir Linea	Finalizaci ón			

Tabla IV-8: Diagrama Estructurado de Verificación de Peso

Posicionar en un Punto de Contacto al Robot		
Valor Sensor_encoders		
ESTADO ← to_verificar		
Evaluar(ESTADO)		
Caso <b>Retroceder</b>	Caso <b>Girar</b>	Caso <b>distancia_atraz_1</b>
Encerar variable sensor_encoder	Encerar variable sensor_encoder	Encerar variable sensor_encoder
Energizar Motores	Mover llantas diferencialmente (3s)	Energizar Motores
Desplazar al frente (6cm)		Desplazarse Atraz (3cm)
Salta <b>Girar</b>	Parar Motores	Salta <b>giro_posicion</b>
	Salta <b>distancia_atraz_1</b>	

Tabla IV-9: Diagrama Estructurado de Posición Punto de Contacto

Posicionamiento Final		
ESTADO ← giro_posición		
Evaluar(ESTADO)		
Caso <b>giro_posicion</b>	Caso <b>posicion_final</b>	
Encerar variable sensor_encoder	Desplazar al frente	
Mover llantas diferencialmente (3s)	Valor sensor_contacto <b>V</b> <b>F</b> ==1	
Salta <b>posicion_final</b>	Salta <b>BT_CONECTAR</b>	Salta <b>posicion_final</b>

**Tabla IV-10: Diagrama Estructurado de Posición Final de Robot Antes de Mover Una caja Pesada**

Comunicaciones Bluetooth		
Variable mensaje		
ESTADO ← BT_CONECTAR		
Evaluar(ESTADO)		
Caso <b>BT_CONECTAR</b>	Caso <b>BT_ENVIA_SOLICITUD</b>	Caso <b>BT_RECIBE_MENSAGE</b>
Enlazando con Bluetooth	Enviar cadena de Caracteres	Leer Función Respuesta
Verificación Comunicaciones	Leer Función Recibir	Encerar variable sensor_encoder
Esperando Respuesta	Espera Respuesta de Sincronización	Desplaza al frente(50cm)
Salta <b>BT_ENVIA_SOLICITUD</b>	Salta <b>BT_RECIBE_MENSAGE</b>	Encerar Variable Finalizar

**Tabla IV-11: Diagrama Estructurado de Comunicaciones Bluetooth**

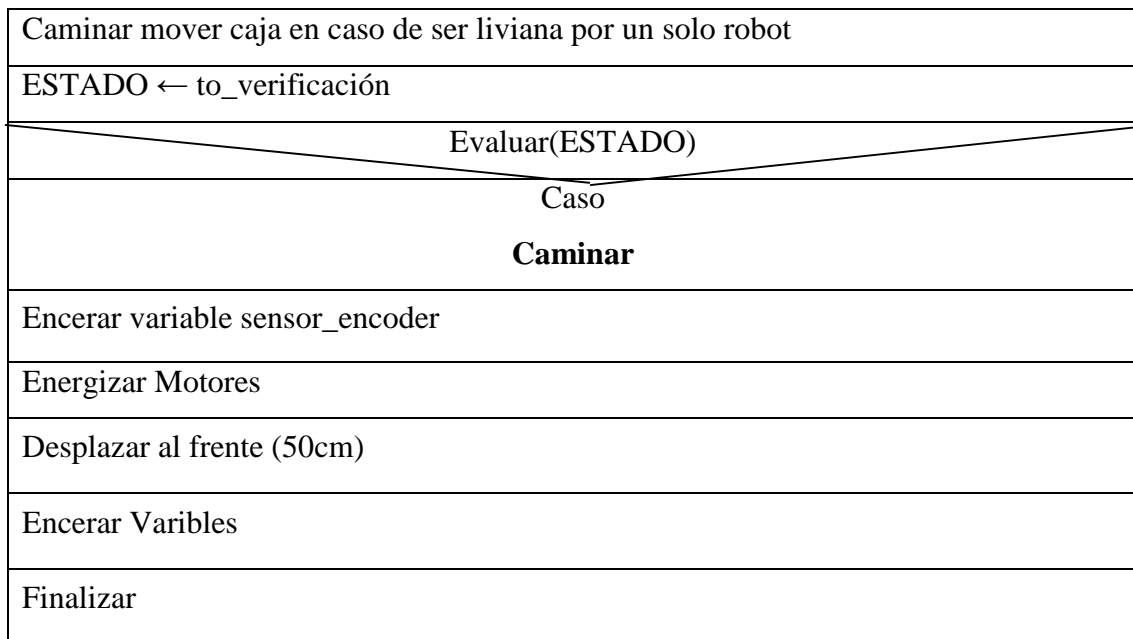


Tabla IV-12: Diagrama Estructurado de Caminar

#### 4.1.2.3.1 Variables Utilizadas en el Programa

En la Tabla IV 13: Variables y Descripción, se detalla el nombre, el tipo y descripción de cada una de las variables utilizadas en la programación de “TANQUE”.

Variable	Tipo	Descripción
envio_estado;	static INT16	Envía la señal al agente 2, para pedir ayuda.
i_intencidad_luz;	static INT16	Variable auxiliar para verificar comunicación Bluetooth
i_tamano_caja;	static INT16	Envía al Agente 2 el tamaño de la caja
a_ultrasonico;	static INT16	Obtiene el valor del sensor ultrasónico
i_estado_I_8;	static INT16	Obtiene el valor de la entrada I_8
c_Encoder_1;	static INT16	Obtiene el valor del sensor movimiento del Rotor 1
c_Encoder_2;	static INT16	Obtiene el valor del sensor movimiento del Rotor 2
Peso_caja;	static INT16	Determinar la cantidad de agentes necesarios

Tabla IV-13: Variables y Descripción Parte 1

Variable	Tipo	Descripción
d_sens_contacto;	static INT16	Obtiene el valor del sensor de contacto
arrancar;	static INT16	Sincroniza el arranque con el agente 2
pausa;	static INT16	Tiempo de espera para una acción
tiempo_espera;	static INT16	Tiempo de espera para probar el peso de la caja
tiempo_prueba;	static INT16	Prueba por 40 ms sí la caja es pesada
aux_u_mapeocaja_2;	static INT16	Retrocede 5 cm
aux_u;	static INT16	Auxiliar del sensor ultrasónico
d_rastreador1;	static INT16	Obtiene el valor del sensor de pista entrada I5
d_rastreador2;	static INT16	Obtiene el valor del sensor de pista entrada I6
activacion;	static INT16	Activa él envió de datos
foco;	static INT16	Auxiliar verifica la comunicación con el agente 2
aux_dis;	static INT16	Variable auxiliar para intervalos de movimiento lineal luego de la sincronización
aux_recto;	static INT16	Variable auxiliar para intervalos de movimiento lineal
timer_arra;	static INT16	Tiempo de arranque de los motores
sincronizacion;	static INT16	Variable permite sincronizar a los 2 agentes
BT_ESTADO_MSG;	static enum bt_commands_e	Variable de los estados del programa
BT_ESTADO_CONECION;	static CHAR8	Verifica el estado de conexión Bluetooth
BT_RECIVE_ESTADO_CONEC;	static CHAR8	Estado de la comunicación con el agente 2
BT_RECIVIR;		Variable permite recibir datos vía Bluetooth
DYS_MENSAJE[128];	static char	Mensaje a enviar al pantalla del controlador

Tabla IV-14: Variables y Descripción Parte 2



#### 4.1.2.4 Funciones de la Librería ftMscLib

Para programar en el controlador se cuenta con una librería, a continuación se describen las principales funciones para mayor información anexo [2].

##### 4.1.2.4.1 Funciones Principales

En las siguientes tablas se describen las funciones y la acción que realiza cada una:

- Tabla IV 15: Funciones de Inicialización
- Tabla IV 16: Funciones de Configuración
- Tabla IV 17: Funciones del Programa Parte 1
- Tabla IV 18: Funciones del Programa Parte 2
- Tabla IV 19: Funciones de Bluetooth

Nombre Función	Acción
<b>ftxGetLibVersion</b>	La versión de la librería ubicada en el controlador
<b>ftxInitLib()</b>	Función de Inicialización de la librería, retorna si existe errores en la librería
<b>ftxOpenComDevice</b> (Número COM, Velocidad, errores)	La función abre la comunicación con el controlador, cuando se especifica la interfaz de un COM, el cual puede ser leído como un strings.
<b>ftxCloseDevice</b> (controlador)	Función de cierra de comunicación abierta por un controlador específico, para todos los hilos de comunicación
<b>ftxCloseAllDevices()</b>	Función de cierra de comunicación abiertas para todos los controladores encontrados.
<b>ftxStartTransferArea</b> (controlador)	Activa la transferencia en la librería para el modo on line, este lee los valores de corriente en las salidas para la transferencia (configuración de los valores de salida) y envía la configuración de las entradas.
<b>ftxStopTransferArea</b> (controlador)	Deshabilita la transferencia de datos con el controlador, para la comunicación.
<b>ftxIsTransferActiv</b> (controlador)	La función chequea si la transferencia de datos fue activada.

**Tabla IV-15: Funciones de Inicialización**

Nombre Función	Acción
<b>StartCounterReset</b> (controlador, Identificación del controlador ,número de contador)	Función utilizada para Reset las entradas de los contadores, este es un proceso asíncrono y es completado en el retorno de la llamada a la función.
<b>SetCBCounterReseted</b> (Identificación del controlador ,número de contador)	Función reporta el estado del reseteo del contador
<b>SetFtUniConfig</b> (controlador, Iden. controlador, número de entrada, modo de medición, digital/análogo)	Función de configuración de las entradas universales (entradas combinadas) análogas o digitales.
<b>SetFtCntConfig</b> (controlador, Iden. controlador, numero de contador, modo)	Configura las entradas a contadores según el estatus del contador. Modo 0 normal, Modo 1 invertido
<b>SetFtMotorConfig</b> (controlador, Iden. controlador, numero de motor, estado)	Función activa o desactiva las salidas a motor, análogas o PWM Estado TRUE motor, o Estado FALSE PWM.

Tabla IV-16: Funciones de Configuración

Nombre Función	Acción
<b>SetOutMotorValues</b> (controlador, Iden. controlador, número de motor, velocidad +, velocidad - )	La función setea los valores de velocidad de los motores tanto en alto como en bajo.
<b>SetOutPwmValues</b> (controlador, Iden. controlador, número de motor, velocidad)	La función setea el valor de velocidad para un PWM

Tabla IV-17: Funciones del Programa Parte 1

Nombre Función	Acción
<p><b>GetInIOValue</b></p> <p>(Controlador, Iden. controlador, número de entrada, dato, sobrepaso el dato)</p>	<p>Función lee el valor de la entrada universal</p>
<p><b>GetInCounterValue</b></p> <p>(controlador, Iden. controlador, numero de contador, dato, estado)</p>	<p>La función lee el valor de la entrada al contador.</p> <p>Estado True = invertido</p> <p>Estado False = normal</p>
<p><b>StartMotorExCmd</b></p> <p>(Controlador, Iden. controlador, numero de motor, velocidad, dirección, número de motor 2, dirección, valor del contador.)</p>	<p>La función activa la inteligencia del motor, sincronizando los 2 motores, para moverse a una misma velocidad, usan la información del contador</p>
<p><b>StopMotorExCmd</b></p> <p>(controlador, Iden. controlador, número de motor)</p>	<p>La función para inmediatamente la sincronización de los motores, resetea el valor del contador.</p>
<p><b>StopAllMotorExCmd</b></p> <p>(Controlador, Iden. Controlador)</p>	<p>La función inmediatamente para la sincronización de todos los motores se ha activado en el controlador.</p>
<p><b>SetCBMotorExReached</b></p> <p>(Controlador, Iden. Controlador)</p>	<p>Función devuelve el estado de los motores sincronizados</p>
<p><b>GetInDisplayButtonValue</b></p> <p>(Controlador, Iden. Controlador, botón derecho, botón izquierdo.)</p>	<p>Función lee el estado de los 2 botones del Display al mismo tiempo en el cual se pulsa uno de ellos.</p>
<p><b>SetRoboTxMessage</b></p> <p>(Controlador, Iden. Controlador, mensaje.)</p>	<p>Función genera un texto visible en la pantalla del controlador el texto puede ser máximo de 98 caracteres ASCII, se pueden borrar los mensajes.</p>

**Tabla IV-18: Funciones del Programa Parte 2**

Nombre Función	Acción
<b>StartScanBtDevice</b> (Controlador, función de resultados)	Analiza los dispositivos Bluetooth en el alcance del controlador conectados, (no se debe confundir con la búsqueda interna de la pc), se utiliza para iniciar búsqueda de los dispositivos disponibles este proceso dura 30 segundos, y se lo puede hacer varias veces.
<b>CancelScanBtDevice</b> (Controlador)	Cancelación anticipada de la búsqueda de dispositivos Bluetooth.
<b>ConnectBtAddress</b> (Controlador, canal , dirección, función de resultado de conexión )	Establece una comunicación Bluetooth con otro dispositivo remoto, con una dirección de destino específica. Después de realizada la comunicación con éxito se gestiona para activar la comunicación mediante el canal.
<b>BtListenConOn</b> (Controlador, canal , dirección, función de resultado de conexión )	Establecimiento pasivo de una conexión vía Bluetooth, mediante un mando a distancia identificado por la dirección Bluetooth de origen específico, la devolución de llamada informa de manera asíncrona tan pronto como se inicie el estado.
<b>BtListenConOff</b> (Controlador, canal , función de resultado de conexión )	Esta función desactiva el estado de la función BtListenConOn ,esto significa la conexión Bluetooth ya no son aceptadas, y se establece con la llamada de nuevo a la función BtListenConOn
<b>DisconnectBt</b> (Controlador, canal , función de resultado de conexión )	Termina una conexión Bluetooth activada hace referencia al índice de canal específico no importa si la conexión se realizó activamente o pasivamente.
<b>SendBtMessage</b> (Controlador, canal, dimensión del dato, dato, función de resultados)	Escribe datos a una conexión para un canal especificado.
<b>BtReadMsgOn</b> (Controlador, canal, función )	Función devuelve el estado de los datos a recibir, en otra función se vuelve a crear los datos recibidos.,
<b>StatusBtConnection</b> (canal, puntero donde almacene el error)	Devuelve el estado actual de una conexión de Bluetooth, con el problema específico del fallo de la comunicación.

Tabla IV-19: Funciones de Bluetooth

En las siguientes tablas se describen las variables de la estructura, son importantes estas permiten comunicación con el controlador.

Las variables para la configuración de entradas universales, salidas a motor y entradas a contador se describen en la tabla IV-20,IV-21,IV-22.

Nombre	Tipo	Descripción
Motor	BOOL8[]	Especifica la configuración sí las salidas están definidas en 4 salidas a motor Modo = TRUE y 8 salidas digitales en Modo = FALSE
Unit	UNI_CONFIG[ ]	Especifica la configuración de las 8 entradas universales Modo = 0 digital = TRUE - digital 10[V] Modo = 0 digital = FALSE - analoga 10[V] Modo = 1 digital = TRUE - digital 5[kΩ] Modo = 1 digital = FALSE - analoga 5[kΩ] Modo = 3 digital = No importa - ultrasónico
Cnt	CNT_CONFIG[ ]	Especifica la configuración de los contadores Modo = 0 - normal Modo = 1 - invertido

**Tabla IV-20: Variables de Configuración**

Nombre	Tipo	Función
Unit	INT16[]	Constantes por cada entrada universal, el valor de corriente determinado por el firmware
cnt_in	INT16[]	Configuración de corriente de la entrada a contador
Counter	INT16[]	Estado corriente del contador
cnt_reseted	BOOL16[]	Se setea en 1 sí el contador se resetea satisfactoriamente
motor_pos_reached	BOOL16[]	Bandera para controlar el motor, sí la notificación proviene del objetivo, acondiciona las especificaciones de la bandera

**Tabla IV-21: Variables de Entradas**

Nombre	Tipo	Función
cnt_reset_cmd_id	UINT16[]	Reset contador, Puede ser incrementado por 1 cada tiempo si un reseteo es requerido
Master	UINT8[]	En el caso de activar el motor de control, el ID es determinar para el motor esclavo desde el motor master. El master ID en este caso el motor índice ( a 3) Ejemplo: Sincroniza el control del motor con motor 1 y motor 2
Duty	INT16[]	Valor de la velocidad para el motor de salida; sí el interruptor del motor está activado sobre cada parte la diferencia de los 2 valores correspondiente a la dirección de rotación
Distance	UINT16[]	Valor del contador, el valor es la posición del motor, este se para durante activación del control del motor. Motor control está activado solo si el valor de contador es distinto de cero
Motor_ex_cmd_id	UINT16[]	Siempre está incrementando por 1 sí se setea para activación del control del motor (velocidad / distancia).

**Tabla IV-22: Variables de Salidas**

## CAPÍTULO V .

### PRUEBAS Y RESULTADOS

En el presente capítulo se detalla las pruebas de trayectoria lineal y circular, se analiza resultados obtenidos, para demostrar la presión y exactitud del sistema.

#### 5.1 SISTEMAS MULTI – ROBÓTICOS PRUEBAS Y RESULTADOS

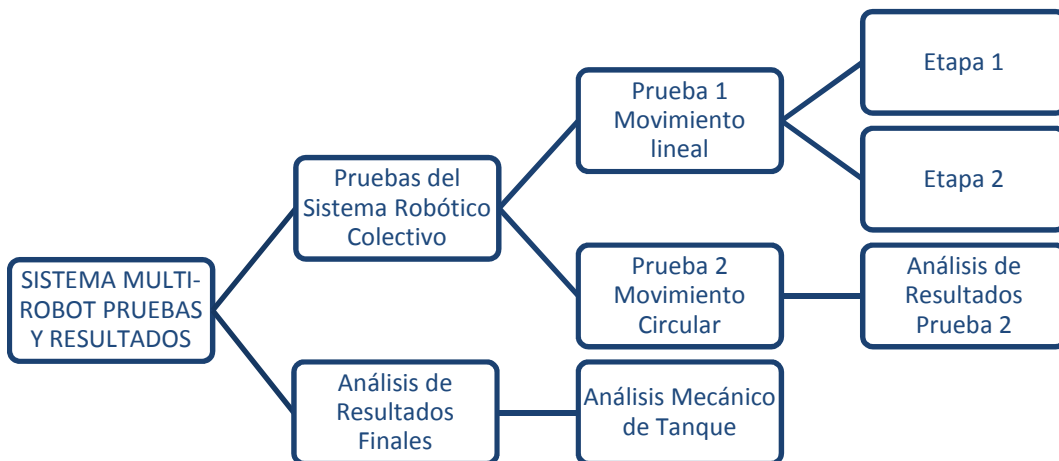


Figura V-1: Estructura del Capítulo V

##### 5.1.1 Pruebas del Sistema de Robótica Cooperativa

Una vez concluidos los algoritmos de programación, y el análisis de hardware se procedió a probar el sistema, para la cual se realizó las pruebas descritas a continuación:

### 5.1.1.1 Prueba 1: Movimiento lineal de una caja

En esta prueba se mueve el robot en una trayectoria lineal desde el punto A hasta el punto B y una caja del punto B al punto C, se muestra en la Figura V-2: Prueba 1 Movimiento Lineal de una caja, en la cual se explica la trayectoria a seguir en esta prueba.

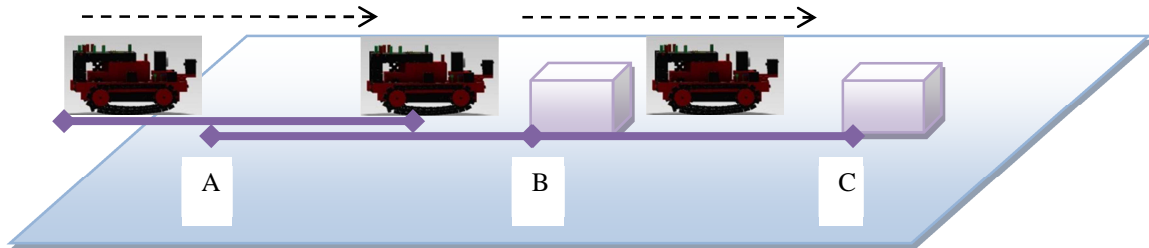


Figura V-2: Prueba 1 Movimiento Lineal de una caja

Esta prueba se la dividió en 2 etapas:

- ❖ Primera mover el robot desde el punto A hasta el punto B en una trayectoria lineal sin empujar objetos.
- ❖ Segunda mover el robot con una caja desde el punto B al punto C en una trayectoria lineal.

#### 5.1.1.1.1 Primera Etapa

Mover el robot desde el punto A hasta el punto B en una trayectoria lineal la misma se muestra en la Figura V-3: Prueba 1 Movimiento Lineal de un Robot, en esta prueba se determina la exactitud y precisión del robot al seguir una trayectoria lineal para lo cual se realizó la tabulación en la Tabla V 1: Puntos de partida del Robot, en la mismas se determina la trayectoria realizada por el robot para posteriormente analizarla.



Figura V-3: Prueba 1 Movimiento Lineal de un Robot



<b>Distancia eje Y partida</b>	8 cm
<b>Distancia eje X partida</b>	79,6 cm

**Tabla V-1: Puntos de partida del Robot**

Distancia eje y	Distancia eje x
8	79,6
12	79,7
15,5	79,5
19,5	79,7
24	79,6
27,5	79,6
31,1	79,6
36	79,7
39	79,4
43,5	79,5
48,5	79,6
Promedio	79,62

**Tabla V-2: Puntos seguidos por el robot mediante una trayectoria lineal**

Para determinar la precisión del robot se debe calcular la media aritmética con la ecuación 1

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (1)$$

$$\bar{x} = \frac{x_1 + x_2 + x_3 + x_4 + \dots + x_n}{n} \quad (2)$$

$$\bar{x} = \frac{79,6+79,7+79,5+79,9+79,6+79,6+79,6+79,7+79,3+79,5+79,6+79,6}{12} \quad (3)$$

$$\bar{x} = 79,62 \quad (4)$$

Cálculo del error absoluto de las mediciones tomadas se lo realiza en base a la fórmula 5, para lo cual se tomará el valor medio calculado en la fórmula 4.

$$\delta_x = \frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}| \quad (5)$$

$$\delta_x = \frac{|x_1 - \bar{x}| + |x_2 - \bar{x}| + |x_3 - \bar{x}| + |x_4 - \bar{x}| + \dots + |x_n - \bar{x}|}{n} \quad (6)$$

$$\delta_x = \frac{|79,6-79,62|+|79,7-79,62|+|79,5-79,62|+|79,9-79,62|+|79,6-79,62|+|79,6-79,62|+|79,6-79,62|+|79,7-79,62|+|79,3-79,62|+|79,5-79,62|+|79,6-79,62|+|79,6-79,62|}{12} \quad (7)$$

$$\delta_x = 0,05 \quad (8)$$

El error porcentual se calculó con una sola medición con la fórmula 9 y se tomará el valor del error absoluto calculado:

$$e_p(\%) = \frac{\delta_x}{x} * 100 \quad (9)$$

$$e_p(\%) = \frac{0,05}{79,5} * 100 \quad (10)$$

$$e_p(\%) = 0,5\% \quad (11)$$

Se muestra en la Figura V-4: Trayectoria Seguida por el Robot y la trayectoria correcta a seguir por el robot, los datos obtenidos en la Tabla V-2: Puntos seguidos por el robot mediante una trayectoria lineal, se ajusta mediante el método de mínimos cuadrados, del cual se obtiene una trayectoria lineal, se demuestra la precisión del robot debido al valor de su error, este es mínimo como se muestra en la fórmula 8, al ser su desviación relativamente pequeña, se demostró la inexactitud de “TANQUE” para seguir una línea.

Se puede ver en la Figura V-4: Trayectoria Seguida por el Robot, a “TANQUE” corrigiendo su trayectoria cuando se desvía la misma, se analizó la trayectoria descrita por

el método de ajuste por mínimos cuadrados , para la cual se utiliza la ecuación 12, describe la ecuación general de la recta.

I	$x_i$	$y_i$	$x_i y_i$	$x_i^2$	$y_i^2$	$b + m * x_i - y_i$
1	8	79,6	636,80	64,00	6336,16	0,00
2	12	79,7	956,40	144,00	6352,09	0,00
3	15,5	79,5	1232,25	240,25	6320,25	0,02
4	19,5	79,9	1558,05	380,25	6384,01	0,06
5	24	79,6	1910,40	576,00	6336,16	0,00
6	27,5	79,6	2189,00	756,25	6336,16	0,00
7	31,1	79,6	2475,56	967,21	6336,16	0,00
8	36	79,7	2869,20	1296,00	6352,09	0,00
9	39	79,3	3092,70	1521,00	6288,49	0,13
10	43,5	79,5	3458,25	1892,25	6320,25	0,03
11	48,5	79,9	3875,15	2352,25	6384,01	0,06
12	49,5	79,6	3940,20	2450,25	6336,16	0,00
$\sum_{i=1}^N$	354,1	955,5	28193,96	12639,71	76081,99	0,31

Tabla V-3: Tabulación de datos para el método de ajuste lineal

$$y = mx \pm b \quad (12)$$

Para la ecuación 12 es necesario determinar la pendiente m de la recta (ecuación 13) y el desplazamiento en el eje de las abscisas ecuación 16, el error determinado en la ecuación 19 y 22 para el desplazamiento y pendiente, se utiliza la tabulación de datos de la Tabla V-3: Tabulación de datos para el método de ajuste lineal.

$$m = \frac{N \sum_{i=1}^N x_i y_i - \sum_{i=1}^N x_i * \sum_{i=1}^N y_i}{N * \sum_{i=1}^N x_i^2 - \sum_{i=1}^N x_i * \sum_{i=1}^N x_i} \quad (13)$$

$$m = \frac{12 * 28193,96 - 354,1 * 955,5}{12 * 12639,71 - 354,1 * 354,1} \quad (14)$$

$$m = -0,00057 \quad (15)$$

$$b = \frac{\sum_{i=1}^N x_i^2 * \sum_{i=1}^N y_i - \sum_{i=1}^N x_i * \sum_{i=1}^N x_i y_i}{N * \sum_{i=1}^N x_i^2 - \sum_{i=1}^N x_i * \sum_{i=1}^N x_i} \quad (16)$$

$$b = \frac{12639,71 * 955,5 - 354,1 * 28193,96}{12 * 12639,71 - 354,1 * 354,1} \quad (17)$$

$$b = 79,6418 \quad (18)$$

$$\sigma_b = \sqrt{\frac{\sum_{i=1}^N x_i^2}{N \cdot \sum_{i=1}^N x_i^2 - \sum_{i=1}^N x_i \cdot \sum_{i=1}^N x_i} * \frac{\sum_{i=1}^N (b + m \cdot x_i - y_i)^2}{N-2}} \quad (19)$$

$$\sigma_b = \sqrt{\frac{12639,71}{12 \cdot 12639,71 - 354,1 \cdot 354,1} * \frac{0,31}{10}} \quad (20)$$

$$\sigma_b = 0,043 \quad (21)$$

$$\sigma_m = \sqrt{\frac{N}{N \cdot \sum_{i=1}^N x_i^2 - \sum_{i=1}^N x_i \cdot \sum_{i=1}^N x_i} * \frac{\sum_{i=1}^N (b + m \cdot x_i - y_i)^2}{N-2}} \quad (23)$$

$$\sigma_m = \sqrt{\frac{12}{12 \cdot 12639,71 - 354,1 \cdot 354,1} * \frac{0,31}{10}} \quad (24)$$

$$\sigma_m = 0,0013 \quad (25)$$

$$r = \frac{N \cdot \sum_{i=1}^N x_i y_i - \sum_{i=1}^N x_i \cdot \sum_{i=1}^N y_i}{\sqrt{N \cdot \sum_{i=1}^N x_i^2 - \sum_{i=1}^N x_i \cdot \sum_{i=1}^N x_i} \sqrt{N \cdot \sum_{i=1}^N y_i^2 - \sum_{i=1}^N y_i \cdot \sum_{i=1}^N y_i}} \quad (26)$$

$$r = \frac{12 \cdot 28193,96 - 354,1 \cdot 955,5}{\sqrt{12 \cdot 12639,71 - 354,1 \cdot 354,1} \sqrt{12 \cdot 76081,99 - 955,5 \cdot 955,5}} \quad (27)$$

$$r = 0,048 \quad (28)$$

$$y = (-0,00057 \pm 0,0013)x + (79,64 \pm 0,043) \quad (29)$$

De la ecuación 29 describe la fórmula general de la recta se realizó el gráfico descrito en la Figura V-4: Trayectoria Seguida por el Robot, en la cual se verifica la trayectoria seguida es una línea recta, se muestra la precisión del robot 95% en trayectorias lineales y la exactitud del 98%.

$$\text{precisión} = 100\% - ep \quad (30)$$

$$\text{precisión} = 95\% \quad (31)$$

$$\text{exactitud} = |79,6 - 79,62| \quad (32)$$

$$\text{exactitud} = 98\% \quad (33)$$

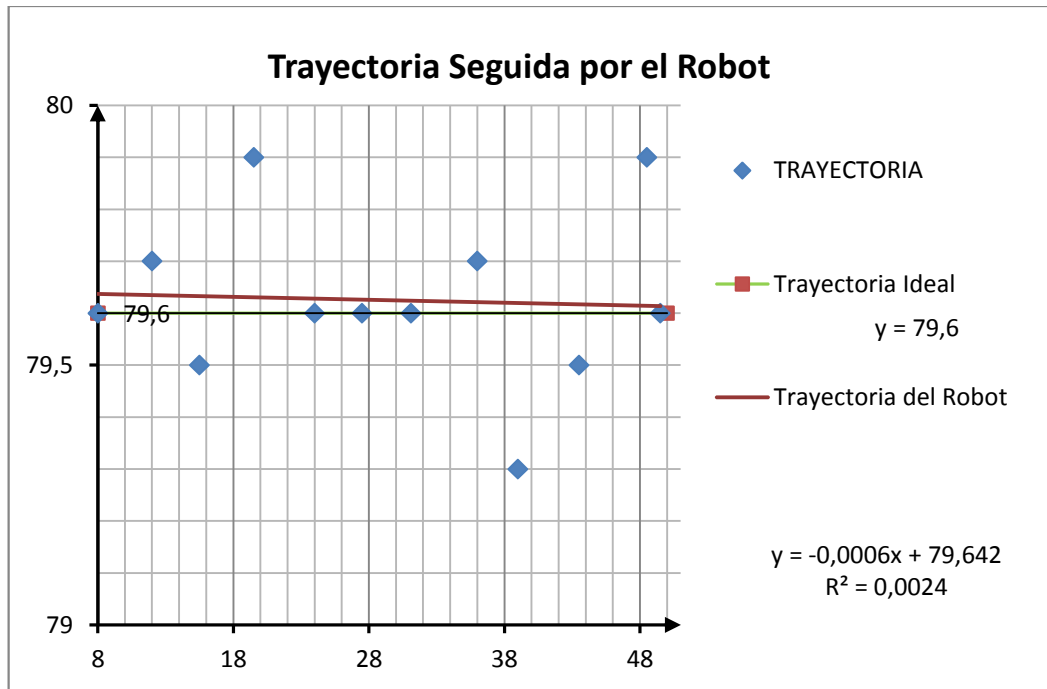


Figura V-4: Trayectoria Seguida por el Robot

#### 5.1.1.1.2 Segunda Etapa

El robot debe mover una caja en una trayectoria lineal como se muestra en la Figura V-5: Robot mueven una Caja desde el punto B al Punto C, para realizar esta etapa se subdividió en fases parciales las cuales son:

- ❖ Fase 1: Determinar la cantidad de robots necesarios para la tarea a realizar (mover caja)
- ❖ Fase 2: Verificar si la trayectoria es lineal cuando empuja el robot una caja

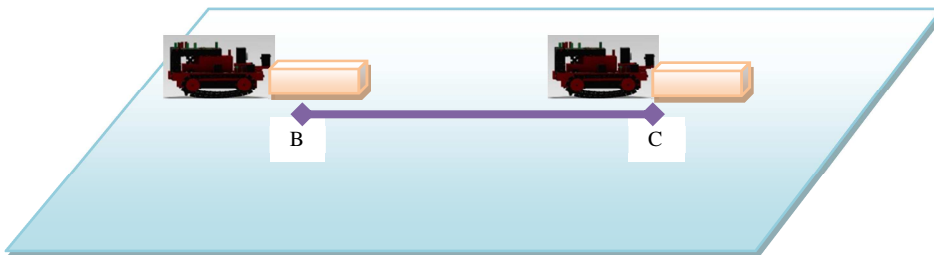


Figura V-5: Robot mover una Caja desde el punto B al punto C

**Fase 1:** Determinar la cantidad de robots necesarios para la tarea a realizar (mover caja)

#### 5.1.1.1.2.1 Fase 1: Determina el peso de la caja

Se realizó una prueba sencilla y experimental para determinar la cantidad de robots necesario para mover la caja, mediante el sensor de rotación de motor (Encoder), la prueba realizada fue activar los motores del robot por un lapso de 1 segundo y observar el valor obtenido de los sensor de rotación de motor, y tabular estos datos en la Tabla V 4: Determinación de las revoluciones para el peso. Sí las revoluciones son menores o iguales a 2 se considera a la caja es pesada por lo tanto un solo robot no puede ejecutar la acción, caso contrario el robot estará en la capacidad de cumplir su meta sin ningún tipo de ayuda, se evaluó la capacidad de los motores de realizar la acción de prueba con la misma caja pesada y se tabulo en la Tabla V-5: Determinación la exactitud del robot

N°	Numero de Vueltas	Peso
1	3	Liviana
2	1	Pesado
3	5	Liviano
4	2	Pesado
5	6	Liviano
6	12	Liviano
7	4	Liviano
8	0	Pesado
9	14	Liviano
10	2	Pesado

**Tabla V-4: Determinación de las revoluciones para el peso**

N	Numero de Vueltas	Peso	P
1	2	Pesado	1
2	1	Pesado	1
3	2	Pesado	1
4	3	Pesado	0
5	2	Pesado	1
6	1	Pesado	1
7	0	Pesado	1
8	0	Pesado	1
9	2	Pesado	1
10	3	Pesado	0
Numero de aciertos			8

**Tabla V-5: Determinación la exactitud del robot**

1 Acierto

0 No acertó

Se muestra en la Tabla V-5: Determinación la exactitud del robot, los datos obtenidos se puede determinar la exactitud del robot para determinar el peso de la misma. La prueba se realizó 10 veces, el robot falló 2 veces dando un grado de exactitud para determinar el peso de la caja de un 80%.

$$\mathbf{error} = n - \sum_{i=1}^n \mathbf{peso} \quad (33)$$

$$\mathbf{error} = 10 - 8 \quad (34)$$

$$\mathbf{error} = 20\% \quad (35)$$

$$\mathbf{exactitud} = 100\% - 20\% \quad (36)$$

$$\mathbf{exactitud} = 80\% \quad (37)$$

#### 5.1.1.1.2.2 Fase 2: Verificar Si la trayectoria es lineal cuando empuja el robot una caja

Una vez determinada el número de robots y la presión en cuanto a comunicación, se procede a verificar la prueba de la trayectoria lineal empuja una caja, se tabuló 8 veces, el robot llegó a su destino sin desviarse de su eje y cumple con su objetivo.

N	Distancia recorrida inicialmente sin caja (cm)	Tiempo	Distancia mover la caja	Desviación típica
1	20	4,8	50	0,1
2	20	5,2	50	0,3
3	20	5,3	50	0,7
4	20	∞	49	0,5
5	20	5,4	50	0,9
6	20	5,45	50	1
7	20	5,67	50	0,7
8	20	6,7	50	1,2
Pm	20	5,58	50	0,675

**Tabla V-6: Resultados de tiempo en la Solución Completa**

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}} \quad (38)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - 0,675)^2}{8}} \quad (39)$$

$$\sigma = 0,91 \quad (40)$$

$$e_p(\%) = \frac{\delta_x}{x} * 100 \quad (41)$$

$$e_p(\%) = 9\% \quad (42)$$

$$\text{precision} = 100\% - 9\% \quad (43)$$

$$\text{precision} = 91\% \quad (44)$$

$$\text{exactitud} = |50 - 50,675| \quad (45)$$

$$\text{exactitud} = 0,675 \quad (46)$$

$$\text{exactitud} = 67,5 \quad (47)$$



En la Tabla V-6: Resultados de tiempo en la Solución Completa, se puede evidenciar la falta de precisión de “TANQUE” en su tarea de mover una caja en línea recta al cabo de varias pruebas se tiene como resultado: en la primera prueba el robot tiene una precisión del 91%, y una exactitud para realizar la tarea de un 67,5%

#### 5.1.1.2 Prueba 2: Mover la caja con trayectoria circular

Mover el robot desde el punto A hasta el punto B en una trayectoria circular empujando una caja, se muestra en la Figura V-6: El Robot mueve una Caja en una trayectoria Circular, en esta prueba se verifica la precisión del robot al seguir una trayectoria circular.

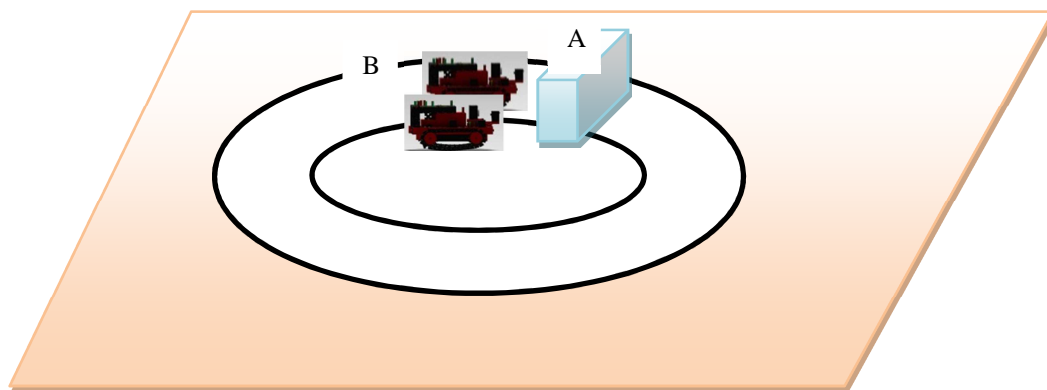


Figura V-6: El Robot mueve una Caja en una trayectoria Circular

Para determinar el funcionamiento del robot se dividió esta prueba en 2 etapas descritas a continuación:

**Etapas 1:** Movimientos de los robots en una trayectoria circular sin mover cajas

**Etapas 2:** Movimientos de los robots en una trayectoria circular moviendo cajas

##### 5.1.1.2.1 Etapas 1

Se realizó la trayectoria descrito en la Figura V-6: El Robot mueve una Caja en una trayectoria Circular, se tabuló los datos en la Tabla V-7: Resultados de la trayectoria realizada por el robot, donde se verificó el tipo de trayectoria realizada por el robot con la trayectoria ideal.

Eje X	Eje Y	Eje x Ideal	Eje Y Ideal	Desviación X	Desviación Y
50	94,5	50	94,5	0	0
38,3	66,1	37,5	67,3	0,8	1,2
46,5	37,8	44	37	2,5	0,8
61,1	22,5	60,5	21	0,6	1,5
82,7	17,9	81	16	1,7	1,9
99,5	21,3	100	20	0,5	1,3
112	33,6	113,5	31	1,5	2,6
124,8	43,9	123	43	1,8	0,9
128,5	67	128	64,5	0,5	2,5
111	98	113,5	97	2,5	1
83,8	111,6	81	109	2,8	2,6
53,5	97,5	50	94,5	3,5	3
82,64	59,31	81,83	57,9	1,56	1,61

**Tabla V-7: Resultados de la trayectoria realizada por el robot**

Para determinar la precisión del robot para seguir una trayectoria circular se utiliza el método de posicionamiento de medias, el mismo necesita transformar a coordenadas polares los datos tabulados en la Tabla V-7: Resultados de la trayectoria realizada por el robot, el método consiste en sumar los senos (ecuación 48) y los cosenos (ecuaciones 49) para determinar la ecuación general de la circunferencia (ecuación 38).

$$\bar{C} = \frac{1}{n} \sum_{i=1}^n \text{Cos} \theta_i \quad (48)$$

$$\bar{S} = \frac{1}{n} \sum_{i=1}^n \text{Sen}\theta_i \quad (49)$$

$$\bar{R} = \sqrt{\bar{C}^2 + \bar{S}^2} \quad (50)$$

$$\bar{R} = \sqrt{0,77^2 + 0,54^2} \quad (51)$$

$$\bar{R} = 0,94 \quad (52)$$

$$\bar{C} = \bar{R} \text{Cos}\bar{\theta} \quad (53)$$

$$\bar{\theta} = \text{cos}^{-1} \frac{\bar{C}}{\bar{S}} \quad (54)$$

$$\bar{\theta} = \text{cos}^{-1} \frac{0,77}{0,54} \quad (55)$$

$$\bar{\theta} = 45^\circ \quad (56)$$

La ecuación 57 describe la ecuación general de la circunferencia con centro en el origen, se determinó en base a los datos tabulados en la Tabla V 8: Datos determinación de la ecuación general de la circunferencia, con la cual se describe la trayectoria circular descrita por el robot.

$$R^2 = x^2 + y^2 \quad (57)$$

$$0,94^2 = x^2 + y^2 \quad (58)$$

$$0,88 = x^2 + y^2 \quad (59)$$

X	Y	X ideal	Y ideal	$\Theta$	R	Cos $\Theta$	Sen $\Theta$	R ideal	$\Theta$ Ideal
50	94,5	50	94,5	1.09	106,91	0.46	0.89	106,91	1,08
38,3	66,1	37,5	67,3	1.06	76,39	0.49	0.87	77,04	1,06
46,5	37,8	44	37	0.70	59,93	0.76	0.65	57,49	0,70
61,1	22,5	60,5	21	0.35	65,11	0.94	0.34	64,04	0,33
82,7	17,9	81	16	0.19	84,62	0.98	0.19	82,57	0,20
99,5	21,3	100	20	0.20	101,75	0.98	0.20	101,98	0,20
112	33,6	113,5	31	0.27	116,93	0.96	0.27	117,66	0,27
124,8	43,9	123	43	0.33	132,30	0.95	0.33	130,30	0,34
128,5	67	128	64,5	0.47	144,92	0.89	0.45	143,33	0,47
111	98	113,5	97	0.71	148,07	0.76	0.65	149,30	0,71
83,8	111,6	81	109	0.93	139,56	0.60	0.80	135,80	0,93
53,5	97,5	50	94,5	1.09	111,21	0.46	0.89	106,91	1,08
<b>Promedio</b>				0,61	106,5	0,77	0,54	107,31	0,61

**Tabla V-8: Datos determinación de la ecuación general de la circunferencia**

De la ecuación 59 se determinó la trayectoria circular descrita por “TANQUE”, en la Figura V-7: Trayectoria Circular realizada por el robot, se observa la trayectoria realiza por “TANQUE”, se determina a los robot precisos para seguir una trayectoria circular, su

exactitud es en tramos cortos de la trayectoria, debido al re direccionamiento del robot cuando se desvía de su trayectoria marcada.

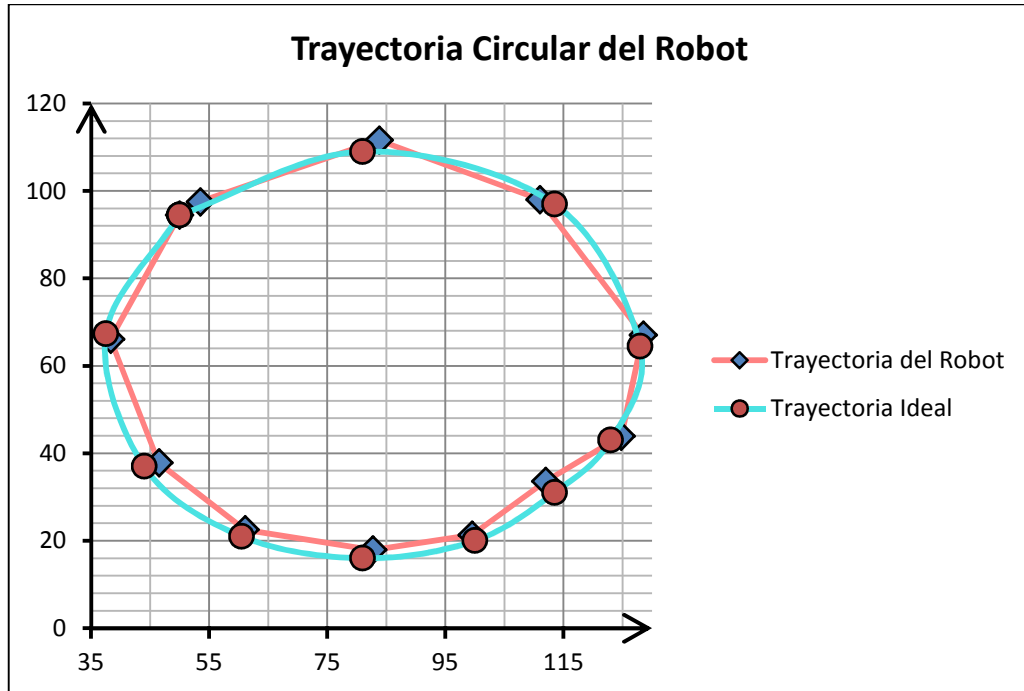


Figura V-7: Trayectoria Circular realizada por el robot

$$e_p(\%) = \frac{\delta_x}{\bar{x}} * 100 \quad (60)$$

$$e_p(\%) = 7\% \quad (61)$$

$$\text{precision} = 100\% - 7\% \quad (62)$$

$$\text{precision} = 93\% \quad (63)$$

$$\text{exactitud} = |107,31 - 106,5| \quad (64)$$

$$\text{exactitud} = 81\% \quad (65)$$

El robot al realizar una prueba en trayectoria circular sin mover ningún caja tiene una precisión del 93% y una exactitud del 81%.

### 5.1.1.2.2 Etapa 2

Se realizó esta prueba y solo se pudo demostrar que el robot empujaba la caja una distancia de 48 [cm] como se determina en la fórmula 64 en su trayectoria circular, se tabuló en la Tabla V-9: Resultados de la distancia recorrida en una trayectoria circular, el robot no cumple su objetivo, mover una caja en una trayectoria circular.

$$L = \frac{2 \times \pi \times r \times \theta}{360} \quad (66)$$

$$L = 48,17[cm] \quad (67)$$

X	Y	$\theta^\circ$	R	Posición angular
49.50	94.50	62,45°	106.68	62,45°
47.0	91.50	60,74°	101.68	60,74°
49.50	93.50	61,11°	104.68	61,11°
48.50	96.50	63,05°	110.68	63,05°

**Tabla V-9: Resultados de la distancia recorrida en una trayectoria circular**

### 5.1.1.3 Nivel de Cooperación alcanzado

En la Tabla V-10: Evaluación de Cooperación de los Agentes, se presenta la tabla de indicadores de desempeño en la cual se evaluó a los agente y se definió el porcentaje de cooperación entre los mismos.

Indicador	Descripción	Valor	Porcentaje
1. Disponibilidad para participar en la tarea.	El Agente disponible realiza adecuadamente la meta a cumplirse. Los agentes cumplen requerimientos físicos necesarios para realizar la acción.	3	60%
2. Gestión del tiempo	Sabe moverse en los parámetros del tiempo establecido. Gestiona bien el tiempo. Es adecuado el tiempo transcurrido en la colaboración de los agentes.	4	80%
3. Adaptación al grupo de trabajo	El agente determina adecuadamente su posicionamiento sin causar inconvenientes a otro agentes Agente cumple con los requerimientos establecidos en cuanto a dimensiones.	3	60%
4. Comunicación entre robots.	Se establece la comunicación entre agentes. El agente emite órdenes claras para el resto del grupo. El tiempo de respuesta entre agentes es corto.	5	100%
5. Cumple satisfactoriamente la tarea asignada	Tienen precisión y exactitud en las tareas asignadas. El grupo de agentes ejecutan la tarea satisfactoriamente. El grupo de agentes recorre distancia requerida en la tarea asignada.	4	80%
<b>Total</b>		19	380%
<b>Promedio sobre 5</b>		3,8	76%

**Tabla V-10: Evaluación de Cooperación de los Agentes**

En el presente proyecto el nivel de cooperación alcanzado entre los agentes es del 76%, este valor se determinó en la Tabla V-10: Evaluación de Cooperación de los Agentes, al igual se evaluó el desempeño de los agentes en el sistema y su disponibilidad en el mismo. La escala de calificación es de 1-5 donde 1 es la peor situación y 5 es la mejor.

#### 5.1.1.4 Sistema Multi – Agentes

Se realizó una evaluación del sistema multi –Agentes en base a la siguiente Tabla V 11:

Evaluación del Sistema Multi - agentes

Indicador	Descripción	Valor	Porcentaje
1. Seguir una Trayectoria	La exactitud para seguir una trayectoria lineal o circular. La precisión para seguir una trayectoria lineal o circular.	3	60%
2. Determinación del Número de Robots	Exactitud para determinar el peso de la caja. Precisión para determinar el número de robots. Precisión para pedir la ayuda necesaria.	4	80%
3. Mover la Caja	La precisión para colocarse en los puntos de fuerza definidos. La precisión para mover la caja la distancia prevista. La exactitud para mover la caja en las trayectorias definidas inicialmente	4	80%
<b>Total</b>		11	220%
<b>Promedio sobre 5</b>		3,66	73,4%

Tabla V-11: Evaluación del Sistema Multi - agentes

En el presente proyecto el sistema multi –agentes, alcanza a cumplir sus metas en un 73,4%, en la Tabla V 11: Evaluación del Sistema Multi - agentes se observa que la meta de mover cajas y la determinación del número de robots es de un 80%. La escala de calificación es de 1-5 donde 1 es la peor situación y 5 es la mejor.

#### 5.1.2 Análisis de resultados finales

En el presente proyecto se alcanzó un nivel de cooperación entre los agentes de un 76%, se demuestra su capacidad de interactuar con otros agentes cumpliendo un objetivo en común.

En las pruebas realizadas en las diferentes trayectorias se obtuvo un 91% de precisión de los robots para mover la caja en línea recta, pero al realizar la tarea repetitivamente se encontraron fallas en la robustez de los motores, se demostró la capacidad de los agentes para cumplir una meta, en una trayectoria circular no se pudo



determinar la precisión de los agentes por presentarse inconvenientes mecánicos producidos por la falta de robustez de los motores, para realizar las tareas mencionadas en el proyecto, estos valores se los encuentra en la Tabla V 12: Resultados Finales Obtenidos.

Indicadores	Resultados			
Nivel de Cooperación	76%			
Nivel de Cumplimiento de sus metas	73,4%			
Trayectoria realizadas	Lineal	Realizó la acción una distancia de: 50[cm]	Precisión	91%
	Circular	Realizó la acción un ángulo de : 45°	Precisión	No se determinó no concluyo la prueba

Tabla V-12: Resultados Finales Obtenidos

### 5.1.2.1 Problemas encontrados en el proyecto

En el presente proyecto se suscitaron los siguientes inconvenientes descritos en la

Tabla V-13: Problemas del Proyecto, a los cuales se buscaron las soluciones adecuadas para cada uno.

Problemas	Descripción	Solución
<b>Limitación del Angulo de Visión del Sensor Ultrasónico</b>	Debido a su ubicación el sensor ultrasónico detecto objetos en un ángulo de visión de 45°, su ubicación muy cercana al piso detecta a 3[cm] objetos.	Ubicar el sensor a una distancia mayor (9[cm])
<b>Exactitud de los motores para seguir una línea</b>	Los robot luego de varias pruebas pierden fuerza y no siguen una línea recta	Se definió una trayectoria pre establecida mediante una línea negra
<b>Motores uso prolongado</b>	Genero pérdida de potencia del mismo, debido al recalentamiento. <ul style="list-style-type: none"> <li>• Caídas de Tensión</li> <li>• Paros Indebidos</li> <li>• Perdidas de Energía</li> <li>• Sobre corrientes</li> </ul>	<ul style="list-style-type: none"> <li>• Cambiar el diseño para alivianar el peso</li> <li>• Comprar nuevos motores de similares características.</li> <li>• Análisis mecánico de la estructura</li> </ul>
<b>Baterías</b>	Su uso es restringido en tiempo, por la cantidad de sensores y actuadores las baterías se descargan rápido	Utilizar un cargado de 9[v] conectado a la corriente

Tabla V-13: Problemas del Proyecto

### 5.1.2.2 Análisis Mecánico de “TANQUE”

Por las contrariedades descritos en la

Tabla V-13: Problemas del Proyecto, se analiza Sí “TANQUE” puede mover su estructura y si es capaz de empujar un peso adicional analizando las características físicas de los motores de “TANQUE” descritas en la Tabla V-14: Características Técnicas del Motor “TANQUE”

Característica	Valor
Torque	0,06[Nm]
Velocidad	173,5[rpm]
Corriente	0,465 <sup>a</sup>
Voltaje	9[V]
Potencia	4,18[VA]

**Tabla V-14: Características Técnicas del Motor “TANQUE”**

Para determinar la fuerza necesaria para mover el robot “TANQUE” necesita de los siguientes datos descritos Tabla V 15: Datos de “TANQUE”.

Datos	Valor
Peso total del robot	0,602[Kg]
Radio de la ruedas motrices	0,022[m]
Numero de ruedas	2

**Tabla V-15: Datos de “TANQUE”**

Para determinar la fuerza necesaria para mover a “TANQUE”, es necesario determinar la fuerza de las ruedas, para lo cual se utilizó la ecuación 68, el radio de la rueda influye directamente proporcional con el Torque.

$$\text{Torque } (\tau) = \text{Fuerza } (F) \times \text{Radio Rueda } (r) \quad (68)$$

$$F = \frac{\tau}{r} \quad (69)$$

$$F = \frac{0,06[N*m]}{0,022[m]} \quad (70)$$

$$F = 3[N] \quad (71)$$

La velocidad lineal de “TANQUE”, se la puede calcular a partir de la fórmula 72.

$$\text{Velocidad (V)} = \text{Velocidad Angular}(\omega) \times 2 \times \pi \times \text{Radio Rueda (r)} \quad (72)$$

$$V = \omega \times 2 \times \pi \times r \quad (73)$$

$$V = 18,16 \left[ \frac{\text{rad}}{\text{s}} \right] \times 2 \times \pi \times 0,02[\text{m}] \quad (74)$$

$$V = 2,51 \left[ \frac{\text{m}}{\text{s}} \right] \quad (75)$$

La cantidad de movimiento da una idea de la agilidad de “TANQUE”, permite la facilidad de cambios de dirección y velocidad en “TANQUE” se la calculará con la fórmula 76.

$$\text{Cantidad de Movimiento} = \text{Masa(M)} \times \text{Velocidad(v)} \quad (76)$$

$$\text{Cantidad de Movimiento} = M \times v \quad (77)$$

$$\text{Cantidad de Movimiento} = 0,602[\text{kg}] \times 2,51 \left[ \frac{\text{m}}{\text{s}} \right] \quad (78)$$

$$\text{Cantidad de Movimiento} = 1,50[\text{N}] \quad (79)$$

La aceleración se la obtiene en base a la fórmula 80, la aceleración es directamente proporcional al número de motores e inversamente proporcional al radio.

$$\text{Fuerza(F)} = \text{Masa (m)} \times \text{Aceleración(a)} \quad (80)$$

$$a = \frac{F}{m} \quad (81)$$

$$a = \frac{3[N]}{0,602[\text{kg}]} \quad (82)$$

$$a = 4,9 \left[ \frac{\text{m}^2}{\text{s}} \right] \quad (83)$$

Condición de movimiento para “TANQUE”, debe cumplir a la potencia entregada por el motor sea mayor a la potencia mecánica del Robot, se describe en la fórmula 84.

$$Pm > Pr \quad (84)$$

Pm = Potencia entrega del motor

Pr = Potencia mecánica

Donde la potencia de un motor se define por la fórmula 85

$$\mathbf{Pm} = \mathbf{Par}(\boldsymbol{\tau}) \times \mathbf{Velocidad}(\boldsymbol{\omega}) \quad (85)$$

$$\mathbf{Pm} = \boldsymbol{\tau} \times \boldsymbol{\omega} \quad (86)$$

$$\mathbf{Pm} = \mathbf{0,06[N * m]} \times \mathbf{173,5[rpm]} \quad (87)$$

$$\mathbf{Pm} = \mathbf{0,06[N * m]} \times \mathbf{18,16 \left[ \frac{rad}{s} \right]} \quad (88)$$

$$\mathbf{Pm} = \mathbf{1,09 [W]} \quad (89)$$

La potencia mecánica de un robot se define por la fórmula 90

$$\mathbf{Pr} = \frac{\mathbf{Masa}(M) \times \mathbf{Aceleración}(a) \times \mathbf{Velocidad}(V)}{2 \times \pi} \quad (90)$$

$$\mathbf{Pr} = \frac{M \times a \times V}{2 \times \pi} \quad (91)$$

$$\mathbf{Pr} = \frac{\mathbf{0,602[kg]} \times \mathbf{4,8 \left[ \frac{m^2}{s} \right]} \times \mathbf{2,51 \left[ \frac{m}{s} \right]}}{2 \times \pi} \quad (92)$$

$$\mathbf{Pr} = \mathbf{1,08[W]} \quad (93)$$

En la Ecuación 84 se define la potencia del motor debe ser mayor a la potencia mecánica, para el caso específico de “TANQUE”, se concluye los motores de “TANQUE” avanza su peso.

De las ecuación 89 y 93 se determina que “TANQUE” avanza su propio peso, solo está diseñado para cargar cargas livianas, por tener características técnicas limitadas. Sí se sobrecargar su peso se limita su funcionamiento.

## CAPÍTULO VI .

### RESULTADOS DEL PROYECTO

En el presente Capítulo se mencionan los conceptos y premisas de robótica cooperativa resultantes de la investigación realizada a lo largo del proyecto de grado.

#### 6.1 RESULTADOS DEL PROYECTO

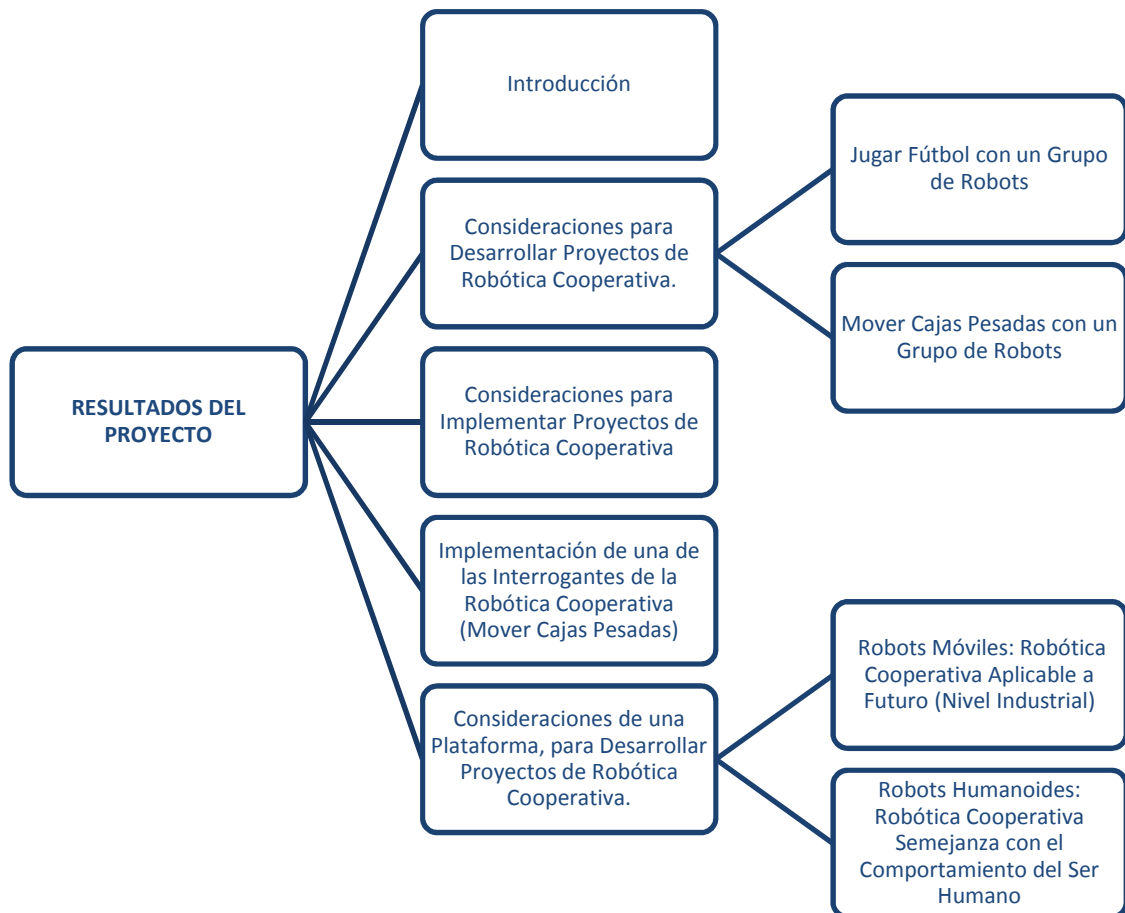


Figura VI-1: Estructura del Capítulo VI

### 6.1.1 Introducción

La robótica cooperativa es un grupo de robots interactuando entre sí, para solucionar un problema por distintos medios y protocolos de comunicación, intercambian información entre ellos, la solución del problema nace del comportamiento del grupo, así en el presente capítulo se pretende demostrar este concepto.

### 6.1.2 Consideraciones para desarrollar proyectos de robótica cooperativa.

En el Anexo 4 tabla 1, se describen las investigaciones realizadas a nivel mundial sobre robótica cooperativa, en el **¡Error! No se encuentra el origen de la referencia.**, se observa a América Latina como el continente pionero en las investigaciones, sin quedarse atrás Europa, con el fin de aumentar producción, reducir costos y el mejor los tiempos de producción, uniendo habilidades de robots simples o aprovechando los recursos robóticos encontrados en casa o en la industria, así se lograría llegar a solucionar varios problemas existentes en el confort del diario vivir del ser humano como la limpieza (patio de las casas).y en la industria, por ejemplo mover objetos pesados.

Las interrogantes planteadas para la demostración de la robótica cooperativa nacé de problemas encontrados en los seres humanos donde es necesario la cooperación las cuales son:

¿Cómo los seres humanos mueven una caja pesada?

¿Cómo coordinan los seres humanos el movimiento de una pelota para jugar fútbol?,

En la mayoría de los casos se resuelven estas preguntas cuando un ser humano pide ayuda a otra persona y coordinan habilidades. Estas mismas interrogantes pueden ser planteadas por un grupo de robot así se puede decir ¿Cómo un grupo de robot pueden mover una caja pesada? O ¿Cómo un grupo de robot pueden coordinarse para jugar fútbol?, para responder estas pregunta se debe realizar un análisis sobre es el nivel de inteligencia y cooperación dada al conjunto de agentes.

La primera interrogante planteada para la demostración de la robótica cooperativa se la realiza con plataformas robóticas móviles, de esta manera se abren nuevas inquietudes como ¿Cuántos robots serían necesarios para mover una caja?, ¿Cuál es la trayectoria a seguir por los robots cuando mueven una caja?, ¿La distancia máxima lograda por el conjunto de robots al empujar el objeto?

La segunda interrogante planteada para demostrar robótica cooperativa origina controversias en cuanto a cuál es el agente en realizar el primer movimiento con la pelota, ¿el más cercano a la pelota o es un agente libre?, para lo cual por su facilidad de maniobrabilidad sería más acertado utilizar plataformas humanoides.

La finalidad de este trabajo es presentar los conceptos y premisas de robótica cooperativa, en base a la tarea o la aplicación a realizar, la coordinación de los agentes, el nivel de cooperación de un grupo de robots es independiente del robot a utilizar como móviles o humanoides, pero si es dependiente del nivel de comunicación y coordinación del sistema, en la presente investigación en el Capítulo 5, tabla V-12, se tiene si el nivel de cooperación es del 76% entre agentes robóticos las tareas asignadas serán cumplidas adecuadamente.

#### **6.1.2.1 Mover cajas pesadas con un grupo de robots**

Para este tema será necesario contestar a las 3 interrogantes mencionadas, así a la primera interrogante se la pudo resolver en base a la creación de un algoritmo general de la robótica cooperativa el cual se observa en el Capítulo 4 en la Figura IV-5: Diagrama de Flujo General del Problema, para saber la cantidad de robot necesarios para mover una caja, el algoritmo hace una prueba básica donde se determina el peso de la caja. Así se establecería el número de robots necesarios para mover la caja, esto se demostró en el Capítulo 5 en base a la determinación de peso de la caja, en la Tabla V 5: Determinación la exactitud del robot, se verifico la exactitud del proceso para determinar la cantidad de robots necesarios para mover una caja dando como resultado un 80% en exactitud.

La segunda interrogante fue resuelta en base al algoritmo general del Capítulo 4 de la Figura IV-5: Diagrama de Flujo General del Problema. los puntos de fuerza de los robot con la caja generan la trayectoria por la cual se va a mover la caja, en el presente proyecto el algoritmo general determina los puntos de fuerza los cuales deben estar en el centro de masas, así la caja se divide en el número de robots necesarios para empujarla encontrando en cada división el centro de masa, de esta forma se asegura el movimiento de la caja en una trayectoria lineal, lo cual se pudo constatar con los resultados obtenidos en el Capítulo 5 en la Figura V-4: Trayectoria Seguida por el Robot.

La tercera interrogante se la pudo verificar en el Capítulo 5 en la Tabla V-7: Resultados de la trayectoria realizada por el robot, donde se determinó, los robots pueden

mover la caja una distancia de 50 [cm] con trayectoria lineal, esta distancia es limitada debido a problemas descritos en el Capítulo 5

Tabla V-13: Problemas del Proyecto.

### 6.1.2.2 Jugar fútbol con un grupo de robots

Se dejó abierta una pregunta en este apartado, la cual es ¿la coordinación para poder jugar fútbol depende exclusivamente de la comunicación de los agentes?, pero esta interrogante no se pudo desarrollar por la falta de una plataforma humanoide para realizar pruebas, sin embargo se demostró en el Capítulo 5

### 6.1.3 Consideraciones para implementar proyectos de robótica cooperativa

En América Latina el país pionero en realizar Investigaciones en el campo de robótica cooperativa es Colombia como se observa en la Tabla V 12: Resultados Finales Obtenidos, para la coordinación entre agentes la base es la comunicación es decir no importa la tarea asignada siempre y cuando la comunicación sea la adecuada la tarea es realizable. **¡Error! No se encuentra el origen de la referencia.**, en Ecuador para seguir a la vanguardia de las investigaciones la ESPE propone y desarrolla un proyecto de investigación científica en área de robótica cooperativa con ayuda de estudiantes de ingeniería electrónica y mecatrónica.

En la demostración de robótica cooperativa no es necesario determinar una plataforma robótica, pero se debe tener al menos dos robot con la capacidad de comunicarse entre sí, se debe considerar las aplicación a realizar así se puede desarrollar proyectos de robótica cooperativa con plataformas móviles y humanoides, en el **¡Error! No se encuentra el origen de la referencia.**, se observa, el 81% de las investigaciones es realizada con robot móviles y solo el 13% con plataformas humanoides.

Así de las interrogantes plantadas anteriormente si se pretende demostrar robótica cooperativa con un grupo de robots jugando fútbol se recomienda una plataforma humanoide a continuación se describe una plataforma en la Tabla VI-1: Robot Ideal.

Características	Requerimientos	Descripción
Procesador	Velocidad 100 MHz Memoria 8 Mbyte	Permitirá autonomía este tipo de controlador.



		Pero en la plataformas humanoides es costoso este tipo de procesadores, se recomienda el uso de una computadora
Sensores	Giroscopio	Permite la estabilidad del robot se sabe si el robot se cayó.
Comunicación	ZigBee	Permite interactuar con otros robots
Actuadores	16 Servos	Permiten movilidad, agilidad,
Cámara	Alta definición Resolución 1280 x 890 30 [fps]	Permite determinar un posicionamiento en el área de trabajo

**Tabla VI-1: Robot Humanoide Ideal**

En el **¡Error! No se encuentra el origen de la referencia.** Humanoides, se especifica más los tipos de robot existentes en el mercado y accesibles al presupuesto de la ESPE, las características para la selección del robot se las encuentra en el Tabla del Anexo 5 1: Características Principales para Adquisición del Robot se las puede observar, los pesos se encuentra especificados en el Tabla del Anexo 5 2: Pesos de los Indicadores en el análisis de estas tablas se determina en el Tabla del Anexo 5 3: Ponderación de los Indicadores, la mejor opción para este proyecto es el humanoide tipo ROBONOVA marca Hitec, sin embargo con los inconvenientes encontrados en el Anexo 7 página 1 la adquisición de los equipos no se logró concretar. Con el fin de demostrar los conceptos descritos en el Capítulo 2, se realizó la aplicación de mover cajas, para esta aplicación es más factible realizarla con robots móviles, se describe en la Tabla VI 2: Robot Móvil Ideal.

Se procedió a realizar pruebas con los equipos disponibles en la institución para realizar el proyecto de grado, se puede observar en la Tabla VI 3: Pruebas Realizadas con el Robot IRobot Create, las cuales dieron como conclusión la necesidad de una computadora para realizar el proyecto minimizando la autonomía, por esa razón se procede a realizar un análisis en el , los cuales existían y se encontraban disponibles en la ESPE, en la Tabla del Anexo 8 1: Características Principales para Selección de Robot, se determina las características más relevantes para la selección robot, los pesos de dichas características en la Tabla del Anexo 8 3: Ponderación de los Indicadores, se determina a Fischertechnik

(“TANQUE”) como la opción adecuada para continuar con el proyecto debido a las particulares técnicas del procesador descrito en la Tabla del Anexo 8 2: Pesos de los Indicadores, , además esta marca tiene su representante legal en el país, y ofrece garantía y asistencia técnica de los equipos adquiridos, estos robot permitirían autonomía de una computadora, al ser su procesador rápido, una memoria RAM suficiente para guardar un programa ejecutable y un algoritmo de búsqueda y seguimiento de trayectoria, además de estar implementados con comunicación interna con otros controladores a través del estándar de comunicación inalámbricas Bluetooth lo cual permiten interacción entre robots, cuenta con una amplia gama de sensores permitiendo tener una interacción con el medio que lo rodea.

<b>Prueba Realizada</b>	<b>Resultado de la Prueba</b>	<b>Conclusión</b>
Comunicación computador robot	La comunicación es sencilla y no requiere de un firmware exclusivo, solo se necesita las librerías y el software seleccionado en este caso Visual C.	El robot es fácilmente manipulable a través de un computador.
Comunicación con otro robot	A través de la tarjeta ZigBee se comunica inalámbricamente el robot con la computadora pero no se comunica con otro robot	Tarjeta ZigBee solo permite eliminar el cable de conexión Rs 232 (pero no se logra trabajar sin la necesidad de una computadora)
Capacidad de Almacenamiento de Información	La memoria de IRobot Create es 1288 [Kbyte] y depende de una computadora para realizar cualquier tarea.	Se necesita una computadora o un controlador con mayor capacidad.

**Tabla VI-2: Pruebas Realizadas con el Robot IRobot Create**

#### **6.1.4 Implementación de una de las interrogantes de la robótica cooperativa (mover cajas pesadas)**

La implementación para demostrar el concepto de robótica cooperativa en base a una de sus interrogantes se realizó con la plataforma “TANQUE”, la cual esta descrita en el Capítulo 3, la misma permitió ampliar los conceptos descritos en el Capítulo 2.

- El concepto de robótica cooperativa planteada en el Capítulo 2, determina la interacción de un robot en un grupo, en el presente proyecto, contó con dos

elementos del mismo tipo llamados “TANQUE”, para la demostración de este concepto se plantea un algoritmo en el Capítulo 4 Figura IV-5: Diagrama de Flujo General del Problema, para mover una caja en una trayectoria lineal, con el cual se observa la interacción de los agentes en su área de trabajo, se sincronizan para moverse los dos una misma distancia, esto se verifica en el Capítulo 5, los robots mueven la misma distancia y en la misma trayectoria, comprobando el concepto de robótica cooperativa .

- Se verificó el concepto de inteligencia cooperativa descrito en el Capítulo 2, es decir si un solo agente no puede cumplir con su tarea, pide ayuda al grupo de agentes, en el proyecto se demostró mediante la prueba de mover cajas pesadas, en el Capítulo 5 en la prueba de determinación de peso, el agente verifica si necesita la colaboración de otro robot para mover la caja o sí puede hacer solo la tarea mencionada, mediante un sistema de comunicación Bluetooth.
- Se desarrolló un algoritmo general de la robótica cooperativa para mover cajas en el Capítulo 4 en la Figura IV-5: Diagrama de Flujo General del Problema, el cual permite determinar el número de robots a utilizar de acuerdo al peso de los objetos a mover, con el fin de mejorar recursos, así se tiene la posibilidad de utilizar los otros agentes en otra tarea.
- Se desarrolló un algoritmo en el Capítulo 4 en la Figura IV-4: Sistema Multi Robots, el cual de acuerdo a la cantidad de robots necesarios para mover una caja pesada, determina el posicionamiento de cada uno de los agentes en el centro de masa determinado por el algoritmo así se logró equilibrar las fuerzas de los agentes para mover cajas en línea recta, esto se demostró en el Capítulo 5 en la prueba de determinación peso en la tabla V-5.
- Uno de los principales puntos para desarrollar un sistema de agentes cooperativos, es la capacidad de los mismos para comunicarse con otros, en el proyecto se logró sincronizar a dos elementos robóticos, los cuales cuenta con un sistema de comunicación Bluetooth este permitió establecer una comunicación para sincronizar a los agentes en el cumplimiento de su meta, esto se verifica en el Capítulo 5, tabla V-12 donde se puede observar la necesidad de la comunicación para sincronizar y realizar su objetivo.

- Se determinó un algoritmo general en el Capítulo 4 Figura IV-4: Sistema Multi Robots, el cual permite demostrar robótica cooperativa mediante la prueba de mover una caja de un peso superior al de un agente, en una trayectoria lineal o circular, el mismo puede ser implementado con cualquier plataforma móvil.
- Una de las metas planteadas en el proyecto fue la autonomía de los robots a una computadora, para lograr esta meta, se diseñó una arquitectura distribuida la cual necesita tener controladores con características de procesamiento y memoria altas descritas, y deben comunicarse entre si esto se observa en el Capítulo 3 Figura III-5: Arquitectura Implementada para el Desarrollo del Proyecto, “TANQUE” cumple con este tipo de controladores, permitiendo crear algoritmos para la simulación de inteligencia en los robots, los cuales le ayudan a decidir, como en el caso de si necesitan ayuda o no.
- Con el fin de demostrar robótica cooperativa se encontró necesario el acondicionamiento de la plataforma de “TANQUE”, a nivel de sensores, actuadores y comunicaciones con otros agentes, en el Capítulo 3 página 37-41, se tiene una descripción detallada de cada uno de estos elementos y su funcionamiento, para desarrollar la aplicación de mover una caja con dos agentes en trayectorias lineales.
- Al realizar las pruebas de mover la caja circularmente descritas en el Capítulo 5, los equipos manifestaron falta de robustez, los motores se sobrecalientan ocasionando paros indebidos, por esta razón no se logró la aplicación de jugar fútbol.
- El nivel de cooperación alcanzado por el sistema se lo realizó en la prueba de trayectoria lineal, este nivel del 76% de la tabla presentada en el Capítulo 5, permite demostrar: dos agentes robóticos de características homogéneas pueden realizar tareas en común sin dificultad, siempre que su comunicación y robustez sea la adecuada.
- Mientras se hicieron las distintas pruebas con “TANQUE” en el Capítulo 5 para moverse en trayectorias lineales se determinó: solo puede moverse 50 [cm], con una línea negra colocada en el piso, este fue la ayuda colocada en el sistema por los diferentes inconvenientes presentados a lo largo del desarrollo del mismo, este limitante en las trayectorias deben ser predefinidas en el área trabajo, por la dificultad para mover una pelota desde un punto A un punto B, donde el punto A

viene a ser un punto cualquiera en la cancha, mientras el punto B representa el arco del contrincante.

- Mover una caja con una trayectoria circular: para esto fue importante la sincronización de las velocidades angulares de los dos equipos, logrando mover un ángulo de  $45^\circ$  en el Capítulo 5, Figura V-7: Trayectoria Circular realizada por el robot, se observa los cambios q posee los robots al moverse en forma circular, está meta no se logró culminar al presentarse problemas de robustez de los motores.

#### 6.1.4.1. Limitantes del proyecto

En el desarrollo del proyecto se encontraron limitantes en hardware, los motores de la plataforma TANQUE, al no estar diseñados para mover grandes pesos, y al no trabajar por largos periodos de tiempo ocasionaron paros indebidos, estos inconvenientes se lograron solventar trabajando en periodos cortos de tiempo y disminuyendo el peso de la caja, logrando sentar las bases y determinar las premisas de robótica cooperativa.

#### 6.1.5 Consideraciones de una plataforma, para desarrollar proyectos de robótica cooperativa.

##### 6.1.5.1 Robots Móviles: Robótica Cooperativa aplicable a futuro (nivel industrial)

La robótica cooperativa se la puede utilizar tareas repetitivas como mover objetos y ubicarlos en determinados sitios, este tipo de aplicaciones se las podría utilizar en la industria una vez realizadas las pruebas necesarias para evitar retrasos o pérdidas en la industria, se propone utilizar robots móviles con las características descritas en la Tabla VI 2: Robot Móvil Ideal. Con las características descritas en la tabla VI-2, se encontró un robot comercial de nombre Pioneer el cual se describe y sus características técnicas se enuncian en la Tabla VI-4 características técnicas del Pioneer.

Característica	Descripción	
Microprocesador	Velocidad	44.2368 MHz
	Arquitectura	RISC
	Memoria RAM	32 Kbyte
	Memoria Flash	128 Kbyte
Puertos	RS 232	Velocidad 9,6 a 115.2 [Kbaud]
Sensores	Ultrasónico	4 arreglos hasta 8 sensores ultrasónicos ( $20^\circ$ )
	Bumpers	2 conectores

	Encoder (Rev.)	Vueltas	76.60 Rev.
			129[mm]
			33,50°
Actuadores	Zumbador		
	Motores	Sistema de Transmisión	
		PWM	
		Dirección	
		Velocidad de Translación	1400[mm/ seg]
		Velocidad de Rotación	300[°/seg]
		Radio de Engrane	38 3:1
Diámetro Rueda		195.3 [mm]	
Entradas	Analógicas	Potencia 5 y 12 VCC	
	Digitales	8 bits	
Puertos	Joystick	2 ejes 2 botones	
Interface	I2C	Con 4 Líneas	
Auxiliares	2 Interruptores de potencia 5 y 12 VCC con indicadores Led		
Reset	Pulsador externo para Reset de motores		
Computador a bordo	Procesador	dual-core 2.26 GHz 32-bit	
	Puertos	Ethernet: dual 10/100/1000	
		USB 2.0	
		Serial RS-232	
		Bluetooth	
	Memoria RAM	2 GB	
	Tarjeta de Video	512 MB con VGA	
	Tarjeta de Audio	HD audio	
	Disco duro	Sata 40 GB solido	
Sistema	Windows Xp 32 bits o Linux		

**Tabla VI-3: Características Técnicas del Robot Pioneer**

Característica	Dimensión
Longitud	44,5[cm]
Ancho	39,3[cm]
Altura	23,7[cm]
Distancia robot con bumpers	3,5[cm]

Peso	9[Kg]
Carga	25[kg]

**Tabla VI-4: Características Físicas del Robot Pioneer**

#### **6.1.5.2 Robots Humanoides: Robótica Cooperativa semejanza con el comportamiento del ser humano**

La robótica cooperativa utiliza robots humanoides teniendo como objetivo simular el comportamiento humano, por ejemplo varios robots jugando fútbol o haciendo coreografías, etc. En la Tabla VI-1: Robot Humanoide Ideal, se describen las características técnicas del robot humanoide ideal.

En el Tabla del Anexo 5 3: Ponderación de los Indicadores, se realiza un análisis de 3 robots humanoides, dando como resultado el segundo robot adecuado para realizar aplicaciones de robótica cooperativa es el BIOLOID, en el **¡Error! No se encuentra el origen de la referencia.**, se describen a los Robots.

## CAPÍTULO VII .

### CONCLUSIONES

En este capítulo se presentan las conclusiones y recomendaciones obtenidas a lo largo del desarrollo del proyecto de grado.

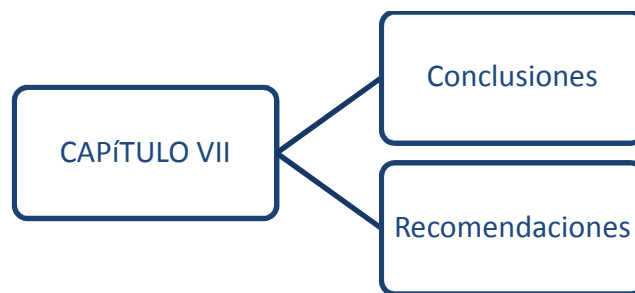


Figura VII-1: Estructura del Capítulo VII

#### 7.1 CONCLUSIONES

- De los resultados de las investigaciones realizadas sobre robótica cooperativa y de la experiencia obtenida se concluye que la robótica cooperativa es un conjunto de robots homogéneos con determinadas habilidades, las mismas son aprovechadas por el grupo para realizar una tarea específica, los robots TANQUE tienen la habilidad de empujar cajas, al utilizar la robótica cooperativa se puede amplificar esta habilidad y mover cajas pesadas.
- De los resultados obtenidos de las investigaciones realizadas en el Anexo 4 en la tabla 1 se concluye que los robots más usados para desarrollar proyectos de robótica cooperativa son los móviles en un 81%, como LEGO, prototipos creados por los diferentes investigadores en base a PIONEER, DAGU, DFRobot, en su mayoría en aplicaciones de mover objetos, y en un 13% los robots humanoides ROBONOVA, BIOLOID, KONDO estos son utilizados en campeonatos como por ejemplo Robo CUP, y el 6% son simulaciones robóticas.



- Se determinó que la robótica cooperativa depende del nivel de comunicación y coordinación de los agentes, por lo tanto se define al tipo de robot a utilizar de acuerdo a la tarea a realizar, por lo general si va empujar un objeto serán móviles.
- El nivel de cooperación de un grupo de agentes alcanzado en este proyecto es del 76% en la tabla V-12 Capítulo 5, demostrando a un grupo de robots trabajando para cumplir una meta en común, se logró el acondicionamiento de la plataforma TANQUE para cumplir su meta, obteniendo la colaboración entre agentes.
- El sistema desarrollado permite a dos robots interactuar entre sí en un área de trabajo controlada, cumpliendo así el objeto de mover una caja en forma lineal una distancia de 50[cm], estos datos fueron obtenidos de las pruebas realizadas previamente en el Capítulo V, el resultado tan limitado en cuanto a la máxima distancia que los dos robots mueven la caja es debido a las limitaciones físicas de los motores enunciadas en la tabla V-14 del Capítulo V.
- Se determinó que el algoritmo implementado para el problema de mover una caja entre varios robots tiene un 80% de exactitud en la determinación del peso de la caja, este dato es importante en el algoritmo ya que es el encargado de determinar el número de agentes para realizar la tarea asignada.
- Se concluye que el tiempo de ejecución aumenta significativamente debido al uso prolongado de los equipos, causando el sobreesfuerzo de los motores por efecto de sobrecalentamiento de los mismos, generando inconvenientes como paros indebidos del robot.
- Debido a las limitaciones en memoria no se logró que el sistema robótico cooperativo alcance el nivel de inteligencia deseado, es decir que no requiera de un seguidor de línea para realizar las diferentes tareas asignadas.
- El uso de cargadores genera limitaciones en las diferentes tareas asignadas, debido a la interferencia entre cables y en el área de trabajo se vuelve limitada reduciendo el campo de acción.

## 7.2 RECOMENDACIONES

- La determinación de la fuerza por medio de un sensor de contacto no es una alternativa confiable, las variantes existentes en la posición de la caja pueden afectar la determinación del número de robots necesarios para realizar la tarea asignada, por eso se recomienda explorar nuevas alternativas como un sensor de fuerza.
- Se sugiere el uso de una memoria de mayor capacidad para lograr un mayor grado de inteligencia en los robots.
- Se recomienda realizar un estudio más detallado de las características técnicas y su robustez (sensores, memoria, motores) debido a que estos elementos son necesarios en el desarrollo de proyectos de robótica cooperativa.
- Al ser una plataforma que permite el uso de múltiples aplicaciones cooperativas así como individuales, por lo que se sugiere integrar este sistema como una herramienta de apoyo pedagógico en el área de robótica.
- Si bien existe error en la trayectoria, podemos afirmar, el error puede ser disminuido, mediante la actualización del programa desarrollado para la sincronización de velocidades. Se recomienda tener presente los errores provocados por la estructura mecánica, los motores y las dimensiones del robot.
- Para trabajos futuros de investigación en el tema de robótica cooperativa se sugiere implementar un control visual con el fin de tener un robot autónomo, y se desplace por cualquier tipo de ambiente y no sólo se lo restrinja a un área de trabajo determinada.
- Se sugiere para futuras investigaciones la necesidad de usar una batería por las ventajas de movilidad.

## Bibliografía

- A. Correa, Sanabria, F., & Sandoval, A. (Mayo de 2010). Robot de servicio Cooperativo. Bogota, Colombia: Universidad Santo Tomas.
- Beltrán, A. (2012). Desarrollo de algoritmos para la coordinación de un sistema multi robots cooperativos para tareas de búsqueda de fuentes de calor en entornos dinámicos. Colombia: Centro de Investigación Robótica Móvil Autónoma.
- Cairo, O., & Olarte, J. G. (s.f.). *Loubeth: An Intelligent Agent for Electronic Negotiation*. Obtenido de [www.lsi.us.es/iberamia2002/confman/SUBMISSIONS/213-svtamamcai.pdf](http://www.lsi.us.es/iberamia2002/confman/SUBMISSIONS/213-svtamamcai.pdf)
- Camazine, S., Deneuborg, F., Sneyd, J., & Bonabeau, E. (2001). *Self - organization in biological system*. Princeton Univerty.
- Chantemargue, F., Courant, M., & Hirsbrunner, B. (2007). Autonomous Agents and Collective Robotics. Fribourg, Suiza: Universidad de Fribourg.
- Fischertechnik. (Diciembre de 2010). *Fischertechnik Home*. Obtenido de [www.fischertechnik.de/en/Home.aspx](http://www.fischertechnik.de/en/Home.aspx)
- Fischertechnik. (2011). *The TX brick accepts a wide variety of inputs*. Obtenido de [stefanbrunner.com/fischertechnik-tx-inputs](http://stefanbrunner.com/fischertechnik-tx-inputs)
- Fischertechnik. (Diciembre de 2012). *Eight MOSFET driven Power Outputs*. Obtenido de [stefanbrunner.com/fischertechnik-tx-outputs](http://stefanbrunner.com/fischertechnik-tx-outputs)
- GNUPro. (mayo de 2010). *Compiler issues for ARM and THUMB targets*. Obtenido de [www.filibeto.or/unix/tru64/lib/osscc/doc/cygnusdoc-99/html/6embed/embARM.html#ARMcompiler](http://www.filibeto.or/unix/tru64/lib/osscc/doc/cygnusdoc-99/html/6embed/embARM.html#ARMcompiler)
- Hípola, P., & Vargas Quesada, B. (01 de 2013). *Agentes de Información*. Obtenido de Agentes Inteligentes: definición y tipología: <http://eprints.rclis.org/18300>
- J.García. (2007). Modelo de un grupo de robots colaboradores cazadores. Barcelona, España.

- Jennings , N., & Wooldridge, M. (2002). *Agent Technology foundations applications and markets*. New York : Unicom.
- Jiménez Builes, J. A., Ovelle Carranza, D., & Branch Bedoya, J. (2008). Recuperado el 10 de 11 de 2012, de Comunicación en Sistemas de Múltiples Robots desde la Metodología MAD-SMART: <http://redelyc>
- Julián, V., & Botti, V. (5 de Julio de 2010). *www.ati.es*. Obtenido de [www.ati.es/novatica/2000/145/vjulia-145.pdf](http://www.ati.es/novatica/2000/145/vjulia-145.pdf)
- Larios, V. (2004). *Advanced Distributed Systems Third International School and Symposium*. México.
- León Fernández de Lara, Y., & Muñoz Meléndez, A. (2007). Obtenido de Investigación sobre los Requerimientos del movimiento Colectivo Coordinado en un Sistema Multi-Agentes: <http://redalyc.uaemex.mx/pdf/925/92503406.pdf>
- Marínez, M. (Febrero de 2011). *Sistema de control colaborativo de robots autónomos*. Obtenido de [www.e-archivo.uc3m.es/bitstream/10016/11828/1/PFCMoisesMartinezMuñoz.pdf](http://www.e-archivo.uc3m.es/bitstream/10016/11828/1/PFCMoisesMartinezMuñoz.pdf)
- Mikelecý, P., Lisková, T., & Cech, P. (2009). *Ambient Intelligence Perspectives* . Amsterdam: IOD.
- Palacios García, A. C. (2009). Obtenido de Aprendizaje Cooperativo de Conceptos para Sistemas Multi-Robots: <http://robotic.inaoep.mx/Tesis/2009-ACPG.pdf>
- Picard, G. (2006). *Cooperative Agent Model Instantiation to Collective Robotics*. Francia: Universidad Paul Sabater.
- Roa, J., Guitierrez, M., & Stegmayer, G. (Diciembre de 2008). *Framework para la enseñanza de agentes en IA*. Argentina.
- Romero Ternero, M. C. (2009). Obtenido de Sistemas Multi Agentes: [www.dte.us.es/personal/mcromero/masredes/docs/SMARD.0910.mas.pdf](http://www.dte.us.es/personal/mcromero/masredes/docs/SMARD.0910.mas.pdf)
- Russell, S., & Norvig, P. (2004). *Artificial Intelligence A Modern* . Madrid: Person Education.

Salvador, J. (2010). Modelo de ciclo de vida para el aprendizaje basado en compartición de conocimientos en sistemas de robots. Plata, Argentina:

Universidad Nacional de la Plata.

Sayde Alcántara, S., Fernández , C. A., & Ruiz Rodríguez, R. (2008). Obtenido de Una Herramienta para el Desarrollo y simulacion de Sociedades de

Agentes Reactivos: [http://www.utm.mx/~caff/doc/paperInAge\\_v4.pdf](http://www.utm.mx/~caff/doc/paperInAge_v4.pdf)

Sempere, M., Aznar, F., Pujol, M., & Rizo, R. (2011). Implementación de una arquitectura de agentes para robótica de enjambre utilizando JADE.

Universidad de Alicante.

## Índice de Figuras

Figura I-1: Estructura del Capítulo I.....	1
Figura II-1: Estructura del Capítulo II.....	7
Figura II-2: Sistema Multi – Robot Muy Acoplado .....	10
Figura II-3 Ejemplo de Sistema Multi –Robot Poco Acoplado.....	11
Figura II-4: Arquitectura Centralizada con Robots Lego .....	14
Figura II-5: Arquitectura Distribuida .....	15
Figura II-6: Agente Simple.....	16
Figura II-7: Agente Inteligente.....	18
Figura II-8: Agente Basado en Metas.....	21
Figura II-9: Funciones de un Sistema Multi agentes (SMA).....	23
Figura III-1: Estructura del Capítulo III .....	27
Figura III-2 : Disposición Básica de las Primitivas de Control .....	28
Figura III-3: Modelos de Razonamiento según la Disposición de las Primitivas de Control .....	29
Figura III-4: Estructura del Paradigma Deliberativo. ....	30
Figura III-5: Arquitectura Implementada para el Desarrollo del Proyecto.....	31
Figura III-6: Descripción Física del Controlador Tx.....	36
Figura III-7: Divisor de Tensión en Entrada Análoga 5[k $\Omega$ ].....	37
Figura III-8: Modo de Funcionamiento Entrada Análoga 10[V].....	37
Figura III-9: Principio Básico de Funcionamiento de un Sensor Ultrasónico .....	41

Figura III-10: Cobertura Horizontal del Sensor Ultrasónico .....	42
Figura III-11: Cobertura Vertical del Sensor Ultrasónico .....	42
Figura III-12: Rango de Luz con el cual Trabaja el sensor.....	43
Figura III-13: Principio de Funcionamiento del Sensor IR .....	44
Figura III-14: Sensor de Contacto .....	44
Figura III-15: Principio de Funcionamiento del Sensor Magnético .....	45
Figura III-16: Comunicación entre los Dos Robots.....	46
Figura III-17: El Principio de una Caja de Reducción.....	47
Figura III-18: Dimensiones del Área de Trabajo.....	48
Figura III-19: Prototipo 1.0 .....	48
Figura III-20: Prototipo 1.0, Rango de Detección del Sensor Ultrasónico .....	49
Figura III-21: Ubicación del Sensor de Color en la Infraestructura .....	49
Figura III-22: Caja Pequeña para los Puntos de Fuerza.....	50
Figura III-23: Prototipo 1.0 y su Área de Trabajo .....	51
Figura III-24: Prototipo 1.1 .....	51
Figura III-25: Posición Puntos de Fuerza en la Caja.....	52
Figura III-26: Distancia Mínima entre el Robot la Caja y la Pared .....	52
Figura III-27: Área de Trabajo Modificada para Sensor IR .....	53
Figura III-28: Prototipo 1.1 y Área de Trabajo.....	53
Figura IV-1: Estructura del Capítulo IV .....	54
Figura IV-2: Descripción Gráfica del Problema.....	55

Figura IV-3: Solución Usando Robots .....	56
Figura IV-4: Sistema Multi Robots .....	57
Figura IV-5: Diagrama de Flujo General del Problema .....	60
Figura IV-6: Puntos de Contacto Dependiendo del Número de Robots .....	62
Figura IV-7: Análisis General de la Caja Liviana .....	64
Figura IV-8: Análisis del Coeficiente de Rozamiento.....	65
Figura IV-9: Análisis del Coeficiente de Rozamiento de Material Tabla Prensada .....	66
Figura IV-10: Análisis del Centro de Masa Caja Liviana .....	66
Figura IV-11: Cara A de la Caja Liviana .....	67
Figura IV-12: Diagrama de Cuerpo Libre .....	67
Figura IV-13: Análisis General de la Caja Pesada .....	69
Figura IV-14: Análisis del Centro de Masa Pesada.....	70
Figura IV-15: Cara A de la Caja Pesada.....	70
Figura IV-16: Fuerzas Necesarias para Mover la Caja.....	71
Figura IV-17: Área de Trabajo para el Experimento 2 .....	73
Figura IV-18: Diagrama de Flujo del Sistema Multi – Robot .....	75
Figura IV-19: Fases para la Compilación y Ejecutable en Código Nativo .....	80
Figura V-1: Estructura del Capítulo V .....	92
Figura V-2: Prueba 1 Movimiento Lineal de una caja.....	93
Figura V-3: Prueba 1 Movimiento Lineal de un Robot.....	93
Figura V-4: Trayectoria Seguida por el Robot .....	98



Figura V-5: Robot mover una Caja desde el punto B al punto C ..... 98

Figura V-6: Robot mueva una Caja en una trayectoria Circular ..... 102

Figura V-7: Trayectoria Circular realizada por el robot ..... 106

Figura VI-1: Estructura del Capítulo VI ..... 114

Figura VII-1: Estructura del Capítulo VII ..... 125

## Índice de Tablas

Tabla II-1: Ventajas y Desventajas Robótica Cooperativa .....	12
Tabla II-2: Características de los Agentes .....	16
Tabla II-3: Clasificación de los agentes por su inteligencia .....	17
Tabla II-4 Características de un agente inteligente.....	19
Tabla II-5: Clasificación de los agentes por tipo de programación .....	20
Tabla II-6: Características de los Agentes Basados en Metas .....	22
Tabla III-1: Descripción de los dispositivos del Sistema Multi- Robot.....	33
Tabla III-2: Características Técnicas del Controlador.....	35
Tabla III-3: Comparación de las Entradas Análogas modo 10[v] y 5[kΩ].....	38
Tabla III-4: Comparación de las Entradas Digitales modo 10[v] y 5[kΩ].....	39
Tabla III-5: Comparación del Contador modo 10[v] y 5[kΩ] .....	40
Tabla IV-1: Agente Movedor de Cajas.....	58
Tabla IV-2: La Meta a Seguir una Trayectoria Lineal.....	58
Tabla IV-3: La Meta a Determinar el Número de Robots .....	59
Tabla IV-4: La Meta es Mover Cajas .....	59
Tabla IV-5: Datos Experimentales .....	68
Tabla IV-6: Herramientas de GNU .....	79
Tabla IV-7: Diagrama Estructurado Inicialización.....	81
Tabla IV-8: Diagrama Estructurado de Verificación de Peso .....	82

Tabla IV-9: Diagrama Estructurado de Posición Punto de Contacto.....	82
Tabla IV-10: Diagrama Estructurado de Posición Final de Robot Antes de Mover Una caja Pesada.....	83
Tabla IV-11: Diagrama Estructurado de Comunicaciones Bluetooth .....	83
Tabla IV-12: Diagrama Estructurado de Caminar .....	84
Tabla IV-13: Variables y Descripción Parte 1.....	84
Tabla IV-14: Variables y Descripción Parte 2.....	85
Tabla IV-15: Funciones de Inicialización.....	86
Tabla IV-16: Funciones de Configuración .....	87
Tabla IV-17: Funciones del Programa Parte 1 .....	87
Tabla IV-18: Funciones del Programa Parte 2 .....	88
Tabla IV-19: Funciones de Bluetooth .....	89
Tabla IV-20: Variables de Configuración .....	90
Tabla IV-21: Variables de Entradas .....	90
Tabla IV-22: Variables de Salidas.....	91
Tabla V-1: Puntos de partida del Robot .....	94
Tabla V-2: Puntos seguidos por el robot mediante una trayectoria lineal .....	94
Tabla V-3: Tabulación de datos para el método de ajuste lineal .....	96
Tabla V-4: Determinación de las revoluciones para el peso.....	99
Tabla V-5: Determinación la exactitud del robot .....	100
Tabla V-6: Resultados de tiempo en la Solución Completa .....	101
Tabla V-7: Resultados de la trayectoria realizada por el robot.....	103

Tabla V-8: Datos determinación de la ecuación general de la circunferencia.....	105
Tabla V-9: Resultados de la distancia recorrida en una trayectoria circular.....	107
Tabla V-10: Evaluación de Cooperación de los Agentes .....	108
Tabla V-11: Evaluación del Sistema Multi - agentes .....	109
Tabla V-12: Resultados Finales Obtenidos .....	110
Tabla V-13: Problemas del Proyecto.....	110
Tabla V-14: Características Técnicas del Motor “TANQUE” .....	111
Tabla V-15: Datos de “TANQUE” .....	111
Tabla VI-1: Robot Humanoide Ideal .....	118
Tabla VI-2: Robot Móvil Ideal.....	<b>¡Error! Marcador no definido.</b>
Tabla VI-3: Pruebas Realizadas con el Robot IRobot Create.....	119
Tabla VI-4: Características Técnicas del Robot Pioneer .....	123
Tabla VI-5: Características Físicas del Robot Pioneer .....	124