



CARTA DE AUTORIZACIÓN

CÓDIGO

AP-BIB-FO-06

VERSIÓN

1

VIGENCIA

2014

PÁGINA

1 de 2

Neiva, 27/septiembre /2018

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Neiva

Los suscritos:

Over Andrés Parra Villalba

con C.C. No. 1075237749,

Alex Humberto Chicue Osorio

con C.C. No. 1081514629,

Autores de la tesis y/o trabajo de grado titulado “Diseño e implementación de un prototipo de robot de transporte de carga con control de posición a través de un sistema de visión por computador.” presentado y aprobado en el año 2018 como requisito para optar al título de Ingeniero Electrónico;

Autorizamos al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que, con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

- Los usuarios puedan consultar el contenido de este trabajo de grado en los sitios web que administra la Universidad, en bases de datos, repositorio digital, catálogos y en otros sitios web, redes y sistemas de información nacionales e internacionales “open access” y en las redes de información con las cuales tenga convenio la Institución.
- Permita la consulta, la reproducción y préstamo a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato Cd-Rom o digital desde internet, intranet, etc., y en general para cualquier formato conocido o por conocer, dentro de los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia.

Vigilada Mineducación



CARTA DE AUTORIZACIÓN

CÓDIGO

AP-BIB-FO-06

VERSIÓN

1

VIGENCIA

2014

PÁGINA

2 de 2

- Continúo conservando los correspondientes derechos sin modificación o restricción alguna; puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación del derecho de autor y sus conexos.

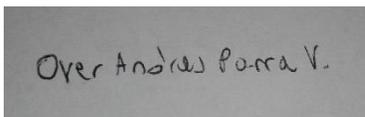
De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, “Los derechos morales sobre el trabajo son propiedad de los autores”, los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

EL AUTOR/ESTUDIANTE:

EL AUTOR/ESTUDIANTE:

Firma:

Firma:



Over Andrés Parra V.



Fredy Claudio Polo



TÍTULO COMPLETO DEL TRABAJO: Diseño e implementación de un prototipo de robot de transporte de carga con control de posición a través de un sistema de visión por computador.

AUTOR O AUTORES:

Primero y Segundo Apellido	Primero y Segundo Nombre
Parra Villalba	Over Andrés
Chicue Osorio	Alex Humberto

DIRECTOR Y CODIRECTOR TESIS:

Primero y Segundo Apellido	Primero y Segundo Nombre
Salgado Patrón	José de Jesús

ASESOR (ES):

Primero y Segundo Apellido	Primero y Segundo Nombre
----------------------------	--------------------------

PARA OPTAR AL TÍTULO DE: Ingeniero Electrónico

FACULTAD: Ingeniería

PROGRAMA O POSGRADO: Ingeniería Electrónica

CIUDAD: Neiva **AÑO DE PRESENTACIÓN:** 2018 **NÚMERO DE PÁGINAS:** 84

TIPO DE ILUSTRACIONES (Marcar con una X):

Vigilada mieducación



DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO

CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	2 de 4
---------------	---------------------	----------------	----------	-----------------	-------------	---------------	---------------

Diagramas Fotografías Grabaciones en discos Ilustraciones en general Grabados
Láminas Litografías Mapas Música impresa Planos Retratos Sin ilustraciones
Tablas o Cuadros

SOFTWARE requerido y/o especializado para la lectura del documento: Word, PDF Adobe Acrobat Reader

MATERIAL ANEXO:

PREMIO O DISTINCIÓN (*En caso de ser LAUREADAS o Meritoria*):

PALABRAS CLAVES EN ESPAÑOL E INGLÉS:

Español	Ingles	Español	Ingles
1. Matlab	Matlab	6. Interfaz	Interface
2. Mecanum	Mecanum	7. control multivariable	Multivariable control
3. Visión por computador	Computer vision	8. control de posición	Position Control
4. Prototipo	Prototype	9. Robot	Robot
5. Tratamiento de imágenes	Imagen processing	10. Patrón de referencia	Reference pattern

RESUMEN DEL CONTENIDO: (Máximo 250 palabras)

En este proyecto, se presenta el diseño e implementación de un prototipo de robot montacarga que usa control de posición y visión por computador para trasladar una carga de un lugar a otro de forma automática, dentro de un escenario definido. El prototipo diseñado puede levantar y transportar una carga asignada de hasta (2.5) Kilogramos. El chasis del robot fue construido en aluminio para otorgarle características de rigidez, resistencia y poco peso mejorando el desempeño de sus movimientos. El robot logra moverse en diferentes direcciones gracias a sus ruedas omnidireccionales tipo mecanum. El prototipo es guiado con ayuda de un sistema de visión por computador que indicará donde está ubicada la carga, y el robot en todo momento. Adicionalmente posee una interfaz gráfica



DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO

CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	3 de 4
--------	--------------	---------	---	----------	------	--------	--------

desarrollada en MATLAB que ejecutará todo el conjunto de operaciones (procesamiento de imagen, control de movimientos, comunicación, trayectorias, etc.) que requiere el sistema.

ABSTRACT: (Máximo 250 palabras)

In this project, we present the design and implementation of a forklift robot prototype that uses position control and computer vision to move a product from one place to another automatically, within a defined stage. The designed prototype can lift and transport an assigned load of up to (2.5) Kilograms. The robot's chassis was constructed in aluminum to give it characteristics of rigidity, resistance and light weight improving the performance of its movements. The robot can move in different directions due to its omnidirectional mecanum wheels. The prototype is guided by computer vision that will indicate where the load is located, and the robot at every time. Additionally it has a graphical interface developed in

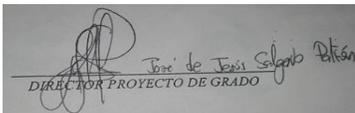


MATLAB that will execute the whole set of operations (image processing, movement control, communication, trajectories, etc.) that the system requires.

APROBACION DE LA TESIS

Nombre Presidente Jurado: José de Jesús Salgado Patrón

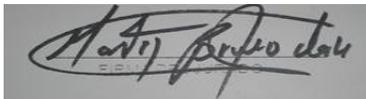
Firma:



José de Jesús Salgado Patrón
DIRECTOR PROYECTO DE GRADO

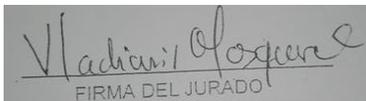
Nombre Jurado: Martín Diomedes Bravo Obando

Firma:



Nombre Jurado: Vladimir Mosquera Cerquera

Firma:



Vladimir Mosquera Cerquera
FIRMA DEL JURADO

Vigilada mieducación

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ROBOT DE
TRANSPORTE DE CARGA CON CONTROL DE POSICIÓN A TRAVÉS DE UN
SISTEMA DE VISIÓN POR COMPUTADOR**

**OVER ANDRÉS PARRA VILLALBA
ALEX HUMBERTO CHICUE OSORIO**

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA HUILA
2018**

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ROBOT DE
TRANSPORTE DE CARGA CON CONTROL DE POSICIÓN A TRAVÉS DE UN
SISTEMA DE VISIÓN POR COMPUTADOR**

**OVER ANDRÉS PARRA VILLALBA
ALEX HUMBERTO CHICUE OSORIO**

**Trabajo de Grado presentado como requisito para optar al título de ingeniero
electrónico**

Director:

JOSÉ DE JESÚS SALGADO PATRÓN

Magister: Ingeniería Electrónica y de Computadores

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA HUILA
2018**

Nota de Aceptación:

Firma del presidente del Jurado

Firma del Jurado

Firma del Jurado

DEDICATORIA

Es la culminación de un proceso que inicio algunos años, quiero aprovechar estas líneas para agradecer de la manera más sincera a mis padres y hermanos por su paciencia, comprensión, aconsejándome para no dejar de intentarlo durante todo este tiempo, por avanzar siempre a mi lado con el deseo colectivo de un futuro mejor.

A demás amigos y compañeros que estuvieron presentes en este proceso, muchas gracias.

Alex Humberto Chicue Osorio

DEDICATORIA

“Y todo lo que hacéis, sea de palabra o, de hecho, hacedlo todo en el nombre del Señor Jesús, dando gracias a Dios Padre por medio de él.” Colosenses 3:17

Quiero empezar agradeciendo al SEÑOR JESUS ya que suyo es el reino, el poder, la gloria y además que siempre me ha acompañado a librar las batallas más difíciles de mi vida cubriéndome siempre con su manto santo. A mi mamá y a mi abuela las personas que adoro con mi corazón y siempre a pesar de todo han creído ciegamente en mí y nunca me han dejado desfallecer, siempre me acompañaron y me apoyaron en toda esta etapa que culmina. A mi papá que a la distancia y a su manera ha estado pendiente de mi proceso educativo y me ha brindado su mano en momentos cruciales. Al ingeniero José de Jesús Salgado Patrón por su compromiso, dedicación y todo el valioso aporte en la dirección de este proyecto, además por todo el conocimiento que me brinda a lo largo de todas las materias que me dictó en la carrera.

Quiero agradecer también a todos y cada uno de los ingenieros que aportaron su grano de arena para formar un nuevo ingeniero electrónico. Por último, quiero dar las gracias a todas y cada una de las personas con las que he compartido esta etapa en la cual estuvieron alentándome, ayudándome y apoyándome. ¡Solo me queda por decirles nuevamente a todos INFINITAS GRACIAS!!!

Over Andrés Parra Villalba

AGRADECIMIENTOS

Los autores expresan sus más sinceros agradecimientos a:

JOSÉ DE JESÚS SALGADO PATRÓN, Ingeniero Electrónico, Magister en Ingeniería Electrónica y de Computadores, director del presente trabajo por la disponibilidad de su tiempo y conocimiento para dirigir este proyecto, supervisión y el aporte continuo al mejoramiento de este.

GERMAN EDUARDO MARTINEZ BARRETO, Ingeniero Electrónico, Especialista en Automatización, jurado del anteproyecto de grado.

VLADIMIR MOSQUERA CERQUERA, Ingeniero Electrónico, Magister en Ingeniería Electrónica, jurado del proyecto de grado.

MARTIN DIOMEDES BRAVO OBANDO, Ingeniero Electrónico, Especialista en desarrollo de software orientado a objetos, Magister en Ingeniería Eléctrica, jurado del proyecto de grado.

TABLA DE CONTENIDO

	Pag.
INTRODUCCIÓN	15
1. OBJETIVOS	17
1.1 Objetivo General.....	17
1.2 Objetivos Específicos	17
2. ESTADO DEL ARTE	18
2.1 EVOLUCIÓN DE LA ROBÓTICA MÓVIL	18
2.2 ACTUALIDAD EN COLOMBIA.....	19
3. DISEÑO E IMPLEMENTACIÓN DEL HADWARE	21
3.1 ELABORACIÓN DE LA ESTRUCTURA MECÁNICA: CHASIS.....	21
3.2 SELECCIÓN DEL SISTEMA DE LOCOMOCIÓN	27
3.2.1 Movimientos del robot de carga.....	30
3.2.2 Torque de los motores.....	31
3.2.3 Determinación del momento de inercia (Jm) de los motores.....	31
3.2.4 Corriente de arranque del motor y Torque de fricción	32
3.2.5 Motores del Sistema de Locomoción.....	33
3.3 SELECCIÓN DEL SISTEMA DE LEVANTAMIENTO DE LA CARGA	34
3.4 ARMADO DEL SISTEMA MECÁNICO	36
3.5 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA ELECTRÓNICO DEL PROTOTIPO	38
3.5.1 Regulador de voltaje de 12 Voltios a 5 Voltios	40
3.5.2 Selección del microcontrolador.....	40
3.6 DISEÑO E IMPLEMENTACIÓN PLACA BASE	41
4. TRATAMIENTO DIGITAL DE LA IMAGEN.....	44
4.1 PRE-PROCESAMIENTO	44
4.1.1 Imagen Original	44
4.1.2 Imagen en escala de grises	45
4.1.3 Imagen en blanco y negro (umbral manual)	46
4.2 Analisis de contornos	46
4.2.1 Extracción de Características	46
4.3 DETECCION DEL PATRON RECTANGULAR.....	47

4.3.1	Filtro por área (descartar contornos muy pequeños)	48
4.3.2	Filtro por área (descartar contornos muy grandes).....	49
4.3.3	Filtro por relación de lados	49
4.3.4	Filtro por área en centímetros cuadrados (cm ²) aproximado	50
4.3.5	Filtro por color del patrón rectangular (centro).....	50
4.4	DETECCION DE PATRONES CIRCULARES.....	51
4.4.1	Filtro por área (descartar contornos muy pequeños)	52
4.4.2	Filtro de contornos dentro del patrón.	52
4.4.3	Filtro por relación de lados	53
4.4.4	Filtro por color del patrón circular (centro) y área	53
5.	SISTEMA DE CONTROL	56
6.	GENERACIÓN DE TRAYECTORIAS.....	57
7.	DISEÑO DE LA INTERFAZ GRÁFICA	60
8.	RESULTADOS	62
9.	CONCLUSIONES.....	66
10.	RECOMENDACIONES	68
	BIBLIOGRAFÍA	69
	ANEXOS	70
	ANEXO A: MONTAJE DEL SISTEMA DE COMUNICACIÓN	70
	ANEXO B: CODIGO DEL PROCESAMIENTO DE LA IMAGEN	74
	ANEXO C: CODIGO DE LA GENERACION DE LA TRAYECTORIA.....	76
	ANEXO D: MANUAL DE USUARIO DE LA INTERFAZ	81

TABLA DE FIGURAS

	Pág.
Figura 1. Diagrama de Bloques del Proyecto	16
Figura 2. Robot Uranus (universidad de Michigan)	18
Figura 3. Robot Seekur (active Media).	19
Figura 4. Diagrama de Bloques del Prototipo.	21
Figura 5. Diseño del Chasis Del Robot.....	22
Figura 6. Primera simulación: punto central.	23
Figura 7. Primera simulación: coordenada X, Simulaciones de desplazamiento..	24
Figura 8. Primera simulación: coordenada Y, Simulaciones de desplazamiento..	24
Figura 9. Primera simulación: coordenada Z, Simulaciones de desplazamiento..	25
Figura 10. Segunda simulación: punto frontal	25
Figura 11. Simulaciones de desplazamiento en ejes X.	26
Figura 12. Simulaciones de desplazamiento en ejes Y.	26
Figura 13. Simulaciones de desplazamiento en ejes Z.	27
Figura 14. Movimiento omnidireccional.	28
Figura 15. Llantas mecanum.	29
Figura 16. Acoples plásticos para sujeción de llantas.	29
Figura 17. Vista superior de las ruedas	30
Figura 18. Disposición de ruedas Mecanum y sistema de coordenadas.....	30
Figura 19. Motor reductor metálico 131:1	33
Figura 20. Torque	34
Figura 21. Sistema Montacargas convencional con motor eléctrico.....	35
Figura 22. Sistema Montacargas prototipo Robot de carga.	35
Figura 23. Estiba para productos.....	36
Figura 24. Estiba para productos.....	36
Figura 25. Diseño del sistema mecánico completo.	37
Figura 26. Diseño del sistema mecánico completo, protección.....	37
Figura 27. Diagrama de Bloques Sistema Electrónico	38
Figura 28. Driver dual para motores (Full Bridge) L298n.....	38
Figura 29. Batería Lipo 3s1p 5100 mAh 11.1 V.....	39
Figura 30. Indicador de voltaje de 8 celdas.	39
Figura 31. Regulador LM 7805.	40
Figura 32. Microcontrolador Pic 16f877a.....	40
Figura 33. Etapas del Robot de Carga.	41
Figura 34. Robot de Carga.	42
Figura 35. Diagrama de Bloques Tratamiento digital de la Imagen.	44
Figura 36. Diagrama de Bloques Pre-Procesamiento digital de la Imagen.	44
Figura 37. Imagen RGB.....	45
Figura 38. Imagen en escala de grises.....	45
Figura 39. Imagen a blanco y negro.	46
Figura 40. Análisis de contorno.	47
Figura 41. Diagrama de Bloques de la detección del patrón rectangular.	47

Figura 42. Descarte de contornos.	48
Figura 43. Filtro por área.	48
Figura 44. Descarte de contornos muy grandes.	49
Figura 45. Filtro por relación de lados	49
Figura 46. Filtro por cm ²	50
Figura 47. Filtro por color del patrón rectangular.	50
Figura 48. Diagrama de Bloques Detección de Patrones Circulares.	51
Figura 49. Análisis de patrones internos.	51
Figura 50. Filtrado descarte de contornos pequeños.	52
Figura 51. Filtro del contorno dentro del patrón.	52
Figura 52. Filtro por relación de lados.	53
Figura 53. Filtro por color de patrón circular.	54
Figura 54. Filtro por área real en cm ²	55
Figura 55. Diagrama de bloque Controladores.	56
Figura 56. Puntos de referencia (1 Robot, 2 Llegada, 3 Carga).	57
Figura 57. Puntos para Generar trayectoria.	58
Figura 58. Trayectoria.	59
Figura 59. Diagrama inicial, Casos de usos	60
Figura 60. Escenario 1	62
Figura 61. Escenario 1, posición 2.	63
Figura 62. Escenario 1, posición 2.	63
Figura 63. Escenario 2.	64
Figura 64. Escenario 2, posición 1.	64
Figura 65. Escenario 2, posición 2.	65
Figura 66. Módulo XBEE s1.	70
Figura 67. Módulo XBEE EXPLORER DONGLE.	71
Figura 68. Configuración del módulo XBEE transparente.	71
Figura 69. Interfaz XCTU.	72
Figura 70. Selección del puerto.	72
Figura 71. Configuración de puertos.	73
Figura 72. Componentes Interfaz.	81
Figura 73. Pop menú para seleccionar la resolución de la cámara.	82
Figura 74. Pop menú donde se selecciona el puerto serial disponible.	82
Figura 75. Interfaz gráfica en funcionamiento.	83

LISTA DE TABLAS

	Pág.
Tabla 1. Tensión de rotura de algunos materiales.....	21
Tabla 2. Propiedades del aluminio 6061.....	22
Tabla 3. Movimiento de la Estiva.....	59
Tabla 4. Resultados escenario 1, posición 1.....	63
Tabla 5. Resultados escenario 1, posición 3.....	64
Tabla 6. Resultados escenario 2, posición 1.....	65
Tabla 7. Resultados escenario 2, posición 2.....	65

LISTA DE ECUACIONES

	Pag.
Ecuación 1. Tiempo mecánico	31
Ecuación 2. Momento de Inercia	31
Ecuación 3. Torque	32
Ecuación 4. Corrientes	32
Ecuación 5. Potencia	33

RESUMEN

En este proyecto, se presenta el diseño e implementación de un prototipo de robot monta-carga que usa control de posición y visión por computador para trasladar una carga de un lugar a otro de forma automática, dentro de un escenario definido.

El prototipo diseñado puede levantar y transportar una carga asignada de hasta (2.5) Kilogramos. El chasis del robot fue construido en aluminio para otorgarle características de rigidez, resistencia y poco peso mejorando el desempeño de sus movimientos. El robot logra moverse en diferentes direcciones gracias a sus ruedas omnidireccionales tipo mecanum.

El prototipo es guiado con ayuda de un sistema de visión por computador que indicará donde está ubicada la carga, y el robot en todo momento. Adicionalmente posee una interfaz gráfica desarrollada en MATLAB que ejecutará todo el conjunto de operaciones (procesamiento de imagen, control de movimientos, comunicación, trayectorias, etc.) que requiere el sistema.

El proceso de visión por computador se realiza mediante capturas constantes de una cámara ubicada en la parte superior del escenario, que observa el movimiento del prototipo y la carga a desplazar. Por cada imagen se realiza un procesamiento para identificar patrones gráficos que ayudan a reconocer las distancias que precisa el sistema de control. El sistema de visión se diseñó de forma robusta para que el robot pueda trabajar sobre superficies de cualquier color y acoplarse a las dimensiones del escenario.

Los datos obtenidos del procesamiento de imagen y el control de posición son transmitidos al robot por medio de una comunicación inalámbrica con módulos xbee. En dicha transmisión se envían los datos necesarios para que el robot realice movimientos en tres direcciones (lateral, longitudinal y rotación), y que además levante y suelte la carga por medio de un sistema convencional de uñas y estiba.

Este proyecto se realiza como iniciativa del área de robótica del programa de ingeniería electrónica, buscando ofrecer soluciones al sector industrial de la región surcolombiana en materia de movilidad y organización de productos en bodegas de almacenamiento.

Palabras claves: MATLAB, mecanum, montacarga, robot, prototipo, visión por computador.

ABSTRACT

In this project, we present the design and implementation of a forklift robot prototype that uses position control and computer vision to move a product from one place to another automatically, within a defined stage.

The designed prototype can lift and transport an assigned load of up to (2.5) Kilograms. The robot's chassis was constructed in aluminum to give it characteristics of rigidity, resistance and light weight improving the performance of its movements. The robot can move in different directions due to its omnidirectional mecanum wheels.

The prototype is guided by computer vision that will indicate where the load is located, and the robot at every time. Additionally it has a graphical interface developed in MATLAB that will execute the whole set of operations (image processing, movement control, communication, trajectories, etc.) that the system requires.

The process of computer vision is made by constant capturing of a camera located at the top of the stage, which observes the movement of the prototype and the load to be displaced. For each image, a processing is performed to identify graphical patterns that help to recognize the distances required by the control system. The vision system was designed in a robust way so the robot can work on surfaces of any color and fit the dimensions of the stage.

The data obtained from image processing and position control are transmitted to the robot via wireless communication by xbee modules. In this transmission the necessary data are sent to the robot to make movements in 3 directions (lateral, longitudinal and rotation), and also lift and release the load by a conventional system of nails and stowage.

This project is carried out as an initiative of the robotics area of the electronic engineering program, seeking to offer solutions to the industrial sector of the South-Colombian region in terms of product movility and organization in storage warehouses.

Key words: MATLAB, mecanum, forklift, robot, prototype, computer vision.

INTRODUCCIÓN

El afán de la industria mundial por optimizar sus procesos, mejorar la calidad de sus productos y de hacer un uso eficiente con el propósito de ser cada vez más competitivos en el mercado, han incentivado la inserción y desarrollo de nuevas tecnologías que garanticen elevar esos niveles de competencia basados en una combinación de cooperación entre el factor humano-máquina.

“Por esta razón las ciencias que ha tomado un mayor auge ha sido la robótica, que se define como la combinación de diversas disciplinas para el diseño, construcción y operación de máquinas capaces de hacer trabajos independientes y/o de cooperación con el ser humano.”¹

La robótica asociada con los sistemas de visión por computador generan sistemas autónomos más efectivos, teniendo en cuenta que un sistema de robot móvil la visión por computador da la posibilidad de planificar un conjunto de movimientos del autómeta; así, una vez dada la información local del entorno donde se encuentra el prototipo o el ambiente que lo rodea se puede realizar un control del desplazamiento del mismo, ejecutando acciones de acuerdo a la información suministrada que permiten alcanzar el objetivo con un rendimiento de calidad, seguro y óptimo funcionamiento del diseño.²

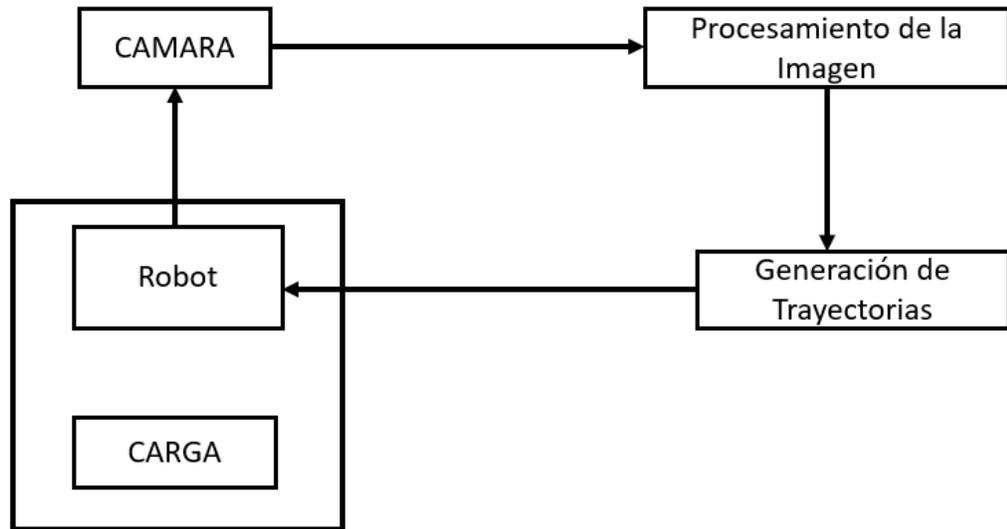
En la región SurColombiana los procesos industriales aún no están totalmente automatizados y en cuestiones de levantamiento los sistemas de cargas se encuentran en desarrollo, realizando los procesos de forma manual especialmente en la industria ferretera donde el humano tiene que “echarse al hombro, un peso superior al permitido el cual puede perjudicar considerablemente su salud”, o en otros casos se utilizan animales de carga para realizar dichas funciones. Adicionalmente el proceso de traslado de recipientes con contenidos químicos implica un contacto directo generando un nivel de riesgo alto para su integridad.

Ante la necesidad de crear una máquina que fuera capaz de realizar el trabajo de los humanos en levantamiento de cargas, se propuso realizar un prototipo de robot con control de posición por medio de un sistema de visión por computador, con la funcionalidad de ejecutar las actividades automáticamente. El prototipo se diseñó para que logre recoger, transportar y descargar una carga no superior a los 2.5 kg desde un punto inicial a un punto final.

¹ JOHN J. CRAIG, Robótica, 1. Introducción, 1.1. Antecedentes Alfonso Vidal Romero Elizondo, ITESM -Campus Ciudad de México 2006. p. 18.

² JOHN J. CRAIG, Robótica, 2. Descripciones espaciales y transformaciones, 2.10. Consideraciones Computacionales, Alfonso Vidal Romero Elizondo, ITESM -Campus Ciudad de México 2006. p. 53.

Figura 1. Diagrama de Bloques del Proyecto



Fuente: Autores

El proceso se dividió en las siguientes etapas:

- Elaboración de la Estructura Mecánica
- Selección del Sistema de Locomoción
- Selección y Elaboración del Sistema de Levantamiento
- Diseño y Elaboración del Sistema Electrónico
- Tratamiento Digital de la Imagen
- Sistema de Control
- Generación de Trayectorias
- Resultados y pruebas calidad.
- Control de cambios y conclusiones.

1. OBJETIVOS

1.1 OBJETIVO GENERAL

Diseñar e implementar un robot móvil capaz de recorrer un entorno controlado y llevar una carga de máximo 2.5 Kg de un punto determinado a otro, usando técnicas de visión por computador para definir la trayectoria y controlar el recorrido del robot, al finalizar el proyecto se pretende donar a la Universidad SurColombiana.

1.2 OBJETIVOS ESPECÍFICOS

- Seleccionar el tipo de locomoción del robot móvil, entre orugas y ruedas, de acuerdo con la carga y el entorno.
- Seleccionar el tipo de mecanismo de sujeción y recolección de la carga.
- Diseñar la estructura mecánica (chasis y actuadores) del robot móvil de tal manera que sea capaz de llevar una carga de máximo 2.5 Kg, incluyendo el tipo de locomoción y el tipo de elemento para recoger la carga.
- Diseñar el sistema electrónico del robot.
- Seleccionar el sistema de adquisición de imágenes que se adecue más al entorno.
- Realizar un algoritmo de visión por computador, por medio de Matlab, que genere las trayectorias y controle la posición del robot en ambientes de diferentes escenarios por donde se movilizará el robot.
- Implementar un sistema de comunicación inalámbrica entre el robot y la estación de comandos.
- Diseñar una interfaz gráfica donde se muestre el desplazamiento del robot y la trayectoria generada.
- Poner a punto el sistema en conjunto.

2. ESTADO DEL ARTE

2.1 EVOLUCIÓN DE LA ROBÓTICA MÓVIL

Los robots móviles han sido satisfactoriamente utilizados con fines prácticos, por ejemplo, en la exploración de otros planetas. El 4 de julio de 1997 el rover Sojourner se convirtió en el primer robot móvil en llegar a la superficie de Marte. Era controlado por control remoto desde la Tierra, aunque incluía navegación autónoma usando un láser para detectar la presencia de obstáculos. En 2004 llegaron al Planeta Rojo los rovers gemelos Spirit y Opportunity dentro de la misión de exploración Mars Exploration Rover. Pertenecen a un modelo tecnológicamente más avanzado que el Sojourner. En este nuevo modelo se ha mejorado cualitativamente la navegación gracias a la incorporación de un sistema de visión estereoscópica para el reconocimiento de obstáculos. Ambos rovers siguen en funcionamiento sobre la superficie de Marte a día de hoy.

En 1999 Sony lanzó al mercado AIBO. Se trata de un robot mascota con apariencia de perro. Dotado de un gran número de sensores, AIBO es capaz percibir estímulos externos provenientes del entorno o de su dueño y actuar en consecuencia³.

En los últimos años se han empezado a comercializar robots diseñados para realizar tareas domésticas de forma autónoma, entre ellos, se pueden nombrar el robot limpiador Roomba de iRobot y el cortador de césped Automower de Husqvarna.

La siguiente figura (Figura 2) muestra el robot omnidireccional Uranus desarrollado en la Universidad de Michigan y el conjunto de movimientos posibles que el mismo puede desarrollar, de acuerdo a la rotación de cada una de las ruedas el robot puede avanzar, girar o desplazarse lateralmente sin necesidad de reorientarse⁴.

Figura 2. Robot Uranus (universidad de Michigan)



Fuente: Tomado de, researchgate.net/figure/Uranus-omnidirectional-mobile-robot-Muir-Neuman-1987-The-benefits-of-a-vehicle-with_fig1_221786657

³ FU, K.S., González, R.C. y Lee, G.S.G. *Robótica, Control, Detección, Visión e Inteligencia*. Mc Graw-Hill, 1990. ISBN: 9789684223578 3

⁴ Bambino IL, *Una Introducción a los Robots Móviles*, 2008

En la siguiente figura (Figura 3) se presenta otra versión de robot omnidireccional con ruedas orientables centradas, en este caso el robot puede cambiar la dirección de su movimiento simplemente cambiando la orientación de las ruedas, este movimiento sincronizado se puede lograr por medios mecánicos, empleando sistemas de tracción y dirección por correas o por medios electrónicos mediante señales de accionamientos simultáneas a partir de la electrónica que comanda cada uno de los motores en las ruedas.

Figura 3. Robot Seekur (active Media).



Fuente: Tomado de, generationrobots.com/en/402404-robot-mobile-seekur.html

Uno de los robots de carga es llamado TITAN, que cuenta con una capacidad de carga de 1000 kilogramos y un alcance de 3200 m.m. Es un robot industrial que posee seis ejes siendo uno de los más grandes y fuertes en el mundo, puede asumir las tareas de una grúa que pesa más y elevar segmentos de escaleras de hormigón. Gracias a su estabilidad y fuerza este robot no tiene ningún problema en levantar cristales de gran espesor o grandes piezas de fundición, esto sin disminuir ni precisión ni exactitud.

A nivel internacional, los robots utilizados para la manipulación de carga son los robots industriales, básicamente se pueden clasificar según su desarrollo, capacidad y aplicación en la industria, así el mercado ofrece desde los brazos robóticos más básicos que desempeñan labores simples de manipulación, hasta sofisticados equipos que utilizan sensores.

2.2 ACTUALIDAD EN COLOMBIA

Colombia cuenta con una compañía dedicada a la fabricación de brazos robóticos para la industria, se trata de Micromac, un proyecto liderado por la Gobernación de Risaralda y la Cámara de Comercio de Dosquebradas con el apoyo del Fondo Colombiano para el Desarrollo de las Mipymes (FoMipyme). Desde hace dos años, con diseño nacional y un 95 por ciento de materiales colombianos, cuatro ingenieros liderados por el diseñador y desarrollador Pablo Pérez Santiago, emprendieron la

tarea de construir dos tipos de equipos robotizados especiales para la industria nacional, de aplicación para la soldadura MIG-MAG, el corte con plasma, paletizado, pintura y manipulación de piezas y herramientas. Hasta el momento se han construido dos prototipos que están trabajando con éxito en una de las compañías metalmecánicas de Risaralda, actualmente se tienen cerca de 40 solicitudes de compra. El primer robot fue nombrado Axon LV 04, este cuenta con cuatro grados de libertad, una capacidad de carga máxima de 4.0 Kg, área de trabajo de 2.0 m y alcance vertical de 80 cm. El segundo equipo es se bautizó Axon LV J3 05, diseñado con seis grados de libertad, para un área de trabajo de 2.40 m y un alcance vertical de 1.70 m, precisión de 0.5 mm. Ambos tienen disponibles sistemas de visión básicos por pixel, sistemas porta pinzas, mesa de trabajo y posibilidad de desplazamiento.

Actualmente, en la Universidad Nacional se está llevando a cabo un proyecto de robot móvil, para agilizar tareas domésticas basadas en dos aplicaciones ya existentes llamadas robot scooba y robot roomba, este proyecto busca ofrecer al hogar una herramienta con la cual se puedan adelantar labores de servicios domésticos, en menor tiempo, de forma automática y sin la presencia de un ser humano.

En la pontificia Universidad Javeriana, se están trabajando dos proyectos de robótica móvil que son: el proyecto amaranta, el cual consiste en un robot móvil que interviene en zonas de alto riesgo para el desminado humanitario; y el proyecto arcadio, el cual es un robot móvil con un brazo, diseñado para la desactivación de cargas explosivas dentro de la ciudad⁵.

⁵ [HTTP://fing.javeriana.edu.co/simposio/memorias/.../robotica/Presentación50.ppt](http://fing.javeriana.edu.co/simposio/memorias/.../robotica/Presentación50.ppt) 7

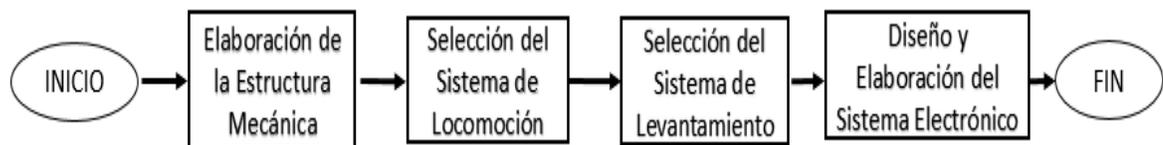
3. DISEÑO E IMPLEMENTACIÓN DEL HARDWARE

Para iniciar el proceso del diseño e implementación del hardware se tuvo en cuenta la funcionalidad que el prototipo requiere, que es levantar una carga de 2.5 Kg.

El proceso se dividió en las siguientes etapas que se muestran en la Figura 4:

- Elaboración de la estructura mecánica
- Selección del sistema de locomoción
- Selección y elaboración del sistema de levantamiento
- Diseño y elaboración del sistema electrónico

Figura 4. Diagrama de Bloques del Prototipo.



Fuente: Autores

3.1 ELABORACIÓN DE LA ESTRUCTURA MECÁNICA: CHASIS

Para la elaboración de la estructura mecánica o chasis del robot, se analizaron las características de tensión de rotura de diferentes materiales como el polipropileno, aluminio, cobre, acero A36 y aleación de aluminio estas características se muestran en la Tabla 1, dicho material debe ser adecuado, resistente y que a su vez presente características de peso ligeras.

Tabla 1. Tensión de rotura de algunos materiales.

Material	Polipropileno	Aluminio	Cobre	Acero A36	Aleación de aluminio
Tensión de Rotura (Mpa)	17.1 – 80	40 - 50	220	400	483

Fuente: <http://tecniaceros.com/pdfs/aluminio.pdf>, TecniAceros Aluminio 6061-T6 (AISI/ASTM).

Teniendo en cuenta las propiedades físicas, mecánicas y las aleaciones de los materiales anteriormente mencionados, se escogió el aluminio, en el que se encontró la aleación 6061⁶, en la Tabla 2 se puede observar las propiedades de la aleación, que es una fusión entre aluminio-silicio-magnesio que proporciona mejoras de dureza y resistencia a la corrosión.

⁶ alacermas.com, Se aplica en la industria para la fabricación de moldes troqueles, maquinaria. [alacermas.com/img/galeria/files/aluminio/chapa_6061_aluminio\(1\).pdf](http://alacermas.com/img/galeria/files/aluminio/chapa_6061_aluminio(1).pdf). p. 1-2.

Tabla 2. Propiedades del aluminio 6061.

Composición química										
%	Si	Fe	Cu	Mn	Mg	Cr	Zn	Ti	Otros elementos	Al
Mínimo	0.40		0,15		0,80	0,04				
Máximo	0.80	0,70	0,40	0,15	1,20	0,35	0,25	0,15	0.15	El resto

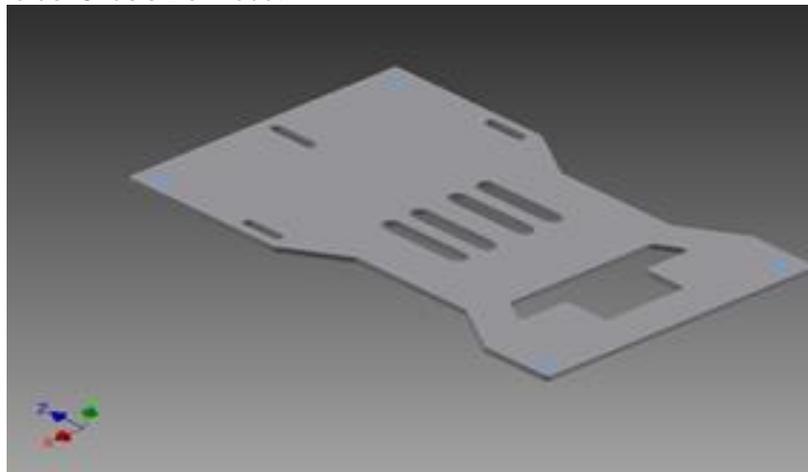
Fuente: <http://tecniaceros.com/pdfs/aluminio.pdf>, TecniAceros Aluminio 6061-T6 (AISI/ASTM).

Esta es una aleación dúctil, ligera, con gran resistencia y excelentes características de acabado; el aluminio 6061 es ideal para la elaboración de piezas maquinadas con calidad, de perfección para trabajos que requieran buena definición superficial.

“Posee mayor resistencia a la corrosión, además de facilidad de soldadura y una severidad parecida a la del acero. Es una aleación de propósito general muy popular con buena facilidad de maquinado a pesar de su tratamiento de envejecimiento artificial”.⁷

Una vez seleccionado el material de uso para el chasis, se procedió a realizar el modelado de la superficie mostrado en la Figura 5, se debe tener en cuenta la necesidad de la ubicación de un sistema de carga en la parte de adelante, y la ubicación de la batería como contrapeso en la parte de atrás. Para brindarle estabilidad al robot, la distancia desde la parte frontal hasta la parte trasera es de mayor dimensión a las laterales. El modelado fue realizado con el programa AUTOCAD “(es un software de diseño asistido por computadora utilizado para dibujo 2D y modelado 3D. Actualmente es desarrollado y comercializado por la empresa Autodesk)” a través de su herramienta AUTODESK.

Figura 5. Diseño del Chasis Del Robot.



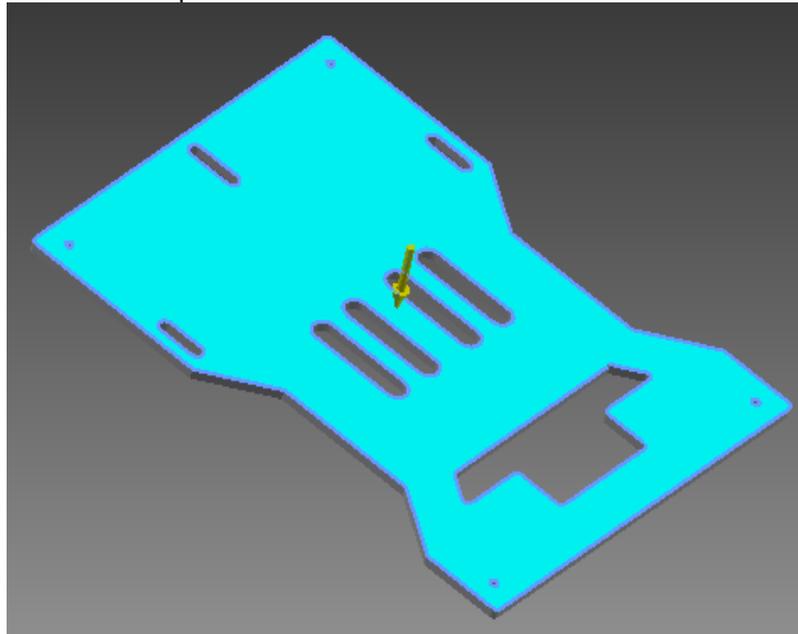
Fuente: Autores

⁷ Ficha técnica Aluminio 6061-T6 <http://www.tecniaceros.com/pdfs/aluminio.pdf>, TecniAceros Aluminio 6061-T6 (AISI/ASTM).

El diseño del chasis se realizó de forma rectangular dando sentido a la posición frontal y trasera, mediante orificios internos y recortes laterales, con el objetivo de poder pasar los cables de conexión y reducir el peso del robot. Las dimensiones del chasis son 22.8 centímetros (cm) de largo x 17 centímetros (cm) de ancho, la lámina tiene un espesor de 3 milímetros (mm). Sobre este chasis se monta la estructura completa del robot, a este van acoplados: los motores por medio de soportes de aluminio; un gabinete en acrílico que contendrá los componentes electrónicos, la batería y el sistema de levantamiento de montacarga.

El chasis es la base del sistema robotizado, por lo tanto, se requiere que cumpla con diferentes características mecánicas. Después de realizado el modelado y tener definida su forma, dimensiones y el material de elaboración, fue conveniente realizar pruebas de simulación mediante el programa de diseño en los puntos en donde la estructura pudiera presentar mayor debilidad, con el fin de determinar la capacidad del chasis para soportar la fuerza ejercida debido al levantamiento de una carga. Al seleccionar los puntos para esta simulación, el primero de ellos fue ubicado en el centro del chasis y el segundo en la parte frontal, donde ira soportado el sistema de levantamiento, mostrado en la Figura 6.

Figura 6. Primera simulación: punto central.

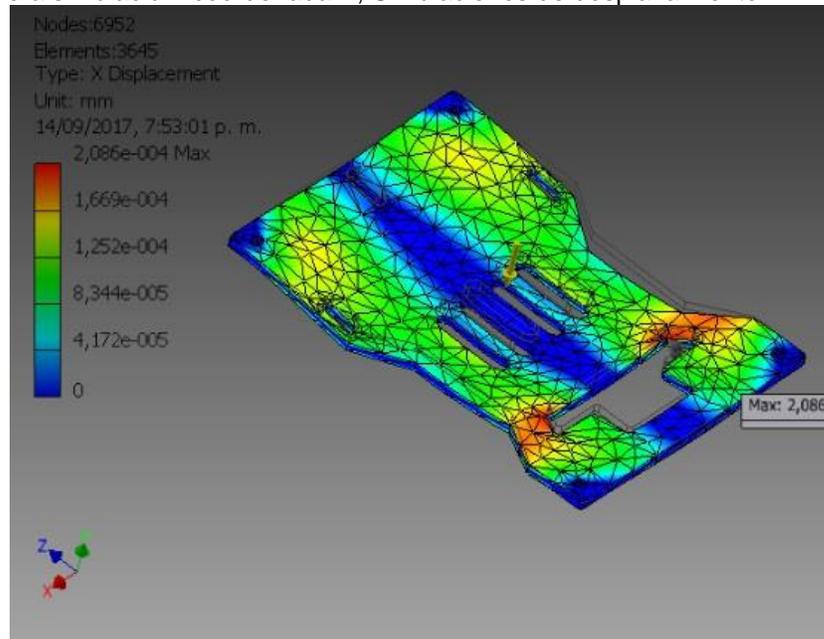


Fuente: Autores.

El puntero amarillo indica, el punto sobre el cual se va a ejercer una fuerza simulada equivalente a 20 Newton de fuerza. Se analizan los resultados de desplazamiento en las tres coordenadas.

Simulación en la coordenada X mostrado en la Figura 7.

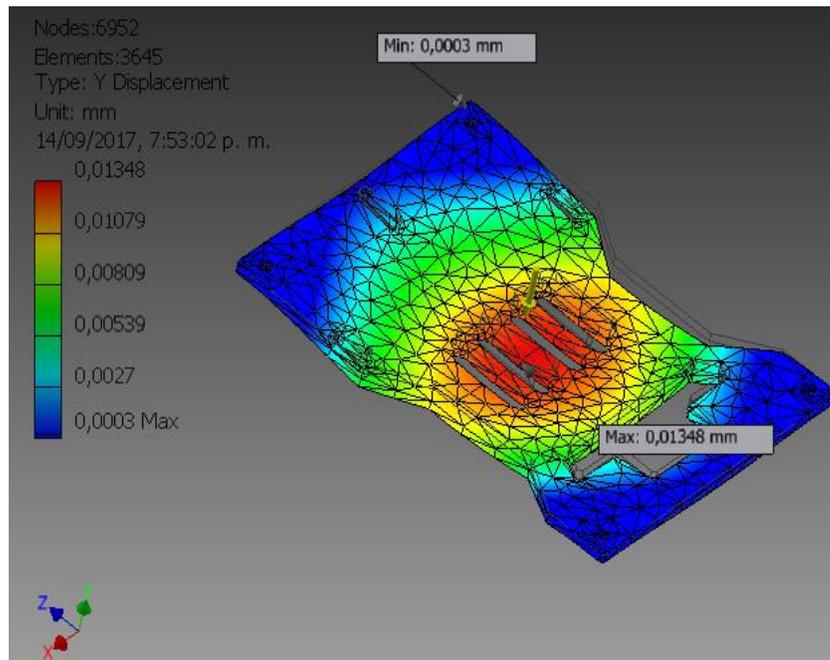
Figura 7. Primera simulación: coordenada X, Simulaciones de desplazamiento.



Fuente: Autores.

Simulación en la coordenada Y mostrado en la Figura 8.

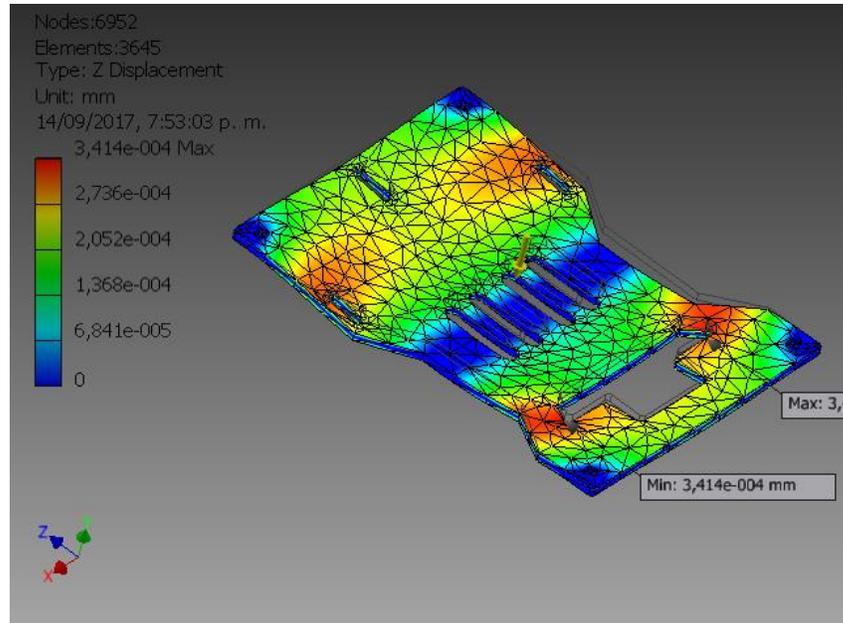
Figura 8. Primera simulación: coordenada Y, Simulaciones de desplazamiento



Fuente: Autores.

Simulación en la coordenada Z mostrado en la Figura 9.

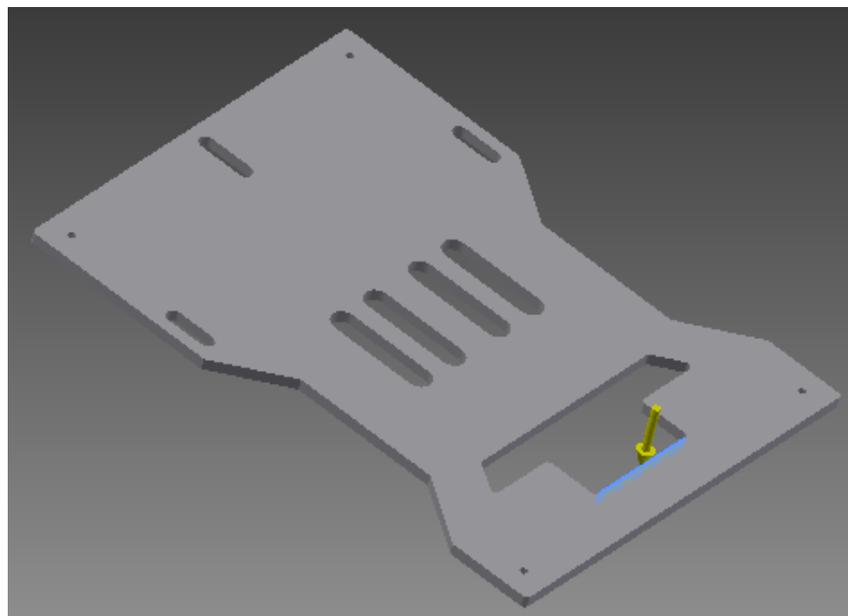
Figura 9. Primera simulación: coordenada Z, Simulaciones de desplazamiento



Fuente: Autores.

Se realizó la segunda simulación en la parte frontal del chasis mostrado en la Figura 10.

Figura 10. Segunda simulación: punto frontal

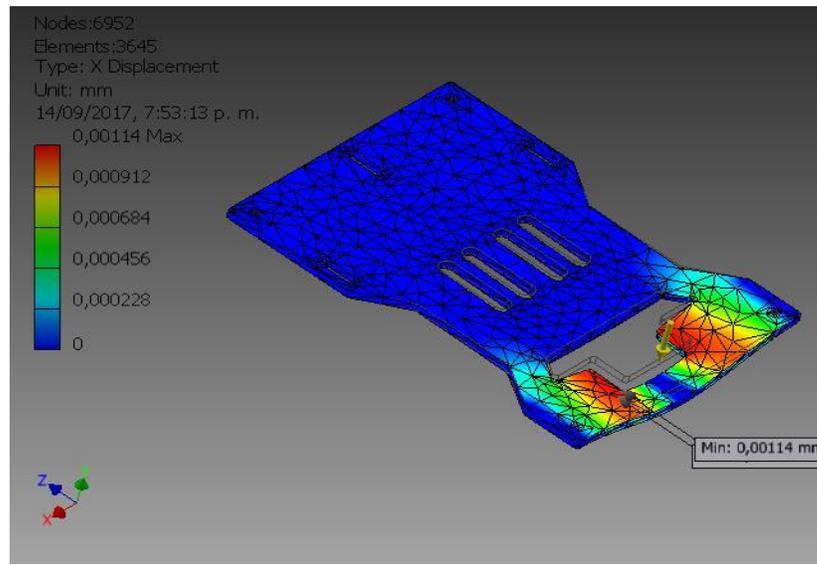


Fuente: Autores.

Igualmente se aplica una fuerza 20 Newton y se analizan los resultados de desplazamiento en las tres coordenadas.

Simulación en la coordenada X mostrado en la Figura 11.

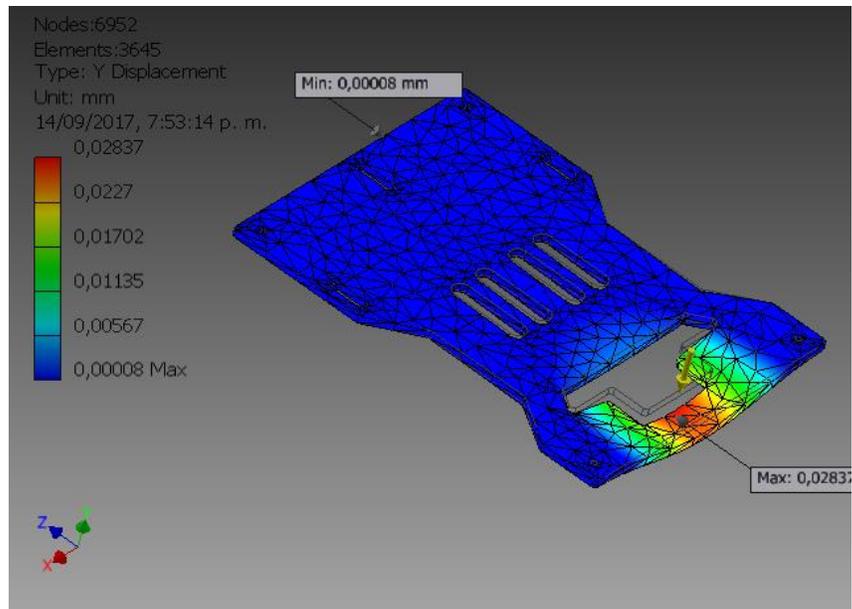
Figura 11. Simulaciones de desplazamiento en ejes X.



Fuente: Autores.

Simulación en la coordenada X mostrado en la Figura 12.

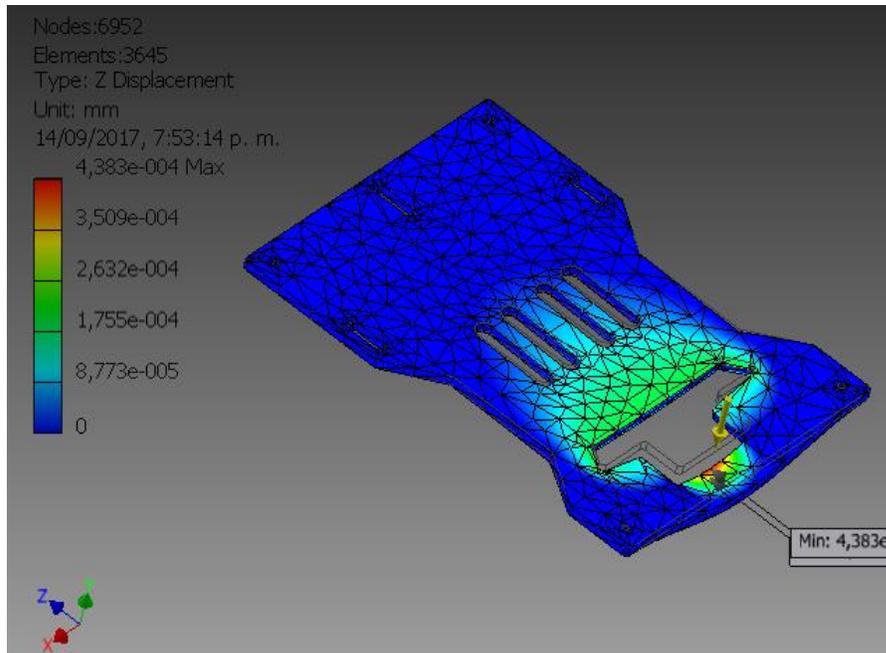
Figura 12. Simulaciones de desplazamiento en ejes Y.



Fuente: Autores.

Simulación en la coordenada Z mostrado en la Figura 13.

Figura 13. Simulaciones de desplazamiento en ejes Z.



Fuente: Autores.

De las simulaciones realizadas con el software *AutoCAD* se puede determinar: que la máxima deformación obtenida fue de 0.028 mm en el eje Y de la fuerza aplicada en la parte frontal, la cual resulta ser irrelevante para la seguridad de la estructura. Obtenido este resultado se procede a la creación de la pieza por medio de corte con láser.

3.2 SELECCIÓN DEL SISTEMA DE LOCOMOCIÓN

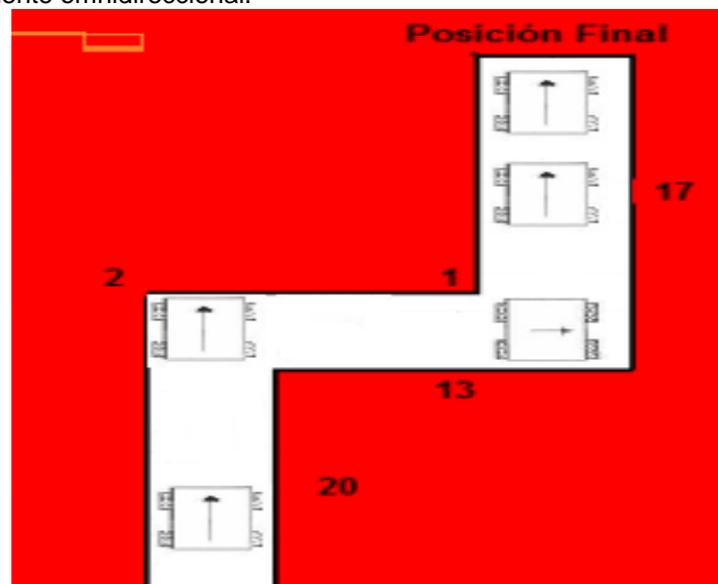
Para la selección apropiada del sistema de locomoción, se tuvo en cuenta la variedad de llantas disponibles en el mercado, encontrando diferentes características especiales en todos los modelos utilizables en este proyecto que cubren sus diversas necesidades.

En la aplicación del sistema de desplazamiento de carga, se hace necesario un sistema que cuente con flexibilidad de movimientos, y al mismo tiempo con características de un prototipo móvil de modelamiento. Se encontró que las ruedas convencionales son más económicas y facilitan su instalación, pero presentan limitaciones a la hora de realizar movimientos en ambientes reducidos; la programación de trayectorias resulta ser muy compleja en este tipo implementación.

Existen otros tipos de mecanismos llamados tipo oruga, los cuales permiten tener mayor versatilidad en diferentes terrenos y mayor tracción, pero son sistemas costosos y aún no poseen los grados de libertad necesarios para cumplir trayectorias en ambientes reducidos.

Una tercera opción son las ruedas omnidireccionales que ofrecen una versatilidad mayor; aunque su configuración es más compleja y su costo más elevado, brindan al robot una variedad de movimientos, que permiten cumplir trayectorias de diversas maneras mostrado en la Figura 14. La programación de los movimientos resulta ser compleja, pero los resultados y rendimientos son los esperados para un robot de transporte de carga.

Figura 14. Movimiento omnidireccional.



Fuente: Autores.

Los desplazamientos que deben realizar entre el punto de recolección de carga y la ubicación final de la misma, sugiere el uso de las llantas omnidireccionales mostrado en la Figura 15.

Las ruedas omnidireccionales Mecanum, cuentan con rodillos ubicados en la periferia de la rueda principal en una orientación de 45° respecto al eje de rotación de la misma, permitiendo que el prototipo pueda cambiar su dirección de movimiento sin cambiar su orientación⁸.

⁸ Improved mecanum Wheel desing for omni-directional robots, Olaf dieguel, Aparna Badve, Glen Bright, Johan Potgieter, Sylvester Tlale. p.56-58.

Figura 15. Llantas mecanum.

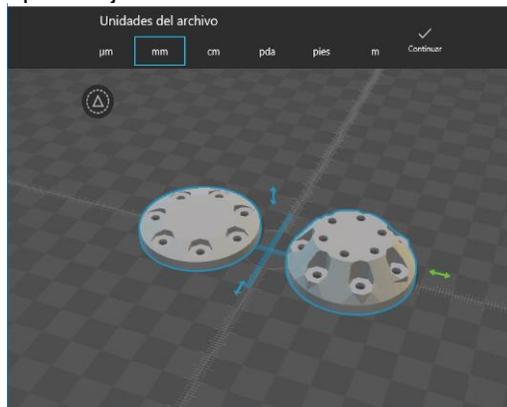


Fuente: tomado de la revista vexrobotics.com sesión: <http://vexrobotics.com/mecanum-wheels.html>

La instalación de las ruedas omnidireccionales a los motores no está preestablecida ni estandarizada, fue necesario el diseño y elaboración de piezas plásticas impresas en una máquina de prototipado 3D.

Las piezas en 3D corresponden a dos piezas de plástico, las cuales encajan en el buje de sujeción del motor DC, y sostienen cada llanta tipo omnidireccional aseguradas por medio de 7 tornillos con tuerca mostrado en la Figura 16.

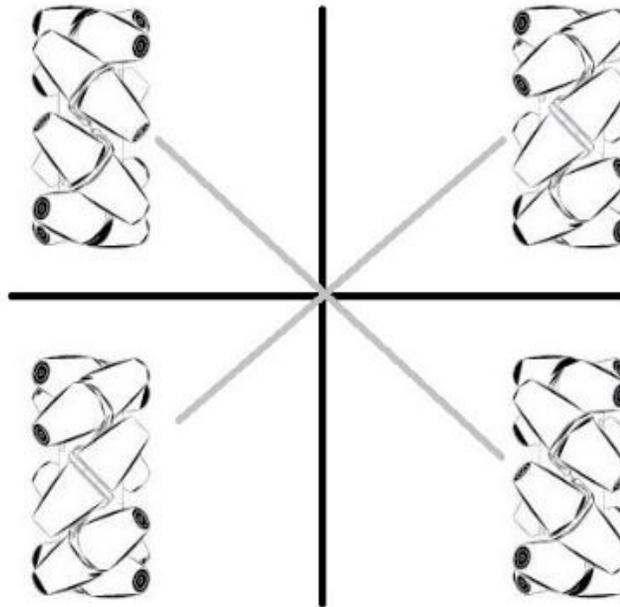
Figura 16. Acoples plásticos para sujeción de llantas.



Fuente: tomado de la revista vexrobotics.com sesión: <http://vexrobotics.com/mecanum-wheels.html>

Las cuatro llantas deben ubicarse de forma estratégica en cada uno de los motores, estas vienen en dos pares, un par con rodillos alineados con pendiente negativa, y el otro par con pendientes positivas, mostrado en la Figura 17. Las llantas deben ser ubicadas de tal forma que los rodillos de las cuatro llantas apunten al interior del robot, viéndolos desde una vista superior mostrado en la Figura 18.

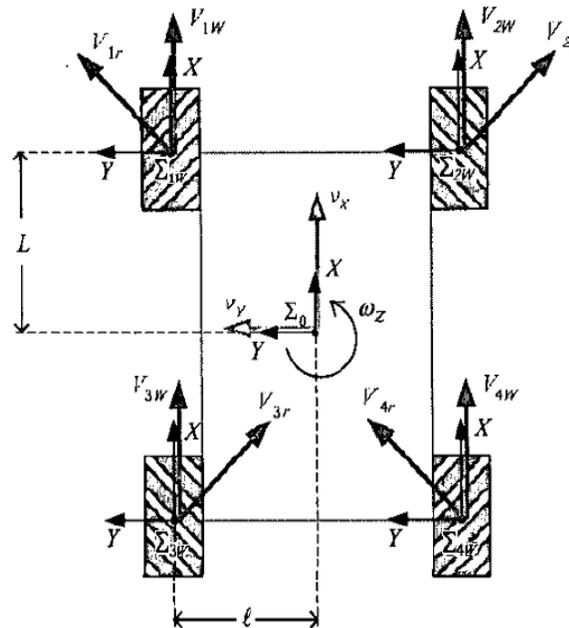
Figura 17. Vista superior de las ruedas



Fuente: Autores.

3.2.1 Movimientos del robot de carga

Figura 18. Disposición de ruedas Mecanum y sistema de coordenadas.



Fuente: proyecto de grado Control y Comportamiento de Robots Omnidireccionales, Santiago Martínez, Rafael Sito. p. 34-35.

“(Vix) Es el componente que aporta la velocidad lineal a la rueda (Viw) más la velocidad aportada en el sentido de los rodillos libres (Vir/cos 45°). Análogamente, (Viy) es (Vir/ sen 45°). Mostrado en la Figura 18.”⁹

“Mientras tanto, $v_x, v_y, w_z \in R$ representan los elementos respecto a los ejes X e Y y la velocidad angular del vehículo, respectivamente. Además, V_{ix}, V_{iy} se obtienen a partir de v_x, v_y, w_z ”.¹⁰

3.2.2 Torque de los motores

Para seleccionar el torque adecuado, se debe encontrar la respuesta en una serie de parámetros desconocidos, incluida la carga que puede tomar un motor específico, corrientes, voltajes, peso, estructura.

3.2.3 Determinación del momento de inercia (Jm) de los motores

Para determinar el momento de inercia se utiliza la ecuación del tiempo mecánico que se muestra en la ecuación 1.

Ecuación 1. Tiempo mecánico

$$t_m(s) = \frac{j_m \times R_a}{K_t \times K_e} = \frac{(kg-m^2, Ohmios)}{(Nm / A, V / rad -s)}$$

Fuente: tomado de,

https://repository.upb.edu.co/bitstream/handle/20.500.11912/504/digital_17633.pdf?sequence=1

Donde por el método paramétrico se obtiene:

- Resistencia de la armadura (Ra).
- Constante de torque (Kt).
- Constante de Voltaje (Ke).
- Constante de tiempo mecánico (tm)
- Momento de inercia (Jm).

De la Ecuación 1 se despeja el momento de Inercia (Jm) y se obtiene la Ecuación 2 que se a continuación

Ecuación 2. Momento de Inercia

$$J_m = \frac{(t_m)(k_t)(K_e)}{R_a}$$

Fuente: tomado de,

https://repository.upb.edu.co/bitstream/handle/20.500.11912/504/digital_17633.pdf?sequence=1

⁹ SANTIAGO MARTÍNEZ, RAFAEL SISTO, Control y Comportamiento de Robots Omnidireccionales, 5.3 Omnidireccional de 4 ruedas Mecanum Universidad de la República, Montevideo-Uruguay, 22 de noviembre del 2009.

¹⁰ SANTIAGO MARTÍNEZ, RAFAEL SISTO, Control y Comportamiento de Robots Omnidireccionales, 5.3 Omnidireccional de 4 ruedas Mecanum Universidad de la República, Montevideo-Uruguay, 22 de noviembre del 2009.

3.2.4 Corriente de arranque del motor y Torque de fricción

Torque de fricción: corresponde al par necesario para romper la fricción estática.

Corriente necesaria para que venza la inercia mecánica.

“Mediante una fuente DC variable, se incrementa el valor de la tensión gradualmente, observando atentamente el momento en que el eje del motor inicia su movimiento, cuando ello ocurre, se suspende este proceso y se toma la lectura del amperímetro correspondiente. Este será el valor de la corriente de arranque.

El torque de fricciones es igual a la constante de torque por la corriente de arranque”¹¹mostrado en la Ecuación 3.

Ecuación 3. Torque

$$Tf = Kt * Iarr$$
$$0.0879 \frac{Nm}{A} * 94.46mA = 8.30 * 10^{-3} Nm$$
$$Tf = 8.30 * 10^{-3} Nm \text{ (Torque)}$$

Fuente:https://repository.upb.edu.co/bitstream/handle/20.500.11912/504/digital_17633.pdf?sequence=1&isAllowed=y

El motor que se utiliza depende de la carga que el robot va a levantar.

La batería debe alimentar a los circuitos y motores, sin necesidad de que el robot se encuentre conectado al sistema de energía eléctrica. Se deben tener en cuenta las características, tamaño, peso, costo y cantidad de energía a suministrar por la batería.

El consumo de la corriente del microcontrolador está dado por 250 miliamperios y la cámara para el reconocimiento de imágenes 250 miliamperios.

Se seleccionarán motores que trabajen a 12 V y cuya corriente de operación normal no supere los 600 mA, para el sistema de locomoción y el sistema de carga.

Por lo tanto, se tiene que para cuatro motores de locomoción y uno de carga, un consumo máximo de 3000 mA en operación normal.

Se obtiene:

Ecuación 4. Corrientes

$$I_T = I_m + I_c + I_{Mo} + I_{Mt}$$
$$I_T = 250mA + 250mA + 600mA + 3000mA$$

¹¹ tomado de: obtención experimental de los parámetros del motor que se utilizará en el sistema de locomoción de una esfera rodante

$$I_T = 4100mA = 4.1A$$

Donde,

$$\begin{aligned} I_T &= \text{Corriente del circuito} \\ I_m &= \text{Corriente del microcontrolador} \\ I_c &= \text{Corriente de la cámara} \\ I_{Mo} &= \text{Corriente del Motor} \\ I_{Mt} &= \text{Corriente del Motor de translación} \end{aligned}$$

Potencia de los motores:

Se determina la potencia de cada motor de la siguiente manera:

Ecuación 5. Potencia

$$\begin{aligned} P &= V * I \\ V &= \text{equivale a la tensión del motor } 12V \\ I_{Mo} &= \text{Corriente del Motor} \\ P &= 12V * 600mA = 7.2W \end{aligned}$$

3.2.5 Motores del Sistema de Locomoción

Para el sistema de locomoción del prototipo del robot de carga, es indispensable tener en cuenta que el prototipo se debe desplazar con una carga de 2.5 Kg, adicionalmente la carga que representa el chasis del prototipo, el sistema de levantamiento y la estructura de acrílico diseñada para la protección del circuito electrónico, para ello debe tener unos motores que tengan la fuerza de 8 kg*cm para romper el momento de inercia. Por tal motivo, se trabajó con 4 motores con reductores de escala 131:1 (Ver Figura 19), los cuales teniendo las configuraciones adecuadas pueden alcanzar un torque de 18 Kg*cm, lo suficiente para poder desplazar la estructura del prototipo con o sin carga.

Figura 19. Motor reductor metálico 131:1



Fuente: tomado de, robotica-dec.com. Motorreductor metálico 131:1 (37D x 57L mm)

Cada motor cuenta con un encoder de cuadratura, que provee 64 pulsos por vuelta en el eje del motor y equivale a 8384 pulsos por vuelta después de la caja reductora.

Figura 20. Torque

Gear Ratio	No-Load Speed @ 12 V	Stall Torque		Stall Current	
		@ 12 V	@ 12 V	@ 12 V	@ 12 V
1:1	11,000 RPM	5 oz-in	5 A	With Encoder motor without gearbox	Without Encoder
19:1	500 RPM	84 oz-in	5 A	37Dx52L mm	37Dx52L mm
30:1	350 RPM	110 oz-in	5 A	37Dx52L mm	37Dx52L mm
50:1	200 RPM	170 oz-in	5 A	37Dx54L mm	37Dx54L mm
70:1	150 RPM	200 oz-in	5 A	37Dx54L mm	37Dx54L mm
100:1	100 RPM	220 oz-in	5 A	37Dx57L mm	37Dx57L mm
131:1	80 RPM	250 oz-in	5 A	37Dx57L mm	37Dx57L mm

Características:

- Velocidad de 80 RPM a 12 V
- Consumo de corriente 300 mA sin carga
- Torque 250 oz-in (18 kg-cm)

Fuente: <http://teslabem.com/motorreductor-pololu-18kg-encoder.html>

3.3 SELECCIÓN DEL SISTEMA DE LEVANTAMIENTO DE LA CARGA

El mecanismo de sujeción y recolección de la carga fue diseñado y construido después de analizar varios sistemas entre los cuales se encontraban: pinza, brazo, actuador con cilindro hidráulico, tornillo sin fin, entre otros. Se optó por un movimiento con motor eléctrico, usando poleas y una correa de distribución que convierte el movimiento circular del motor, en un movimiento lineal de las palas del prototipo por medio de unas guías verticales de plástico, este diseño es similar al fabricado en montacargas convencionales, mostrado en la Figura 21.

Figura 21. Sistema Montacargas convencional con motor eléctrico.



Fuente: tomado de, staubli.com, robots de carga serie SP.

Este sistema resulto fácil de acoplar y presento buenos rendimientos en las pruebas experimentales. El sistema de levantamiento cuenta con dos pulsadores amortiguados que funcionan como finales de carrera para activar y desactivar los movimientos de levante y descarga.

Las palas o uñas del montacarga se diseñaron en aluminio para darle rigidez y un peso liviano, cuenta con una longitud de 7.5 cm de largo, cada pala tiene un ancho de 1.1 cm y están separadas a una distancia de 2.5 cm, mostrado en la Figura 22.

Figura 22. Sistema Montacargas prototipo Robot de carga.



Fuente: Autores

Para facilitar el trabajo del levantamiento, fue necesario realizar un soporte proporcional al tamaño de las palas. Este diseño se realizó en base a una estiba donde se pondrá el producto a transportar, mostrado en la Figura 23. Las dimensiones de la estiba a escala son: 14.5 x 9.7 x 4.6 (ancho x largo x alto), mostrado en la Figura 24.

Figura 23. Estiba para productos.



Fuente: tomado de, maderplast.com, estibas, pallets y contenedores.

Figura 24. Estiba para productos.



Fuente: Autores

- **Motor del Sistema de Levantamiento**

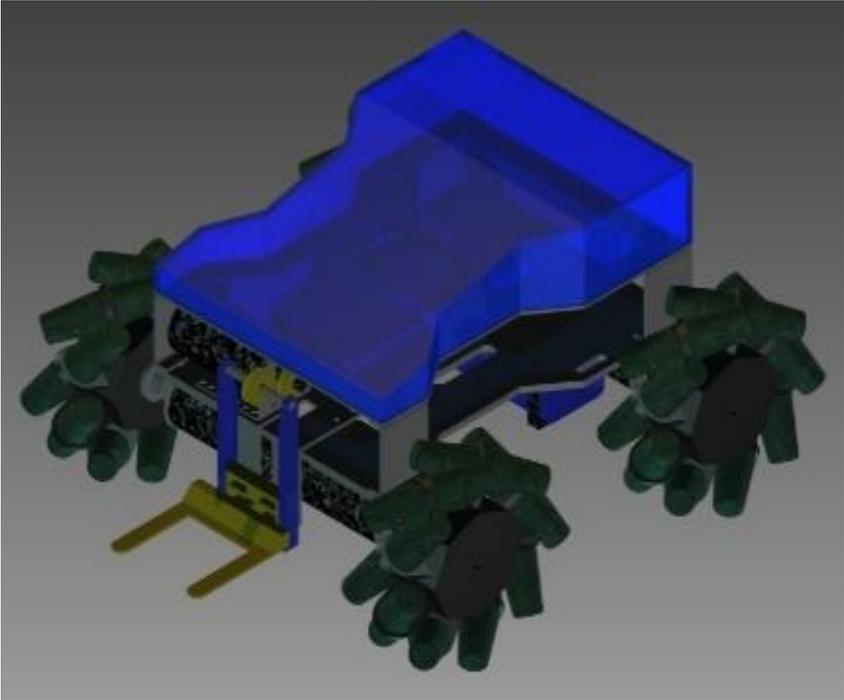
Para el sistema de levantamiento del prototipo del robot de carga, es necesario tener en cuenta, que el prototipo debe levantar desde el piso una carga de hasta 2.5 kg, para esto se necesita un motor que tenga una gran fuerza, por tal motivo se trabajó con un motor con reductor de escala 131:1, mostrado en la Figura 19, este motor se puede configurar para que tenga un torque de 18 Kg*cm siempre y cuando tenga las condiciones adecuadas.

Este motor, así como los motores del sistema de locomoción, también cuentan con un encoder de cuadratura que provee 64 pulsos por vuelta en el eje del motor y equivale a 8384 pulsos por vuelta.

3.4 ARMADO DEL SISTEMA MECÁNICO

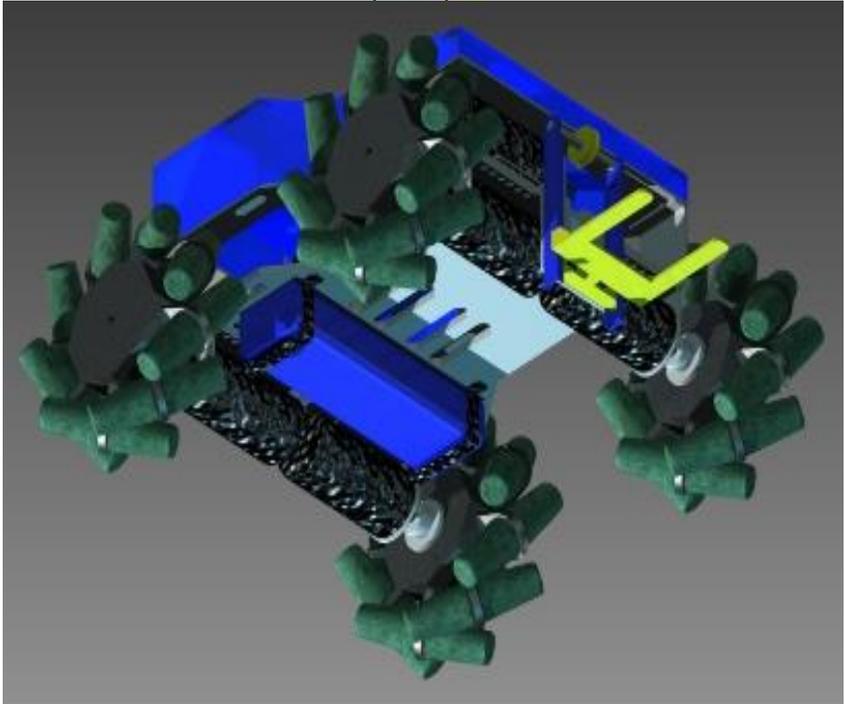
Una vez definida la locomoción y el sistema de levantamiento del prototipo, se procedió a realizar la integración de todas las partes y elementos que componen al prototipo de montacarga, mostrado en la Figura 25. Se adiciono una base en acrílico para poder instalar y proteger la tarjeta del circuito electrónico y se dejó un espacio para poder sujetar la Batería, mostrado en la Figura 26.

Figura 25. Diseño del sistema mecánico completo.



Fuente: autores.

Figura 26. Diseño del sistema mecánico completo, protección.

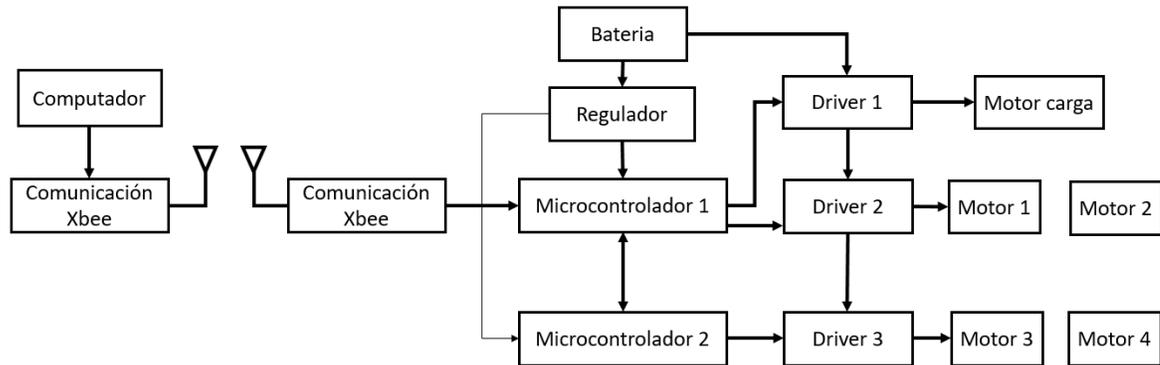


Fuente: autores.

3.5 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA ELECTRÓNICO DEL PROTOTIPO

En la Figura 27 se muestra el diagrama de bloques del sistema electrónico.

Figura 27. Diagrama de Bloques Sistema Electrónico

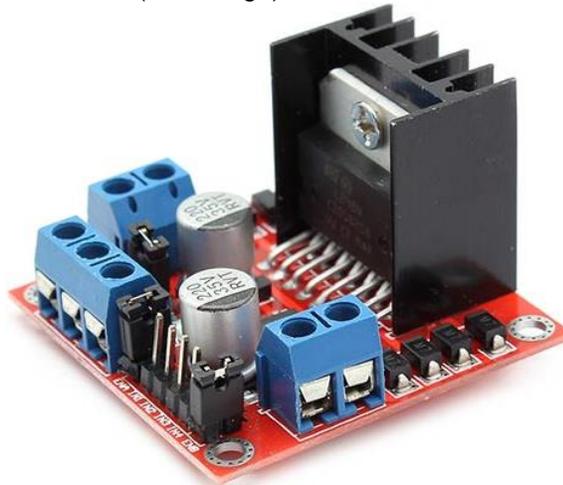


Fuente: Autores

Definido el sistema de locomoción, la estructura que hace parte del chasis y el mecanismo de sujeción de la carga, se procedió a diseñar el sistema electrónico del prototipo, teniendo en cuenta que se deben manejar cinco motores, cuatro del sistema de translación y uno del sistema monta carga. Se necesita un driver que sea capaz de manejar 750 miliamperios por motor.

Se utilizó el driver dual L298n, mostrado en la Figura 28. Este driver permite actuar dos motores DC de hasta 2 amperios, siendo necesario utilizar tres drivers para actuar los 5 motores.

Figura 28. Driver dual para motores (Full Bridge) L298n.



Fuente: electronilab.co, circuito de implementación Driver dual para motores.

Para efectos del proyecto, era necesario utilizar una batería de 12 V, que permitiera energizar los drivers seleccionados previamente para el funcionamiento de los motores del sistema de locomoción y el sistema de carga, pero además, capaz de suministrar los miliamperios necesarios para romper el punto de inercia de los motores que en su conjunto sería un acumulativo de cuatro unidades. Teniendo en cuenta lo anterior, la sumatoria de las corrientes de todos los elementos que componen el prototipo que da un total de 4100 mAh y las características proporcionadas por el fabricante de los motores, donde la corriente pico es bastante alta, se escogió una batería Lipo de tres celdas de 5100 mAh y 11.1 Voltios mostrado en la Figura 29. Cada celda de la batería es de 3.7 Voltios, las cuales pueden ser cargadas hasta un valor nominal de 4.2 Voltios, y presentan un nivel crítico de voltaje de 3.3 Voltios. Este rango no debe ser superado para evitar daños internos de las celdas y un mal funcionamiento de la batería. El proceso de carga de la batería se le conoce como balanceo.

Figura 29. Batería Lipo 3s1p 5100 mAh 11.1 V.



Fuente: tomado de, rc-innovations.es, 3DR Iris+ 3S 5100mAh 11.1V RC LiPo Drone Battery w/ XT60 Plug x4 packs by Venom.

Por precaución, para conocer la carga de la batería y evitar daños internos en la misma, se incluye un indicador de voltaje de ocho celdas y se configura para las tres celdas que posee la batería. Este indicador muestra el voltaje de cada celda y el voltaje total de la batería mostrada en la Figura 30.

Figura 30. Indicador de voltaje de 8 celdas.

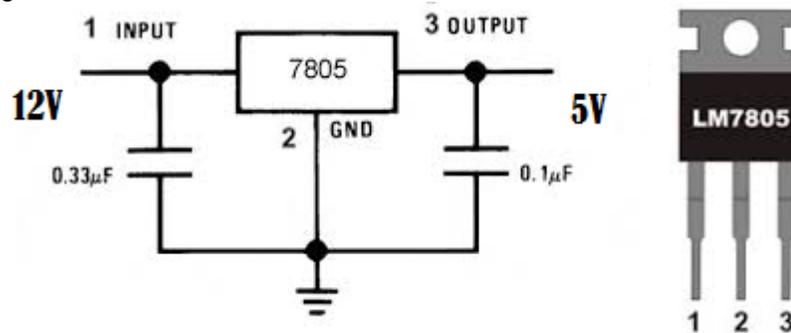


Fuente: tomado de, asturmodel.es Indicador Batería 2-8 celdas + avisador programable.

3.5.1 Regulador de voltaje de 12 Voltios a 5 Voltios

Un regulador de voltaje es un dispositivo electrónico idóneo para entregar una cantidad de voltaje constante a una carga, en un rango de funcionamiento o en un rango de trabajo. En el prototipo del robot de carga se utilizó el regulador LM7805, mostrado en la Figura 31, que de 12 Voltios entrega 5 Voltios esto para energizar aquellos elementos como los microcontroladores y los módulos *XBEE*.

Figura 31. Regulador LM 7805.



Fuente: Tomado de, electronics.stackexchange.com/q/293287 y modificado por Autores

3.5.2 Selección del microcontrolador

Se decide utilizar para el proyecto el Microcontrolador 16F877A (ver Figura 32), el modelo encapsulado de 40 pines, que requiere de una alimentación de 5 voltios, un oscilador de 20 Mhz y un terminal MCLR que este en alto durante la ejecución del programa porque si pasa a 0 V provoca un reset.

Figura 32. Microcontrolador Pic 16f877a.



Fuente: Tomado de, hacktronics.co.in/pic-microcontroller-ic/pic18f452-40-pin-dip-pic-microcontroller

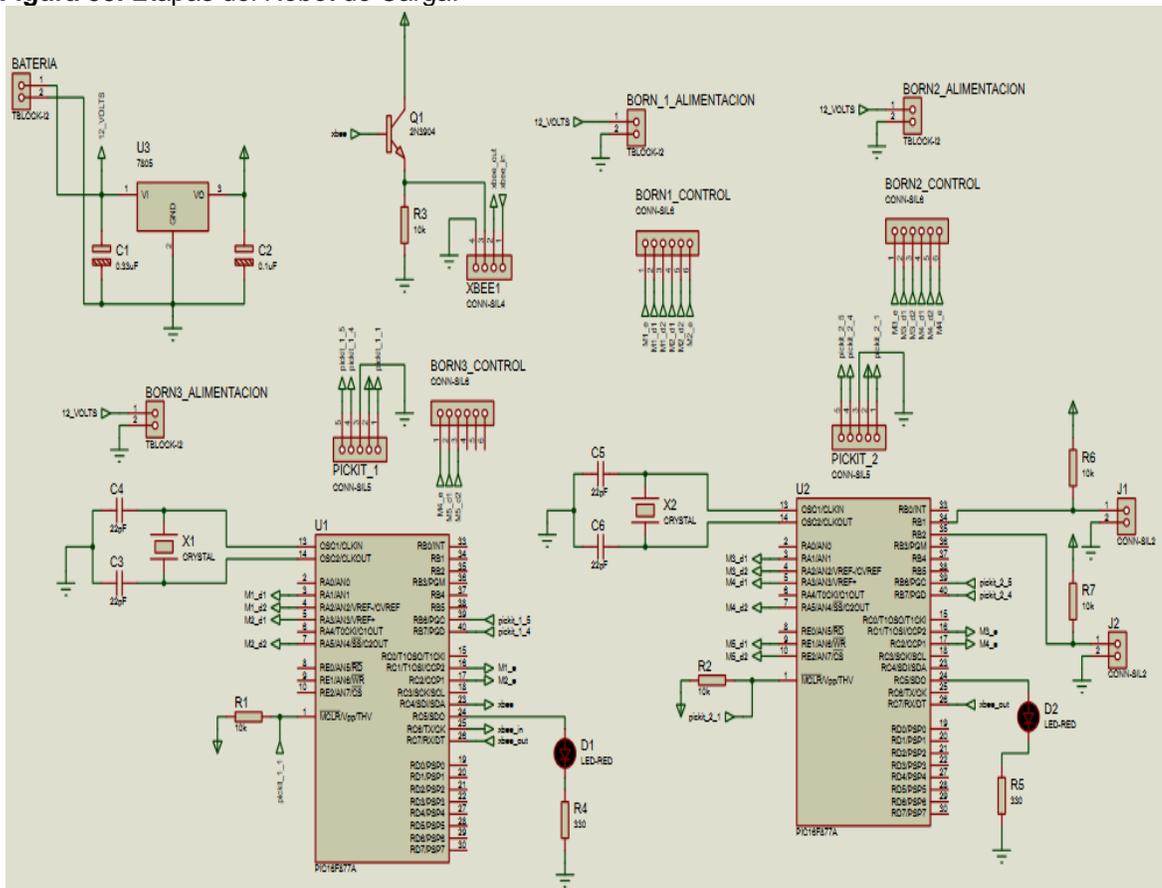
En el prototipo creado se requiere controlar un total de cinco motores, correspondientes al motor de cada rueda y al motor que se utiliza para el sistema de carga. Además, debe realizar diversas acciones como el procesamiento de datos digitales, el control de dispositivos y una comunicación entre varios de ellos. De ahí que haya sido necesario emplear dos microcontroladores PIC 16F877A; cada uno

de ellos posee dos salidas PWM que permiten controlar dos motores, que sumados permiten el control de los motores de las cuatro ruedas; para el quinto motor del sistema de carga se utiliza una de las cuatro salidas PWM ya que el motor montacarga funciona y se activa solo cuando las ruedas se encuentran inactivas.

3.6 DISEÑO E IMPLEMENTACIÓN PLACA BASE

Realizadas las pruebas de comunicación y obtenidos los niveles de alimentación de los microcontroladores, se procedió a el diseño y elaboración de la tarjeta electrónica del prototipo (placa base), utilizando el *software Proteus Isis* el cual permite la ubicación de los componentes y realizar simulaciones, mostrado en la Figura 33.

Figura 33. Etapas del Robot de Carga.

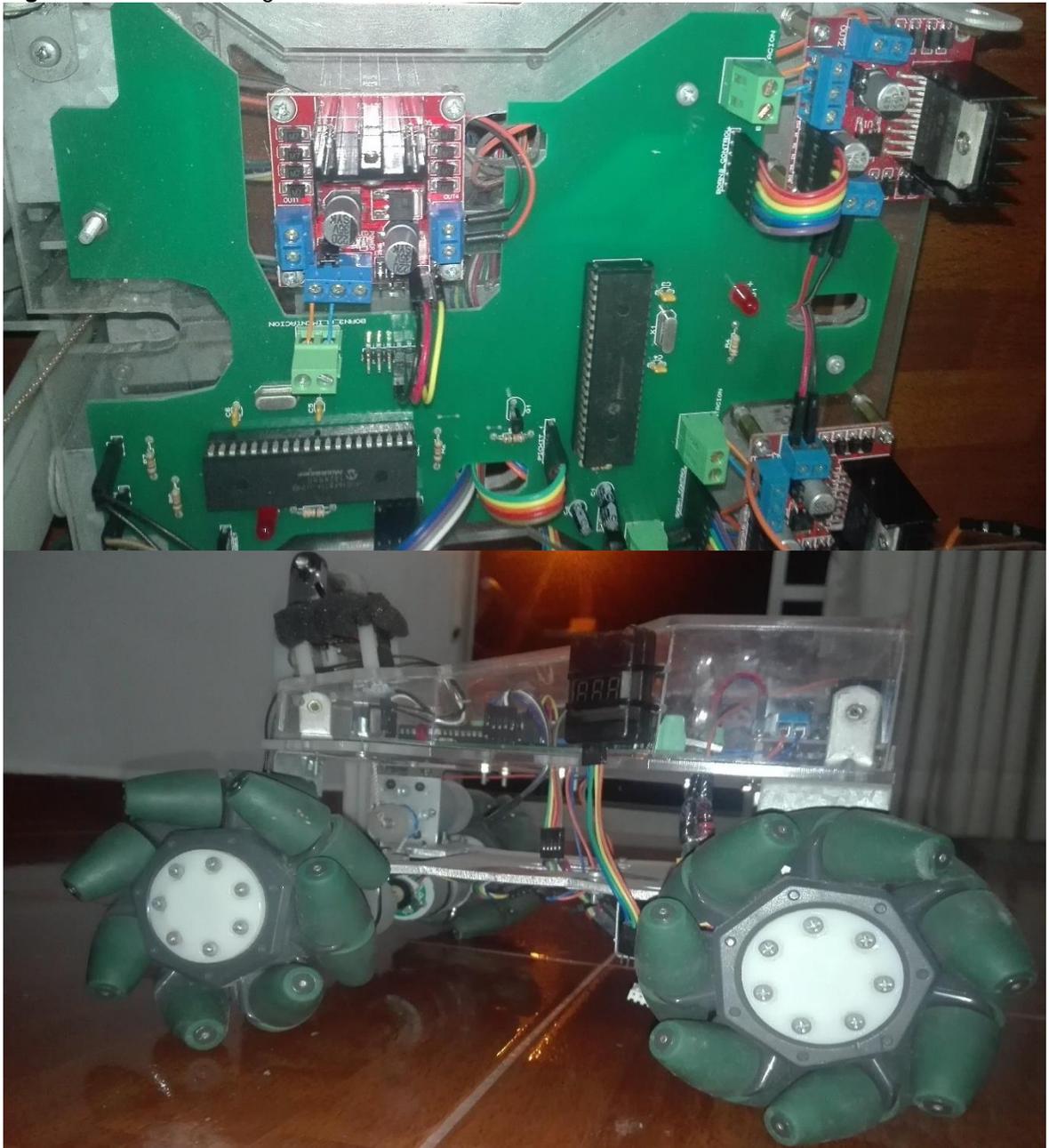


Fuente: Autores

Después de haber realizado el diseño, se implementó el circuito en protoboard para las respectivas pruebas, cuando las pruebas realizadas del circuito en la protoboard

dieron un resultado favorable, se procedió a montar en la baquela, esta será la placa base del prototipo robot de carga mostrado en la Figura 34.

Figura 34. Robot de Carga.



Fuente: autores.

- **Diseño de la programación para el Robot de Carga**

Como se mencionó, la programación utilizada en el microcontrolador es estructurada, es decir va en secuencia, permitiendo ejecutar las funciones establecidas.

Las etapas de la programación son:

- Variables de la recepción serial: datos que reciben para la conexión.
- Variables de transmisión serial: datos que transmitir hacia el circuito.
- Variables de dirección: corresponden a la posición donde se va a dirigir el robot de carga.
- Variables de tiempo: tiempo de ejecución del sistema.
- Dato recibido por la comunicación serial: al obtener los datos se tiene un ciclo que permite la ejecución del movimiento del sistema.
- Interrupción de tiempo: si se necesita detener el robot de carga, está habilitado para ejecutar la detención de este.
- Decodificación: permite ejecutar un proceso para obtener resultados de respuesta del sistema.
- Codificación: realiza el proceso inverso, es decir recibiendo un valor asignado un parámetro de diferente secuencia.
- Módulo *XBEE*.
- PWM: modulación por ancho de pulsos para modificación del ciclo de trabajo.

4. TRATAMIENTO DIGITAL DE LA IMAGEN

El procesamiento digital de la imagen se realizó en cuatro etapas que se describen paso a paso a continuación:

Figura 35. Diagrama de Bloques Tratamiento digital de la Imagen.



Fuente: Autores

4.1 PRE-PROCESAMIENTO

Etapa donde la imagen se mejora para su debido procesamiento, con el objetivo de facilitar el reconocimiento de las características que se desean ubicar en la imagen, para este caso los patrones; los cuales son los puntos de referencia que ubicaran el robot, la carga y el punto de descarga dentro de la imagen.

En el Pre-procesamiento se puede observar tres etapas que se mencionan y describen seguidamente:

Figura 36. Diagrama de Bloques Pre-Procesamiento digital de la Imagen.



Fuente: Autores

4.1.1 Imagen Original

La imagen se obtiene de la cámara posicionada sobre el escenario, visualizando los patrones de reconocimiento (robot, punto final y carga); esta imagen será procesada para encontrar las coordenadas de posición en la imagen. La imagen a color tiene tres componentes RGB, Mostrada en la Figura 37, con la cual una imagen se forma de tres capas de color y al combinarla se obtiene el color real. A continuación, se explica el procedimiento de detección del patrón del robot que constantemente es monitoreado.

Figura 37. Imagen RGB.

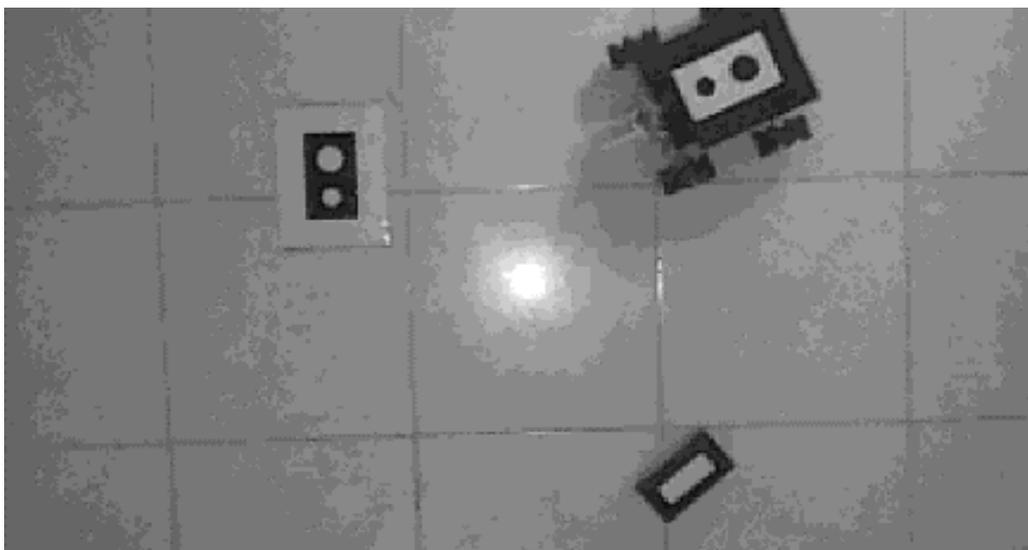


Fuente: autores.

4.1.2 Imagen en escala de grises

Se obtiene una imagen en escala de grises para trabajar con una sola capa de color que combina las tres capas RGB, se elimina el componente del tono y la saturación, cada pixel de la imagen se describe con un valor de 0 a 255 representando la escala de grises que existe del blanco al negro, mostrada en la Figura 38.

Figura 38. Imagen en escala de grises.

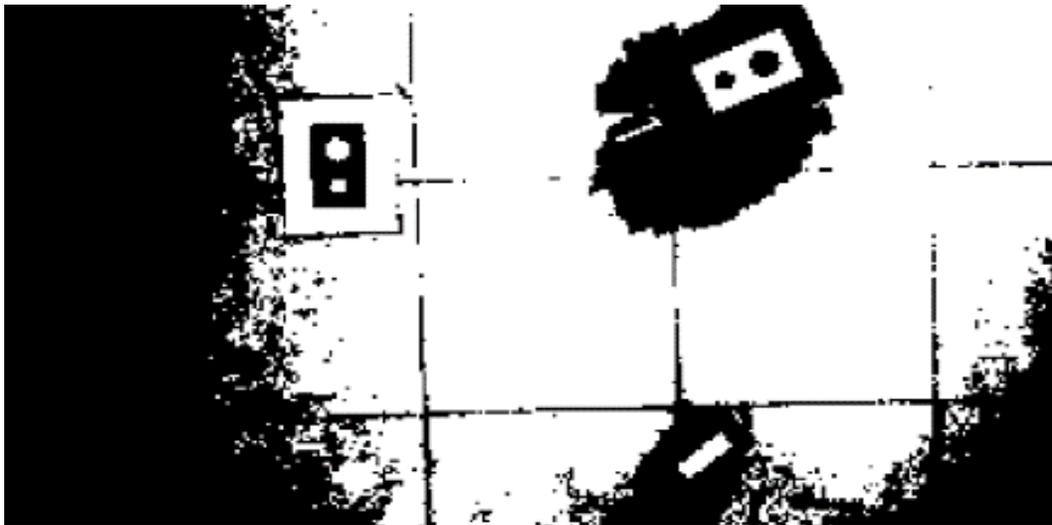


Fuente: autores.

4.1.3 Imagen en blanco y negro (umbral manual)

Con ayuda de un parámetro manual seleccionable en la interfaz gráfica, se obtiene la imagen a blanco y negro; este parámetro se denomina umbral, el cual es un número que varía de 0 a 255 que indica la referencia de cada pixel para convertirse en blanco o negro. Un pixel con un valor superior al valor del umbral será convertido en blanco, y un pixel con un valor inferior al valor del umbral será convertido en negro, mostrada en la Figura 39.

Figura 39. Imagen a blanco y negro.



Fuente: autores.

4.2 ANALISIS DE CONTORNOS

Se realiza un análisis de contornos, mostrado en la Figura 40, a partir de la imagen a blanco y negro para filtrar la información y poder detectar los patrones de reconocimiento de una forma ágil y robusta. Los contornos de cada imagen se definen como el conjunto de pixeles que separados por una unidad abarcan un área uniforme de la imagen ya sea blanca o negra.

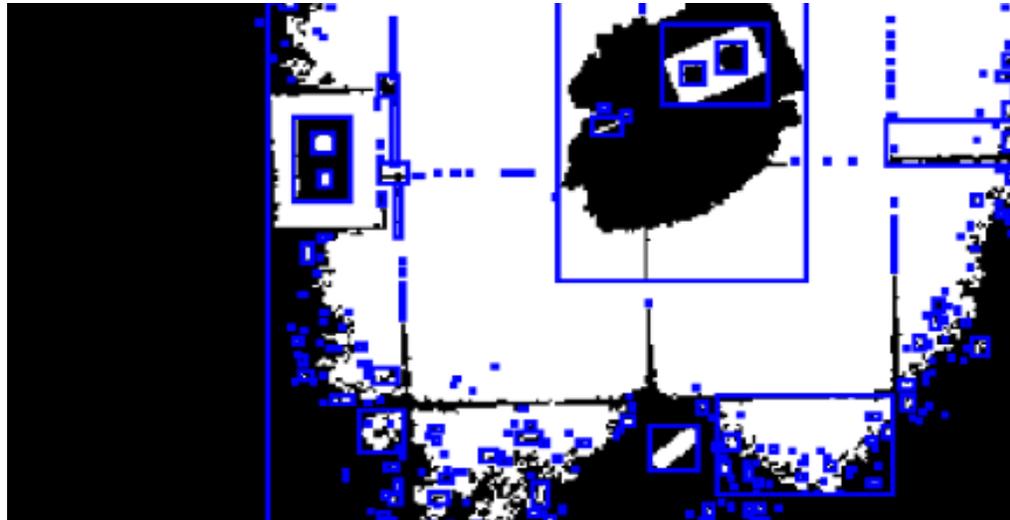
4.2.1 Extracción de Características

El algoritmo de reconocimiento de objetos normalmente extrae características y los compara con una base de datos que en este caso serían las medidas de los patrones para obtener una coincidencia, se pueden utilizar técnicas de visión por computador para realizar la extracción y codificar la información. Algunos métodos pueden ser:

- a. Extracción de características.
- b. Histograma de gradientes.
- c. Regla de Hebb.

El Algoritmo funciona con coeficientes de correlación, buscando hallar las regiones encerradas, retornando coordenadas donde están los componentes, realizando un procesado posterior, pasando la imagen a blanco y negro resaltando los bordes. Teniendo las regiones se halla los componentes conexos y se calcula una matriz de correlación, emparejando un *Template* devolviendo una matriz con picos y llegando a un coeficiente se obtienen las variables.

Figura 40. Análisis de contorno.



Fuente: autores.

4.3 DETECCION DEL PATRON RECTANGULAR

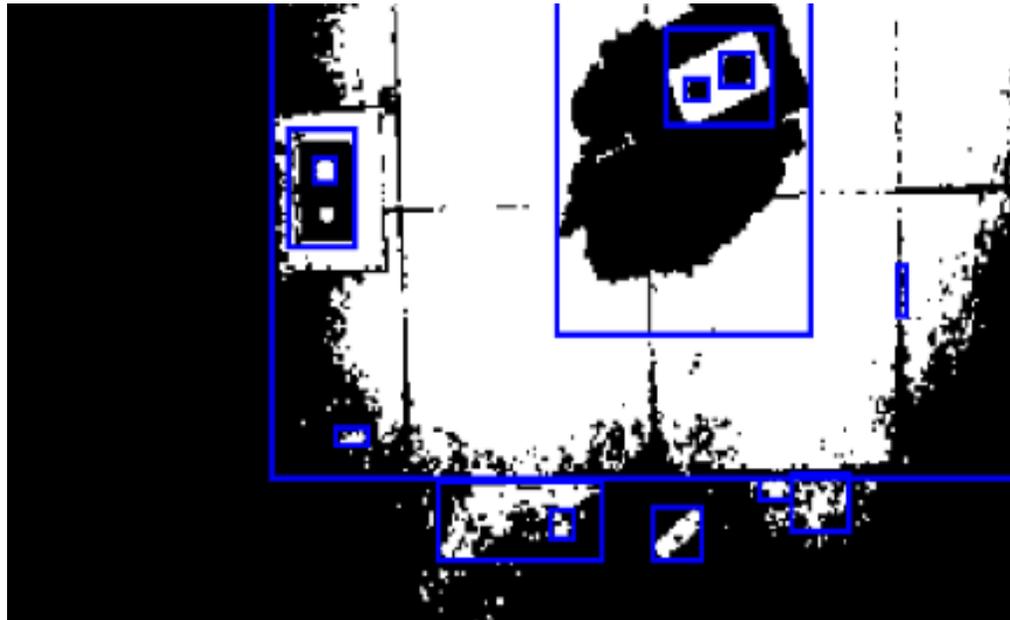
Una vez aplicado el análisis de contorno mediante la extracción de características, se aplican diversos filtros, que irán descartando los contornos, mostrada en la Figura 42, que no cumplen con las condiciones necesarias para la detección, en el diagrama de bloques se nombran todos los filtros que se aplican mostrado en la Figura 41.

Figura 41. Diagrama de Bloques de la detección del patrón rectangular.



Fuente: autores.

Figura 42. Descarte de contornos.

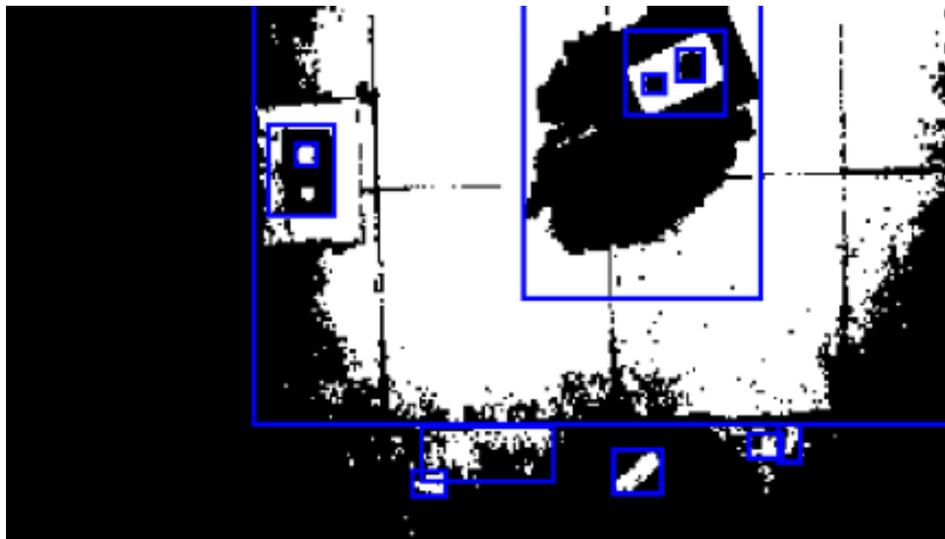


Fuente: autores.

4.3.1 Filtro por área (descartar contornos muy pequeños)

Inicialmente se realiza un análisis de filtrado, mostrado en la Figura 43, para descartar los contornos que no superan un área mínima determinada, que comúnmente es ruido en la imagen. Se obtiene una región con una matriz correlacionada para disminuir y determinar la variable.

Figura 43. Filtro por área.



Fuente: autores.

4.3.2 Filtro por área (descartar contornos muy grandes)

De igual forma se realiza un análisis de filtrado mostrado en la Figura 44, para los contornos que poseen un área muy grande en la imagen, estos contornos son descartados.

Figura 44. Descarte de contornos muy grandes.



Fuente: autores.

4.3.3 Filtro por relación de lados

Para el análisis de los contornos se realiza un cálculo de la relación que existe entre el lado más largo y el más corto, como el patrón buscado es un rectángulo, se obtiene que el valor aproximado es de: $\text{lado mayor/lado menor} = 1.6$ de variación más o menos de 0.3, se filtran los contornos que se encuentran por fuera del rango de aceptación, se ajusta una incertidumbre de variación 0.3 debido a la variabilidad de las imágenes obtenidas mostrada en la Figura 45.

Figura 45. Filtro por relación de lados



Fuente: autores.

4.3.4 Filtro por área en centímetros cuadrados (cm²) aproximado

Se realiza luego un análisis de filtrado, tomando el área de cada contorno que aproximada a un área real en unidades de cm² por medio de un factor de conversión mostrado en la Figura 46, se determina un criterio de aceptación de 170 una variación de más o menos 70 y una variabilidad mayor debido al factor de conversión que es aproximado.

Figura 46. Filtro por cm².

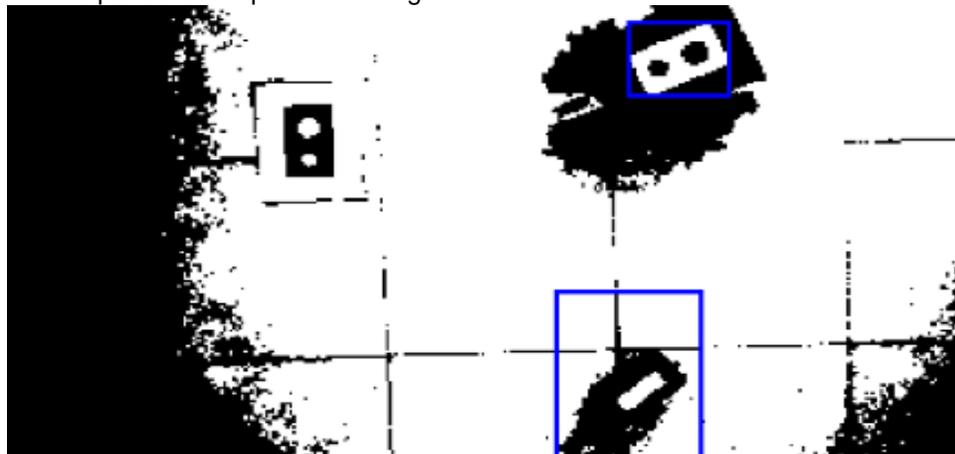


Fuente: autores.

4.3.5 Filtro por color del patrón rectangular (centro)

Se realiza un análisis de los contornos filtrados mostrado en la Figura 47, hasta ese momento y se determinan que regiones son de color negro y blanco, para descartar los que no corresponden con el color del patrón de reconocimiento.

Figura 47. Filtro por color del patrón rectangular.



Fuente: autores.

4.4 DETECCION DE PATRONES CIRCULARES

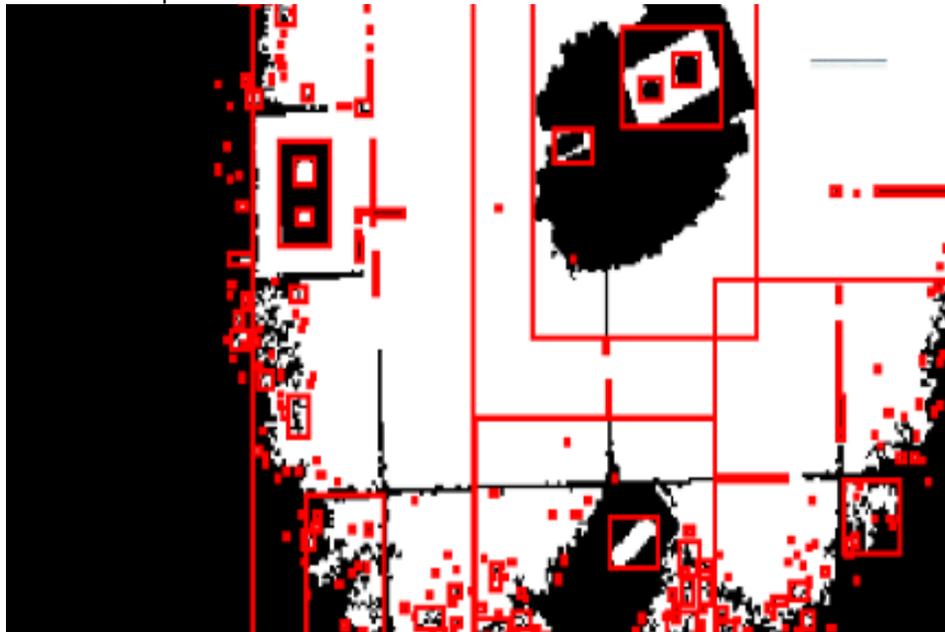
Continuando con el procesamiento de la imagen con el fin de detectar el patrón de reconocimiento del robot, se analizan ahora los contornos internos de cada una de las proximidades rectangulares encontrados en el anterior procedimiento. Se toman nuevamente los contornos de la figura y se empiezan a examinar por análisis de filtrado, los filtros y el proceso de pueden observar a continuación mostrados en las Figuras 48 y 49.

Figura 48. Diagrama de Bloques Detección de Patrones Circulares.



Fuente: autores.

Figura 49. Análisis de patrones internos.

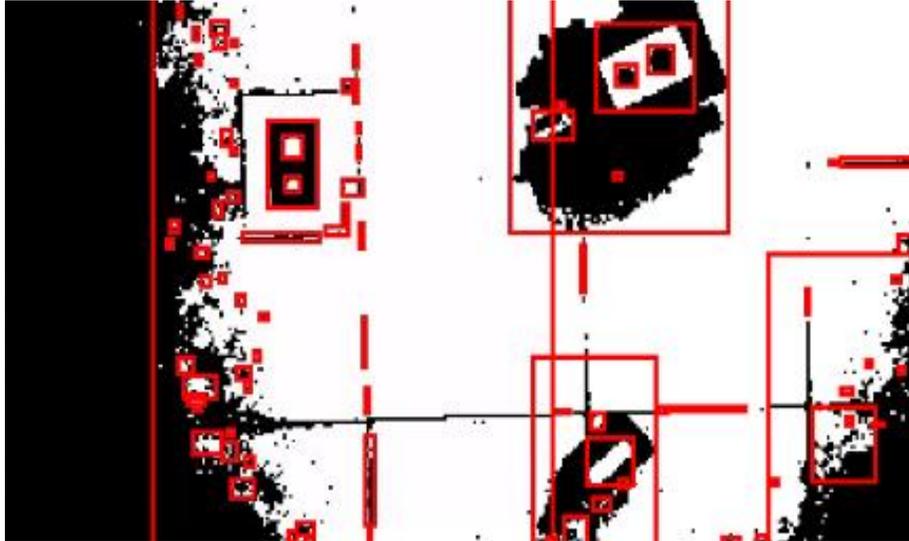


Fuente: autores.

4.4.1 Filtro por área (descartar contornos muy pequeños)

Se descartan los contornos pequeños que no corresponden a los patrones definidos para los círculos internos del patrón, mostrado en la Figura 50.

Figura 50. Filtrado descarte de contornos pequeños.

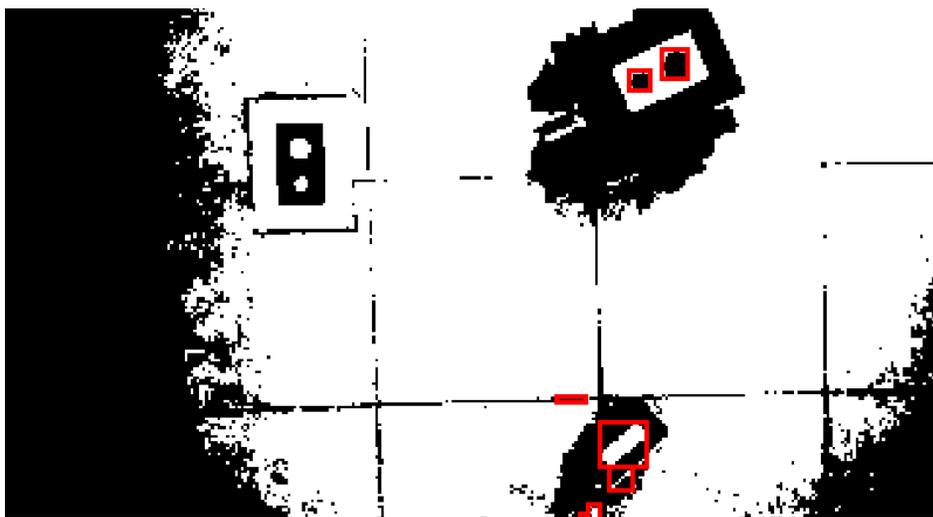


Fuente: autores.

4.4.2 Filtro de contornos dentro del patrón.

Se filtraron los contornos encontrados dentro del área delimitada por las figuras rectangulares previamente identificadas con el fin de evitar errores a la hora de reconocer los patrones circulares. Mostrado en la Figura 51.

Figura 51. Filtro del contorno dentro del patrón.

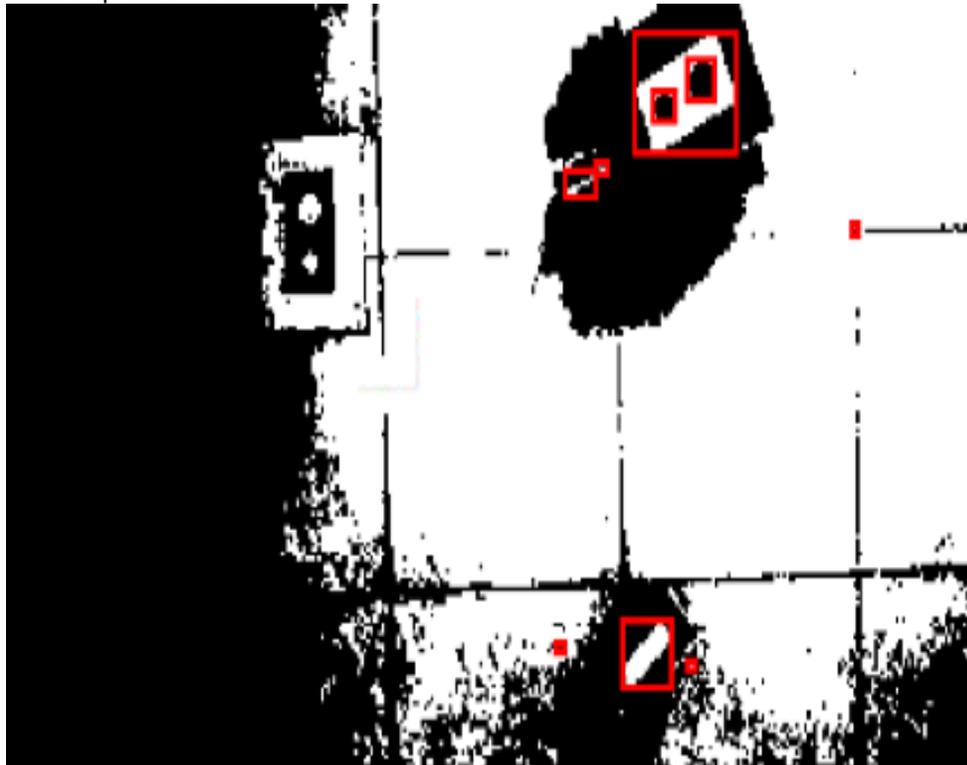


Fuente: autores.

4.4.3 Filtro por relación de lados

Al igual que en los contornos rectangulares, se realiza un análisis de filtrado por la relación que existe entre los lados de cada contorno encontrado, en el caso de las figuras circulares son aproximados a un cuadrado que eventualmente tendrán los lados iguales. Esta relación deberá ser aproximadamente igual a 1, con una tolerancia permisible y una variación de más o menos 0.5 debido a que la calidad de la imagen puede distorsionar los patrones pequeños, mostrado en la Figura 52.

Figura 52. Filtro por relación de lados.



Fuente: autores

4.4.4 Filtro por color del patrón circular (centro) y área

Finalmente se realiza un análisis de color, en el cual descartan los patrones en blanco que no corresponden al patrón circular de reconocimiento. Igualmente se realiza un análisis de área aproximada teniendo en cuenta un factor de conversión de px a cm.

Se realiza un análisis de los contornos internos encontrados en cada patrón rectangular hallado previamente, y se descartan los esquemas rectangulares que no posean exactamente dos estándares internos. Mostrado en la Figura 53.

Figura 53. Filtro por color de patrón circular.



Fuente: autores

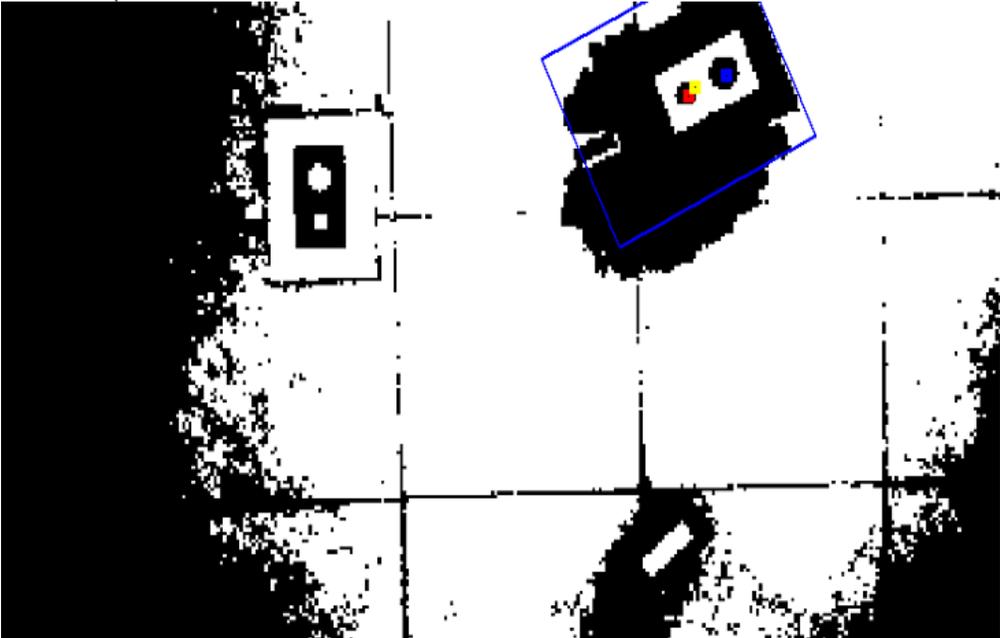
Se realiza el cálculo de la distancia entre el centro de los dos patrones internos, que se tomarán como base para la determinación de un factor de conversión más preciso y conveniente.

Con este factor de conversión se podrá realizar un filtro más riguroso por área real en cm^2 de cada uno de los patrones de reconocimiento, haciendo el sistema robusto para así tener consistencia en las determinaciones de cada uno de los fotogramas, mostrado en la Figura 54.

Realizando este proceso de filtrado en cada una de las imágenes, se obtiene el reconocimiento del patrón de interés, que al ser diseñado de forma robusta podrá ser determinado en cada fotograma que se realice del escenario en tiempo real, de esta forma el sistema de control de posición obtendrá valores más consistentes haciéndolo eficiente.

Posterior al procesamiento de imagen, se procede a calcular las distancias que se requieren para el sistema de control de posición. Estos trayectos corresponden a los recorridos relativos en cm que existen desde el centro del robot a los puntos de interés determinados por la trayectoria.

Figura 54. Filtro por área real en cm².



Fuente: autores

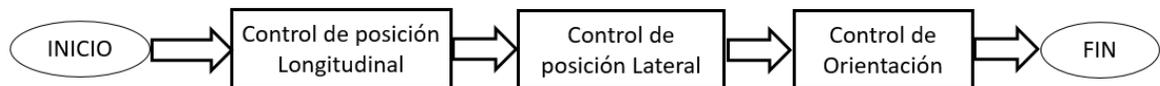
5. SISTEMA DE CONTROL

El sistema de control es un sistema multi-variable con tres lazos independientes (movimiento longitudinal, movimiento lateral, orientación) que pueden ser controlados de forma constante para posicionar el robot en una ubicación deseada dentro del escenario visto por la cámara. La implementación de dicho control por su naturaleza posee un integrador en su función de transferencia y una constantes K_n la cual variará para cada sistema.

Se realiza un diseño de controlador en lazo cerrado, retroalimentando el error presentado en cada medición para controlar cada una de las 3 variables independientes.

El diseño de control corresponde a un controlador tipo PD el cual no requiere integrador, puesto que la función de transferencia de cada uno de los sistemas a controlar (Plantas) posee en su naturaleza un integrador. Para la fase de implementación resulto necesaria la sintonización de cada uno de los controladores, y así encontrar los parámetros necesarios para un buen funcionamiento del sistema. El periodo de muestreo del controlador es directamente dependiente del procesamiento de imagen; ya que en un ciclo de trabajo se debe realizar el proceso completo de toma de imagen, procesamiento, análisis de distancias, proceso de los controladores y envío de información por puerto serial a través de *XBEE*. Es por esta razón que se usa un computador (Estación de comandos) con un buen procesador.

Figura 55. Diagrama de bloque Controladores



Fuente: Autores

6. GENERACIÓN DE TRAYECTORIAS

El sistema de reconocimiento de trayectoria se diseñó y se implementó de tal forma que el conjunto de elementos interactuaran de forma autónoma. La generación de trayectoria consiste en la obtención de siete puntos sobre el plano de la imagen mostrado en la Figura 57, capturando la superficie de movimientos en la cual se encuentra la carga a ser transportada, el robot monta-carga y el punto destino final de la carga. Con estos tres puntos se definen una serie de pequeños trayectos organizados con el fin de lograr la tarea de recoger la carga y dejarla en su lugar de destino.

Figura 56. Puntos de referencia (1 Robot, 2 Llegada, 3 Carga).



Fuente: Autores

La generación de trayectoria se define en un plano cartesiano de tres coordenadas, delimitado por la resolución de la cámara y la altura a la cual es instalada. Como primera condición, se debe obtener una imagen en la cual se incluyan los tres puntos de interés (carga, robot y destino), si esto no es posible, no se podrá calcular la trayectoria. Es de vital importancia poder determinar de manera precisa la orientación de cada uno de los puntos en relación con el plano de la imagen.

La generación de la trayectoria se realiza mediante la toma estática de un solo fotograma inicial, que detalla la posición y orientación de cada uno de los puntos objetivos. Se realiza un procesamiento de imagen especial de contornos y filtros para encontrar cada uno de los elementos de interés.

La trayectoria se empieza a definir desde el último punto (punto 7, mostrado en la Figura 57) “final de la trayectoria del robot”. Este punto es definido como el alejamiento del robot montacarga cuando ha dejado la carga en el punto 6. El punto 5, mostrado en la Figura 57, se define como el punto en el cual el robot debe

enganchar y levantar la carga. El punto 3, mostrado en la Figura 57, es considerado el punto de alineación a 30 cm de la carga y se evalúan dos posibles opciones para alinear el robot antes de recogerla, la carga sobre una estiva puede ser recogida por delante o por detrás, la mejor opción será escogida teniendo en cuenta que si una de las opciones está por fuera del plano de la imagen será descartada, y si los dos puntos están dentro de la imagen se escogerá el punto más cercano al robot. Si por algún motivo no es posible alinear el robot con la carga dentro del plano de la imagen (los dos puntos (2) posibles están fuera del plano de la imagen), la trayectoria no se podrá realizar.

El punto 4, mostrado en la Figura 57, es un punto intermedio entre la carga y el punto 3 recién hallado, y dependerá en gran medida de la opción escogida entre las dos posibles. Este punto estará a 18 cm de la carga en el mismo sentido del punto 3 hallado. Esto se diseñó con el fin de recoger la carga de una forma más lenta evitando que se caiga.

El punto 2, mostrado en la Figura 57, es considerado el punto de búsqueda, el cual alineará el robot para alcanzar el punto 3 de una forma más cómoda, este punto es relativo al punto 3 y estará ubicado a una distancia lateral de la mitad de la distancia total entre el robot y el punto 3. Igualmente se tendrán dos opciones a la derecha y a la izquierda de la posición relativa del punto 3. Se escogerá el mejor punto teniendo como criterio principal que el punto dentro de la imagen será el ideal y el que esté por fuera de los límites de la imagen, será descartado. Si los dos puntos quedan dentro del plano de la imagen se escogerá el más cercano al robot. Si por algún motivo no es posible posicionar el robot con el punto 3 dentro del plano de la imagen (los dos puntos (2) posibles están fuera del plano de la imagen), la trayectoria no se podrá realizar.

Finalmente, el punto 1 se escoge como el punto de partida o posición inicial del robot montacarga, mostrado en la Figura 57.

Figura 57. Puntos para Generar trayectoria.



Fuente: Autores

Se define el comportamiento de la estiva en cada uno de los puntos, este comportamiento se puede observar en la Tabla 3, esta orden se ejecuta cuando el robot complete el objetivo de llegar al punto de interés.

Tabla 3. Movimiento de la Estiva.

Punto	Movimiento
Punto 1	Abajo
Punto 2	Abajo
Punto 3	Abajo
Punto 4	Arriba
Punto 5	Arriba
Punto 6	Abajo
Punto 7	Abajo

Fuente: Autores

Se define el comportamiento de la orientación del robot en cada uno de los puntos, esta orden se ejecuta cuando el robot completa el objetivo de llegar al punto de interés y realiza el movimiento de la estiva (uñas). El ángulo de orientación será el calculado para posicionar y orientar el robot de forma “directa o derecha” al siguiente punto de interés, excepto el del punto 6 que tendrá la orientación de la posición final de la carga definida por el patrón de reconocimiento de descargue, y el punto 7 que finaliza con la misma orientación que la del punto 6), así se traza la trayectoria que deberá recorrer el Robot dentro del escenario, mostrado en la Figura 58.

Figura 58. Trayectoria.



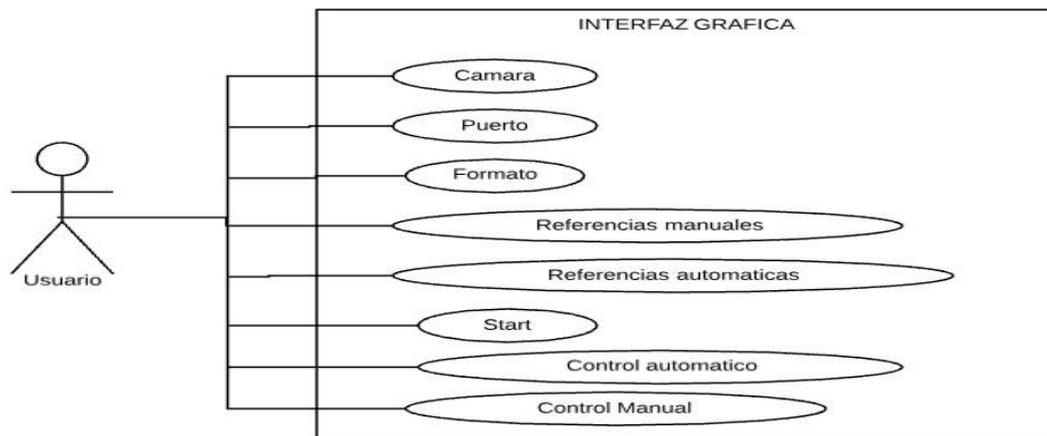
Fuente: Autores

7. DISEÑO DE LA INTERFAZ GRÁFICA

Para el diseño de la interfaz gráfica se pensó en que sea amigable con el usuario y el sistema, para que en caso del que el sistema presente problemas de detención, el usuario sea capaz de resolverlos sin ninguna dificultad.

Por tal motivo en el siguiente diagrama de usos de casos se muestran las características que posee la interfaz.

Figura 59. Diagrama inicial, Casos de usos



Fuente: Autores

La interfaz posee 8 casos de usos que brindan una seguridad y flexibilidad, se pensó en estas para que en caso de algún cambio de elemento utilizado llámese cámara, puerto o formato el usuario sea capaz de diferenciarlo y cambiarlo directamente a través de la interfaz y no se remita al código fuente donde si no tiene el conocimiento necesario para realizar esta modificación podría dañar el sistema.

El primer caso de uso sería la cámara, en el cual se analiza todos los puertos del computador y devuelve un listado de todas las cámaras que están conectadas y el usuario será el encargado de elegir la cámara con la que desee trabajar.

El segundo caso de uso sería el puerto, en este caso analiza los puertos y reconoce el puerto en el que está conectado el módulo XBee S1.

El tercer caso de uso es el formato, en donde se muestra un listado de todos los formatos de imagen que posee la cámara que se eligió anteriormente.

El cuarto caso de uso es las referencias manuales, en donde se toma una foto del escenario y el usuario a través del mouse señala los puntos que desee que el prototipo recorra para recoger, transportar y descargar la carga.

El Quinto caso de uso son las referencias automáticas, en este caso por medio de un algoritmo se eligen los puntos que el prototipo debe recorrer.

El Sexto caso de uso es start, en donde se pone a correr el programa ya con todas las características seleccionadas anteriormente, aquí se evalúa si se encuentra el patrón del robot, la carga, el punto final y si existe previamente una trayectoria generada ya sea de manera manual o automática.

El séptimo caso de uso es el control automático, en donde por medio de un algoritmo envía las órdenes para el control del prototipo y que cumpla a cabalidad la trayectoria generada.

El octavo caso de uso es el del control manual, este es por si el control automático falla el usuario podrá a través de la interfaz dirigir el prototipo, esto lo podría realizar siempre y cuando la comunicación siga funcionando.

8. RESULTADOS

Una vez planteada la necesidad de diseñar un prototipo para el levantamiento de carga, se construye un robot capaz de levantar un peso de hasta 2.5 kg, que utiliza una interfaz gráfica para el control y visualización de los movimientos y desplazamientos del mismo, y que en su conjunto permiten la generación automática de trayectorias a partir de la identificación del prototipo, la carga y el punto final del desplazamiento en donde será ubicada la carga, a continuación se realiza una prueba de trayectorias:

PRUEBA DE TRAYECTORIAS

Se realizaron diferentes pruebas del control y la trayectoria que debe seguir el Robot en dos escenarios diferentes obteniendo los siguientes resultados:

Escenario 1: Piso en porcelanato en un solo color, mostrado en la Figura 60.

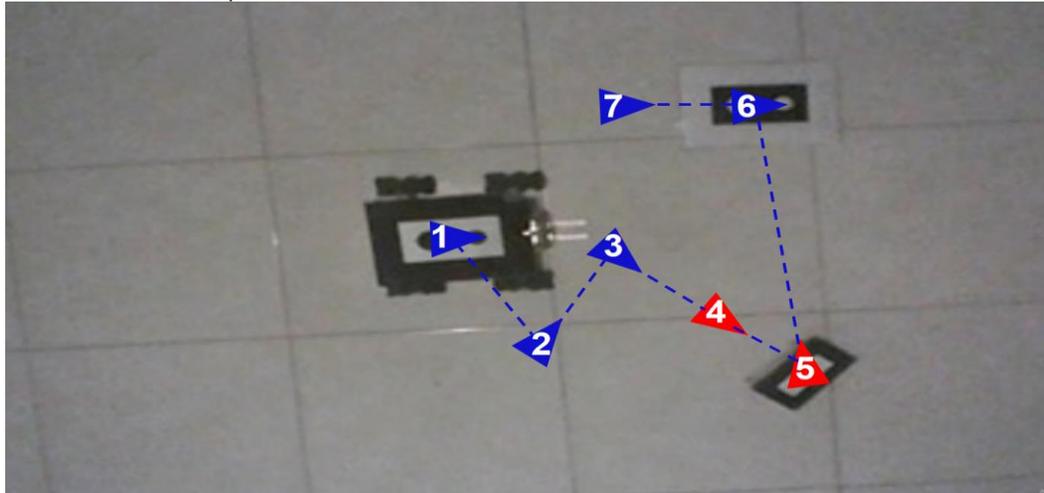
Figura 60. Escenario 1



Fuente: autores

Posición 1: Carga y Punto de llegada al frente del prototipo, mostrado en la Figura 61 y en la Tabla 4.

Figura 61. Escenario 1, posición 2.



Fuente: autores

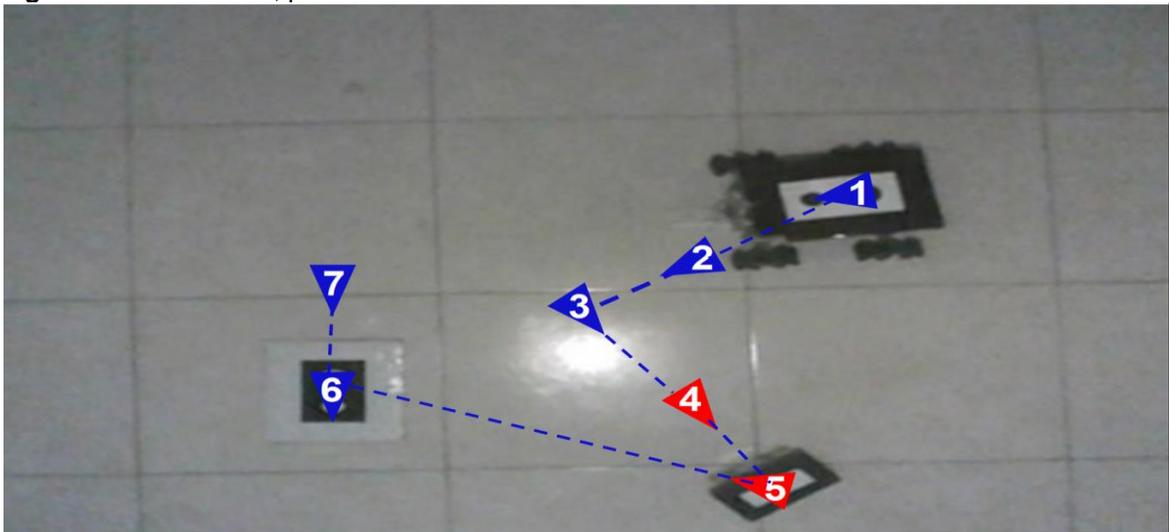
Tabla 4. Resultados escenario 1, posición 1.

Número de Pruebas	Número de veces Correctas	Número de veces incorrectas	Efectividad
30	29	1	96.666

Fuente: autores

Posición 2: Carga y Punto de llegada en diferente posición, mostrado en la Figura 62 y en la Tabla 5.

Figura 62. Escenario 1, posición 2.



Fuente: autores

Tabla 5. Resultados escenario 1, posición 3.

Número de Pruebas	Número de veces Correctas	Número de veces incorrectas	Efectividad
30	29	1	96.666

Fuente: autores

Escenario 2: Piso en porcelanato de color café con rayas, mostrado en la Figura 63.

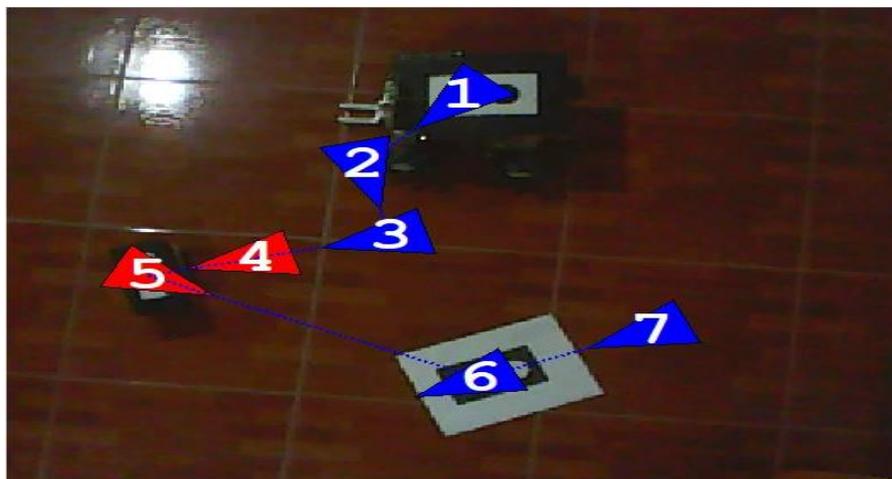
Figura 63. Escenario 2.



Fuente: autores

Posición 1: carga Frente al punto de llegada, mostrado en Figura 64 y en Tabla 6.

Figura 64. Escenario 2, posición 1.



Fuente: autores

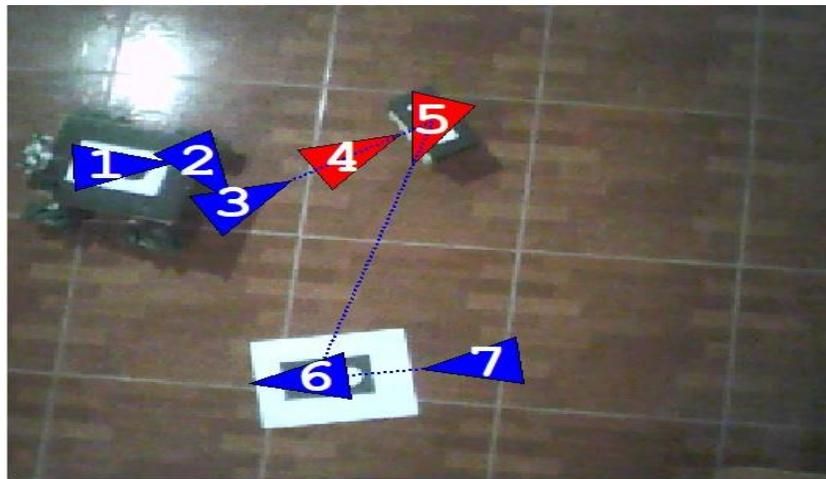
Tabla 6. Resultados escenario 2, posición 1.

Número de Pruebas	Número de veces Correctas	Número de veces incorrectas	Efectividad
30	29	1	96.666

Fuente: autores

Posición 2: carga y descarga en los extremos superiores del escenario con el robot en la mitad, mostrado en la Figura 65 y en Tabla 7.

Figura 65. Escenario 2, posición 2.



Fuente: autores

Tabla 7. Resultados escenario 2, posición 2.

Número de Pruebas	Número de veces Correctas	Número de veces incorrectas	Efectividad
30	29	1	96.666

Fuente: autores

9. CONCLUSIONES

En la adquisición de la imagen de un sistema de visión por computador, el ángulo de visión depende de dos factores: el lente de la cámara y la altura o la distancia.

Al aumentar la altura de la ubicación de la cámara se aumenta el error a la hora de adicionar los patrones del prototipo, teniendo en cuenta que al ampliar el ángulo de visión se tendrá mayor cantidad de elementos para analizar y procesar.

Al ampliar el ángulo de visión se requiere de mayor procesamiento de la imagen, lo que implica que el procesador que se tenga sea eficiente para que el sistema funcione de manera correcta y en tiempo real; lo que implica que la memoria Ram, y el procesador sean de las mejores características, para que los ciclos de lectura y control se ejecuten en tiempo real y el prototipo pueda hacer el levantamiento, desplazamiento y situado de la carga sin interrupciones.

El sistema de carga requirió la instalación de un circuito de fin de carrera en sus puntos superior e inferior, para evitar que el motor siga girando y genere algún tipo de daño en el sistema mecánico.

El tipo de ruedas seleccionadas otorga mayor grado de libertad al prototipo esto permite que el control de éste sea menos complejo.

A la hora de utilizar baterías LiPo se deben seguir las instrucciones y recomendaciones del fabricante, en este caso se utilizó una batería de litio de 3 celdas de un sistema de carga de balanceo que tiene un nivel mínimo y máximo de carga, a manera de experiencia se dejó descargar la batería por debajo del horizonte mínimo establecido por el fabricante, debido a esto fue necesario reemplazar la fuente por una nueva, lo que directamente implicó un gasto adicional y un retraso en la construcción del prototipo.

Se debe tener en cuenta el estado de la antena de las tarjetas xbee debido a que cualquier daño o mal contacto de la misma con la placa podría causar la pérdida de la comunicación entre el pc y el robot, por tal motivo el robot no recibiría las respectivas órdenes de desplazamiento o generara desplazamientos errados y con retardos considerables para cumplir el objetivo.

Para implementar el sistema en un escenario se debe tener en cuenta la luminosidad, el tamaño del recinto, la distancia entre las antenas transmisora y receptora, el reflejo de la superficie y se debe hacer los respectivos ajustes para que el sistema funcione sin ningún inconveniente.

Los ambientes con superficies demasiado brillantes generan al sistema en conjunto gran dificultad para la captura de la imagen, su procesamiento y por consiguiente la elaboración de una trayectoria correcta, teniendo en cuenta que los reflejos de dicha superficie producen en muchos casos la no identificación de los patrones de orientación.

El diseño y elaboración de los patrones de identificación de la carga, el robot y el punto final de descarga se deben hacer con las medidas (diámetros, distancia entre circunferencias, longitudes de los recuadros) establecidas en el software, esto evitara que se generen errores a la hora de ejecutar el programa.

Se realizaron pruebas en dos escenarios diferentes con variación de la ubicación de la carga, el prototipo y el punto de descarga, para cada posición se realizaron 30 intentos de los cuales se falló 1 vez por posición, dicha falla se presentó en el agarre de la carga a la hora de realizar su levantamiento o durante el desplazamiento de la misma.

10.RECOMENDACIONES

Para que el sistema sea más robusto, se debe utilizar una cámara con mayor resolución que cuente con un mayor ángulo de visión y a su vez se debe utilizar un computador con un procesador más avanzado y con mayor capacidad de memoria Ram, con el fin de que la imagen sea de gran calidad o gran resolución y su procesamiento sea con el menor número de errores y retardos posibles.

El sistema de control se puede reducir a un solo microcontrolador que tenga las salidas necesarias PWM para el control de los motores.

Se puede implementar una cuadrilla de prototipos de robot de carga teniendo en cuenta que el módulo XBEE permite realizar una configuración de un emisor y varios receptores (configuración multipunto).

Para mejorar el robot de carga se podrían reemplazar los pulsadores por sensores de proximidad tanto en la parte superior como en la parte inferior.

Se puede diseñar la interfaz gráfica en código abierto para no depender de licencias ni estar sujetos a las entidades que posean estas licencias.

Se puede mejorar el sistema de sujeción de la carga mediante la modificación de las palas que levantan la carga, se podría prolongar la longitud de las pinzas sobre las cuales va soportada la estiva esto permite y mejora el acoplamiento de la carga para el desplazamiento del prototipo.

Si por determinada circunstancia la generación automática de la trayectoria presenta algún error, el sistema ofrece la opción de generar las trayectorias de forma manual para que un operario las establezca.

BIBLIOGRAFÍA

Ackerman, E. (2018). AI Startup Using Robots and Lidar to Boost Productivity on Construction Sites. *IEEE SPECTRUM*, Enero.

Alegre, E. P. (2016). *Conceptos y métodos por computador*. Madrid España: Grupo de Visión del Comité Español de Automática (CEA).

Blazquez, J. (2011). *Introducción a los sistemas de comunicación inalámbricos* (Vol. 1). Catalunya España, Catalunya : Universitat oberta de Catalunya.

Casado Fernández, M. C. (2016). *Servicios informáticos U.C.M.* Madrid España: Apoyo a la investigación y docencia.

CRAIG, J. J. (2015). *Robótica* . México, México: Impreso en México. Pearson.

Febrero 28, 2. R. (28 de Febrero de 2017). *Robótica - REVISTA PROWARE*. Obtenido de Robótica - REVISTA PROWARE: <http://www.revistaproware.com/2017/02/28/robots-reemplazaran-a-dos-tercios-de-trabajadores-en-el-mundo/>

Hurtado, R. J.-F.-D. (2013). Control de Movimiento de un Robot Humanoide por medio de Visión de Máquina y Réplica de Movimientos Humanos. *Revista INGE UC*, 44-51.

Letters, I. R. (2017). A Loop-Shaped Flexible Mechanism for Robotic Needle Steering. *IEEE Robotics and Automation Letters*, December.

López2, A. T. (2014). Exploración integrada probabilista para robots móviles en ambientes complejos. *Computación y Sistemas*, 10.

Magazine, R. -D. (4 de Octubre de 2017). *Robots - Discover Magazine*. Obtenido de Robots - Discover Magazine: <http://discovermagazine.com/2017/may-2017/caring-computers>

Morena, R. d. (27 de Marzo de 2017). *Muy Interesante*. Obtenido de Muy Interesante: <https://www.muyinteresante.es/tecnologia/video/atlas-el-robot-que-es-capaz-de-dar-un-salto-hacia-atrás>

Wierenga, T. (14 de Noviembre de 2017). *SERVO MAGAZINE*. Obtenido de Using Stepper Motors for Wheeled Robot Propulsion: http://www.servomagazine.com/index.php/magazine/article/september2014_Wierenga

ANEXOS

ANEXO A: MONTAJE DEL SISTEMA DE COMUNICACIÓN

Para llevar a cabo el montaje del sistema de comunicación del prototipo se analizaron los diferentes modos de comunicación inalámbrica teniendo en cuenta la relación costo beneficio, se decidió trabajar con el módulo XBEE s1, mostrado en la Figura 66, el cual maneja un protocolo de comunicación llamado Zigbee que está basado en el estándar de comunicaciones para redes inalámbricas IEEE_802.15.4.

“Las comunicaciones Zigbee se realizan en la banda libre de 2.4GHz. A diferencia de bluetooth, este protocolo no utiliza FHSS (Frequency hopping), sino que realiza las comunicaciones a través de una única frecuencia, es decir, de un canal. Normalmente puede escogerse un canal de entre 16 posibles. El alcance depende de la potencia de transmisión del dispositivo así como también del tipo de antenas utilizadas (cerámicas, dipolos, etc) El alcance normal con antena dipolo en línea vista es de aproximadamente (tomando como ejemplo el caso de MaxStream, en la versión de 1mW de potencia) de 100m y en interiores de unos 30m. La velocidad de transmisión de datos de una red Zigbee es de hasta 256kbps. Una red Zigbee la pueden formar, teóricamente, hasta 65535 equipos, es decir, el protocolo está preparado para poder controlar en la misma red esta cantidad enorme de dispositivos”¹².

Figura 66. Módulo XBEE s1.

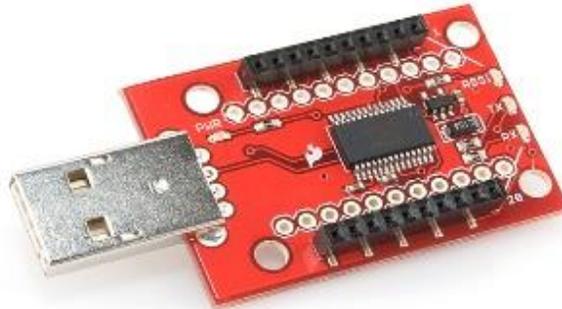


Fuente: tomado de, modulo-xbee.com.co, elementos y equipos de electrónica y telecomunicaciones

Se utilizaron dos módulos XBEE S1, de los cuales, uno es el que lleva la información a los microcontroladores y el otro al computador; para la programación del XBEE S1 fue necesario recurrir al módulo *XBEE EXPLORER DONGLE*, mostrado en la Figura 67, ya que el solo modulo no cuenta con una conexión directa la computador.

¹² Oyarce Andres, Guia del Usuario Xbee Series1 Documento Preliminar, Santiago de chile ,2008.

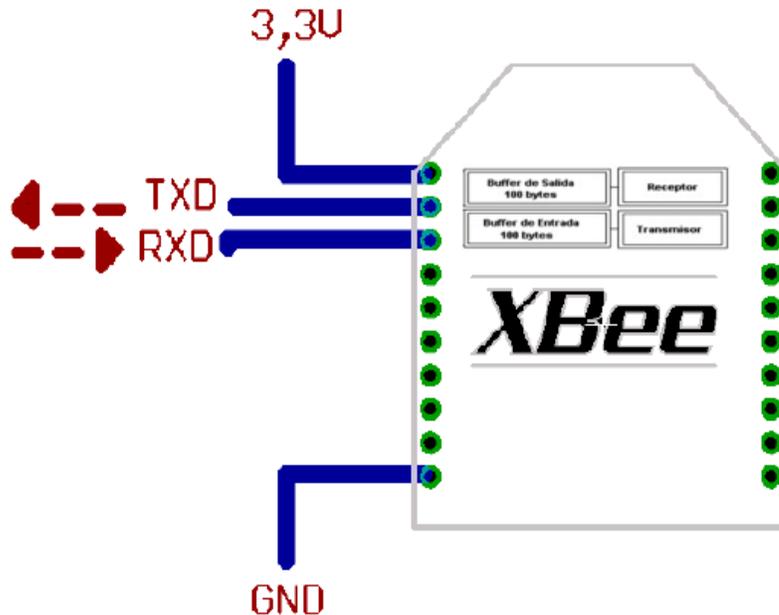
Figura 67. Módulo XBEE EXPLORER DONGLE.



Fuente: tomado de, modulo-xbee.com.co, elementos y equipos de electrónica y telecomunicaciones.

El modo con el que se configuro los *XBEE* fue el modo transparente, este modo es el más fácil de configurar ya que solo se trabaja con el puerto UART (Universal Asynchronous Receiver-Transmitter); este modo por lo general se destina a una comunicación punto a punto donde la información ingresa por el pin 3 (Data in) se guarda en el buffer de entrada dependiendo del tiempo programado por el usuario siendo posterior mente transmitido y todo lo que ingresa como paquete RF, se guarda en el buffer de salida y se envía por el pin 2(Data out), mostrado en la Figura 68.

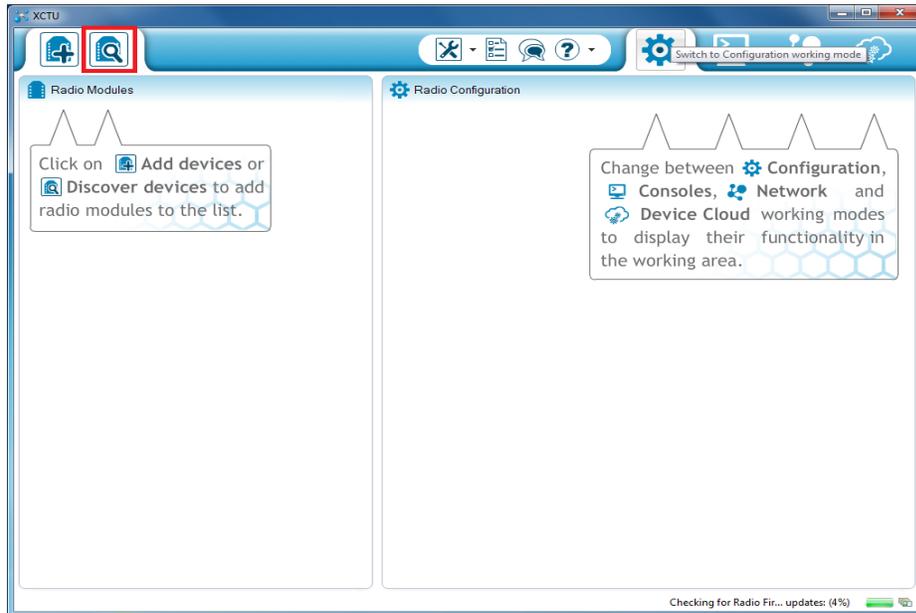
Figura 68. Configuración del módulo XBEE transparente.



Fuente: autores.

Para la configuración de los módulos XBEE se inicia por abrir el software XCTU, buscando y seleccionado la opción "Discover Radio", mostrado en la Figura 69.

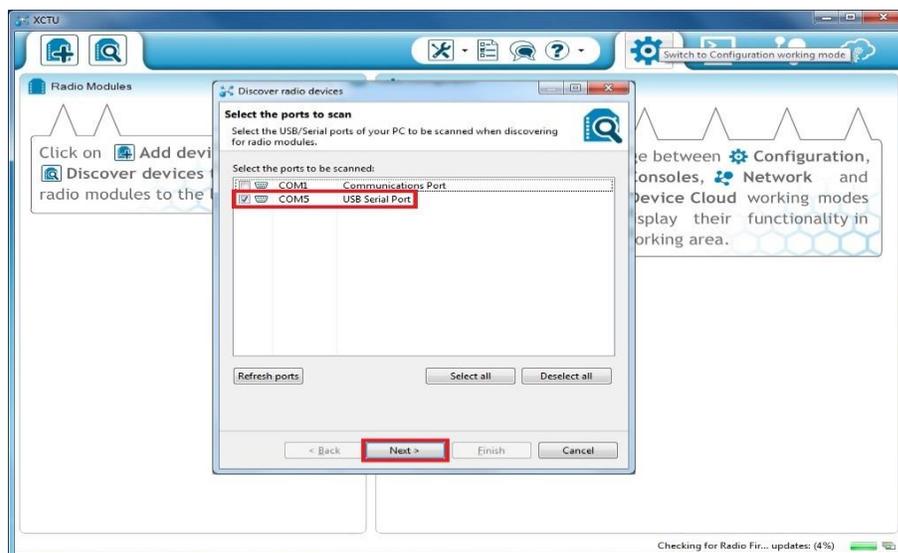
Figura 69. Interfaz XCTU.



Fuente: autores.

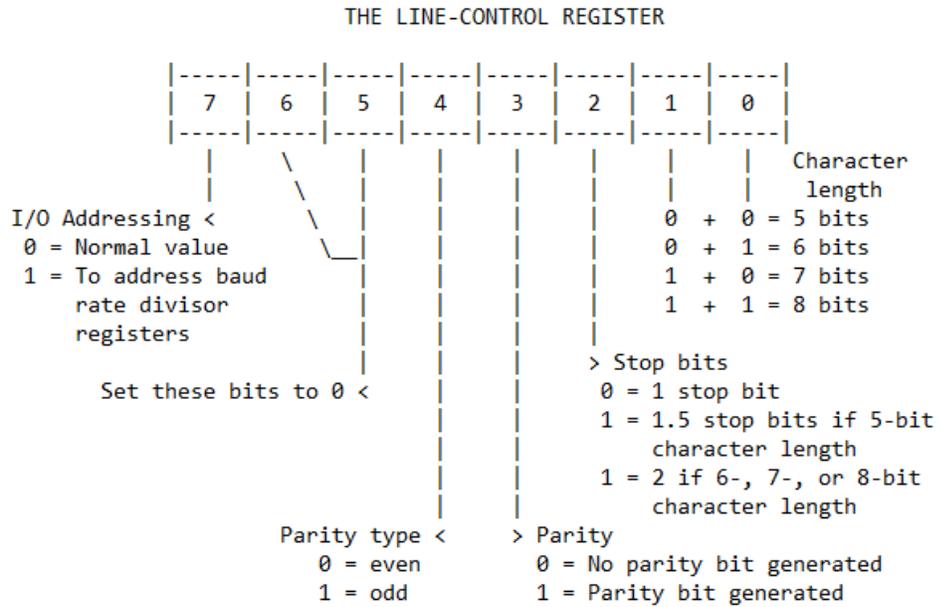
Se asigna un puerto para el modulo de configuración por defecto (COM con una instrucción *OPEN "COM1: ..."* u *OPEN "COM2: ..."* en este caso se asigno el puerto *COM5* se observar en la Figura 70), esto se realiza por medio de las operaciones entradas de comunicaciones, lo que se establece más envío de byte correspondiente para el registro de un control de linea, mostrado en la Figura 71.

Figura 70. Selección del puerto.



Fuente: autores.

Figura 71. Configuración de puertos.



Fuente: tomado de, support.microsoft.com, Cómo cambiar COM1 o COM2.

ANEXO B: CODIGO DEL PROCESAMIENTO DE LA IMAGEN

A continuación, se muestra el código que realiza el procesamiento de la imagen.

```
img=getsnapshot(vid_obj);

imagen_referencia=img;
img_gris=rgb2gray(img);

umb=get(handles.slider_umbral,'value');
set(handles.text_umbral,'String',round(100*umb)/100
bw=im2bw(img_gris,umb);

area_imagen=size(bw,1)*size(bw,2);
[L1 N1]=bwlabel(bw);
[L2 N2]=bwlabel(~bw);
contornos_1=regionprops(L1,'BoundingBox','Area','Centroid','MajorAxis
Length','MinorAxisLength','Eccentricity','Orientation');
contornos_2=regionprops(L2,'BoundingBox','Area','Centroid','MajorAxis
Length','MinorAxisLength','Eccentricity','Orientation');
contornos=[contornos_1; contornos_2];

if(robot_encontrado==0)
for i=1:length(contornos)

if (contornos(i).Area > 20)

if(contornos(i).BoundingBox(3)*contornos(i).BoundingBox(4)<
area_imagen*0.8)

factor_lados=contornos(i).MajorAxisLength/contornos(i).MinorAxisLength;
if (factor_lados>1.3 && factor_lados<1.9)
factor_1=14.4/contornos(i).MinorAxisLength;

Area_Rectangulo_1=contornos(i).Area*factor_1*factor_1;
if(Area_Rectangulo_1>100 && Area_Rectangulo_1<240)
factor_2=22.5/contornos(i).MajorAxisLength; %17+5.5=22.5
Area_Rectangulo_2=contornos(i).Area*factor_2*factor_2;
if(Area_Rectangulo_2>100 && Area_Rectangulo_2<240)

If
bw(floor(contornos(i).Centroid(2)),floor(contornos(i).Centroid(1)))==1)

if (contornos(i).Eccentricity>0.6 && contornos(i).Eccentricity<0.9)
x1=contornos(i).BoundingBox(1);
y1=contornos(i).BoundingBox(2);
x2=contornos(i).BoundingBox(1)+contornos(i).BoundingBox(3);
y2=contornos(i).BoundingBox(2)+contornos(i).BoundingBox(4);
contador_contornos_internos=0;
circulos=[]; for j=1:length(contornos)
```

```

if(contornos(j).Area > 2)

if(contornos(j).BoundingBox(1)>x1 && contornos(j).BoundingBox(1)<x2)
if(contornos(j).BoundingBox(2)>y1 && contornos(j).BoundingBox(2)<y2)
if(contornos(j).Centroid(1)>x1 && contornos(j).Centroid(1)<x2)
if(contornos(j).Centroid(2)>y1 && contornos(j).Centroid(2)<y2)

relacion_lados=contornos(j).BoundingBox(3)/contornos(j).BoundingBox(4);
if(relacion_lados>1), relacion_lados=1/relacion_lados; end
if(relacion_lados>0.5)
distancia_centro=sqrt((contornos(i).Centroid(1)-
contornos(j).Centroid(1))^2+(contornos(i).Centroid(2)-
contornos(j).Centroid(2))^2);
if (distancia_centro*factor_1*factor_1 <
(contornos(i).MinorAxisLength/2.0)*factor_1*factor_1)

if
(bw(floor(contornos(j).Centroid(2)),floor(contornos(j).Centroid(1)))==0
)
Area_circulo_1=contornos(j).Area*factor_1*factor_1;
if(Area_circulo_1>2 && Area_circulo_1<40)
Area_circulo_2=contornos(j).Area*factor_2*factor_2;
if(Area_circulo_2>2 && Area_circulo_2<40)
contador_contornos_internos=contador_contornos_internos+1;
circulos=[circulos j];

```

ANEXO C: CODIGO DE LA GENERACION DE LA TRAYECTORIA

A continuación, se muestra el código que realiza la generación de la trayectoria.

```
% empiezo a realizar trayectoria automatica

% PUNTO 7 (Ultimo punto)
% Coordenadas cartecianas relativas al angulo del punto 5
punto7_u=-30*(1/factor_cm_px);
punto7_v=0;
% Pasar a polares
[punto7_phi,punto7_n]=cart2pol(punto7_u,punto7_v);
% Correccion de angulo
punto7_theta=deg2rad(final_orientacion)+punto7_phi;
punto7_m=punto7_n;
% Pasar a cartesianas
[punto7_x_prima,punto7_y_prima]=pol2cart(punto7_theta,punto7_m);
punto7_x=punto7_x_prima+final_centro_x;
punto7_y=final_centro_y-punto7_y_prima;
punto7_angulo=final_orientacion;
punto7_estiva=0;

% PUNTO 6 (Punto de descarga)
% Coordenadas cartecianas
punto6_x=final_centro_x;
punto6_y=final_centro_y;
punto6_angulo=final_orientacion;
punto6_estiva=0;

% PUNTO 5 (Punto de Carga)
% Coordenadas cartecianas
punto5_x=carga_centro_x;
punto5_y=carga_centro_y;
punto5_angulo=rad2deg(atan2(-(punto6_y-punto5_y),(punto6_x-
punto5_x)));
punto5_estiva=1;

% PUNTO 3 (Punto de alineacion)
% Coordenadas cartecianas relativas al angulo del punto 5 opcion 1
punto3_u=0;
punto3_v=40*(1/factor_cm_px);
% Pasar a polares
[punto3_phi,punto3_n]=cart2pol(punto3_u,punto3_v);
% Correccion de angulo
punto3_theta=deg2rad(carga_orientacion)+punto3_phi;
punto3_m=punto3_n;
% Pasar a cartesianas
[punto3_x_prima,punto3_y_prima]=pol2cart(punto3_theta,punto3_m);
punto3_x1=punto3_x_prima+carga_centro_x;
punto3_y1=carga_centro_y-punto3_y_prima;

% Coordenadas cartecianas relativas al angulo del punto 5 opcion 2
```

```

punto3_u=0;
punto3_v=-40*(1/factor_cm_px);
% Pasar a polares
[punto3_phi,punto3_n]=cart2pol(punto3_u,punto3_v);
% Correccion de angulo
punto3_theta=deg2rad(carga_orientacion)+punto3_phi;
punto3_m=punto3_n;
% Pasar a cartesianas
[punto3_x_prima,punto3_y_prima]=pol2cart(punto3_theta,punto3_m);
punto3_x2=punto3_x_prima+carga_centro_x;
punto3_y2=carga_centro_y-punto3_y_prima;

% limites del perimetro
limite_pixeles=15*(1/factor_cm_px);

% evaluando el punto 3-1 dentro del limite
if (punto3_x1<limite_pixeles || punto3_x1>size(bw,2)-limite_pixeles
|| punto3_y1<limite_pixeles || punto3_y1>size(bw,1)-limite_pixeles)
    punto3_1=0; %0 sale
else
    punto3_1=1; %1 entra
end

% evaluando el punto 3-2 dentro del limite
if (punto3_x2<limite_pixeles || punto3_x2>size(bw,2)-limite_pixeles
|| punto3_y2<limite_pixeles || punto3_y2>size(bw,1)-limite_pixeles)
    punto3_2=0; %0 sale
else
    punto3_2=1; %1 entra
end

disp([punto3_1 punto3_2])

% escoge el punto adecuado
if (punto3_1==1 && punto3_2==0)
    punto3_x=punto3_x1;
    punto3_y=punto3_y1;
    punto3_1=1;
    punto3_2=0;
end
if (punto3_1==0 && punto3_2==1)
    punto3_x=punto3_x2;
    punto3_y=punto3_y2;
    punto3_2=1;
    punto3_1=0;
end
if (punto3_1==1 && punto3_2==1)
    % se escoge el punto mas cercano al robot
    dist_punto1_punto3_1=sqrt((punto3_x1-
robot_centro_x)^2+(punto3_y1-robot_centro_y)^2);
    dist_punto1_punto3_2=sqrt((punto3_x2-
robot_centro_x)^2+(punto3_y2-robot_centro_y)^2);
    if(dist_punto1_punto3_1<dist_punto1_punto3_2)

```

```

        punto3_x=punto3_x1;
        punto3_y=punto3_y1;
        punto3_1=1;
        punto3_2=0;
    else
        punto3_x=punto3_x2;
        punto3_y=punto3_y2;
        punto3_2=1;
        punto3_1=0;
    end
end

% si no encuentra ningun punto para alinearse
if (punto3_1==0 && punto3_2==0)
    errordlg('El robot no se puede alinear con la carga');
    error_trayectoria=1;
else
    punto3_angulo=rad2deg(atan2(-(carga_centro_y-
punto3_y),(carga_centro_x-punto3_x)));
    punto3_estiva=0;
end

% PUNTO 4 (Punto de alineacion)
% Coordenadas cartecianas relativas al angulo del punto 5 opcion 1
if (error_trayectoria~=1)
    punto4_u=0;
    if (punto3_1==1), punto4_v=18*(1/factor_cm_px); end
    if (punto3_2==1), punto4_v=-18*(1/factor_cm_px); end
    % Pasar a polares
    [punto4_phi,punto4_n]=cart2pol(punto4_u,punto4_v);
    % Correccion de angulo
    punto4_theta=deg2rad(carga_orientacion)+punto4_phi;
    punto4_m=punto4_n;
    % Pasar a cartesianas
    [punto4_x_prima,punto4_y_prima]=pol2cart(punto4_theta,punto4_m);
    punto4_x=punto4_x_prima+carga_centro_x;
    punto4_y=carga_centro_y-punto4_y_prima;
    punto4_angulo=rad2deg(atan2(-(carga_centro_y-
punto4_y),(carga_centro_x-punto4_x)));
    punto4_estiva=1;
end

% PUNTO 2 (Punto de Busqueda)
% Coordenadas cartecianas relativas al angulo del punto 3
if (error_trayectoria~=1)
    punto2_u=0;
    punto2_v=sqrt((punto3_x-robot_centro_x)^2+(punto3_y-
robot_centro_y)^2)/2;
    % Pasar a polares
    [punto2_phi,punto2_n]=cart2pol(punto2_u,punto2_v);
    % Correccion de angulo
    punto2_theta=punto2_phi+deg2rad(punto3_angulo);
    punto2_m=punto2_n;

```

```

% Pasar a cartesianas
[punto2_x_prima,punto2_y_prima]=pol2cart(punto2_theta,punto2_m);
punto2_x1=punto3_x+punto2_x_prima;
punto2_y1=punto3_y-punto2_y_prima;

% Coordenadas cartecianas relativas al angulo del punto 3
punto2_u=0;
punto2_v=-sqrt((punto3_x-robot_centro_x)^2+(punto3_y-
robot_centro_y)^2)/2;
% Pasar a polares
[punto2_phi,punto2_n]=cart2pol(punto2_u,punto2_v);
% Correccion de angulo
punto2_theta=punto2_phi+deg2rad(punto3_angulo);
punto2_m=punto2_n;
% Pasar a cartesianas
[punto2_x_prima,punto2_y_prima]=pol2cart(punto2_theta,punto2_m);
punto2_x2=punto3_x+punto2_x_prima;
punto2_y2=punto3_y-punto2_y_prima;

% evaluando el punto 2-1 dentro del limite
if (punto2_x1<limite_pixeles || punto2_x1>size(bw,2)-
limite_pixeles || punto2_y1<limite_pixeles || punto2_y1>size(bw,1)-
limite_pixeles)
    punto2_1=0; %0 sale
else
    punto2_1=1; %1 entra
end

% evaluando el punto 2-2 dentro del limite
if (punto2_x2<limite_pixeles || punto2_x2>size(bw,2)-
limite_pixeles || punto2_y2<limite_pixeles || punto2_y2>size(bw,1)-
limite_pixeles)
    punto2_2=0; %0 sale
else
    punto2_2=1; %1 entra
end

if (punto2_1==1 && punto2_2==0)
    punto2_x=punto2_x1;
    punto2_y=punto2_y1;
end
if (punto2_1==0 && punto2_2==1)
    punto2_x=punto2_x2;
    punto2_y=punto2_y2;
end
if (punto2_1==1 && punto2_2==1)
    % se escoge el punto mas cercano al robot
    dist_punto1_punto2_1=sqrt((punto2_x1-
robot_centro_x)^2+(punto2_y1-robot_centro_y)^2);
    dist_punto1_punto2_2=sqrt((punto2_x2-
robot_centro_x)^2+(punto2_y2-robot_centro_y)^2);
    if(dist_punto1_punto2_1<dist_punto1_punto2_2)
        punto2_x=punto2_x1;
        punto2_y=punto2_y1;
    end
end

```

```

        else
            punto2_x=punto2_x2;
            punto2_y=punto2_y2;
        end
    end

    % si no encuentra ningun punto para alinearse
    if (punto2_1==0 && punto2_2==0)
        errordlg('El robot no se puede alinear con la carga');
        error_trayectoria=1;
    else
        punto2_angulo=rad2deg(atan2(-(punto3_y-punto2_y), (punto3_x-
punto2_x)));
        punto2_estiva=0;
    end
end

% PUNTO 1 (Punto de partida)
% Coordenadas cartecianas relativas al angulo del punto 3
if (error_trayectoria~=1)
    punto1_x=robot_centro_x;
    punto1_y=robot_centro_y;
    punto1_angulo=rad2deg(atan2(-(punto2_y-punto1_y), (punto2_x-
punto1_x)));
    punto1_estiva=0;
    pause(1);
end

if (error_trayectoria~=1)
    x=[punto1_x punto2_x punto3_x punto4_x punto5_x punto6_x
punto7_x];
    y=[punto1_y punto2_y punto3_y punto4_y punto5_y punto6_y
punto7_y];
    ang=[punto1_angulo punto2_angulo punto3_angulo punto4_angulo
punto5_angulo punto6_angulo punto7_angulo];
    est=[punto1_estiva punto2_estiva punto3_estiva punto4_estiva
punto5_estiva punto6_estiva punto7_estiva];
    set(handles.text29,'String','Trayectoria: SI'); % poner valor del
humbral en el text

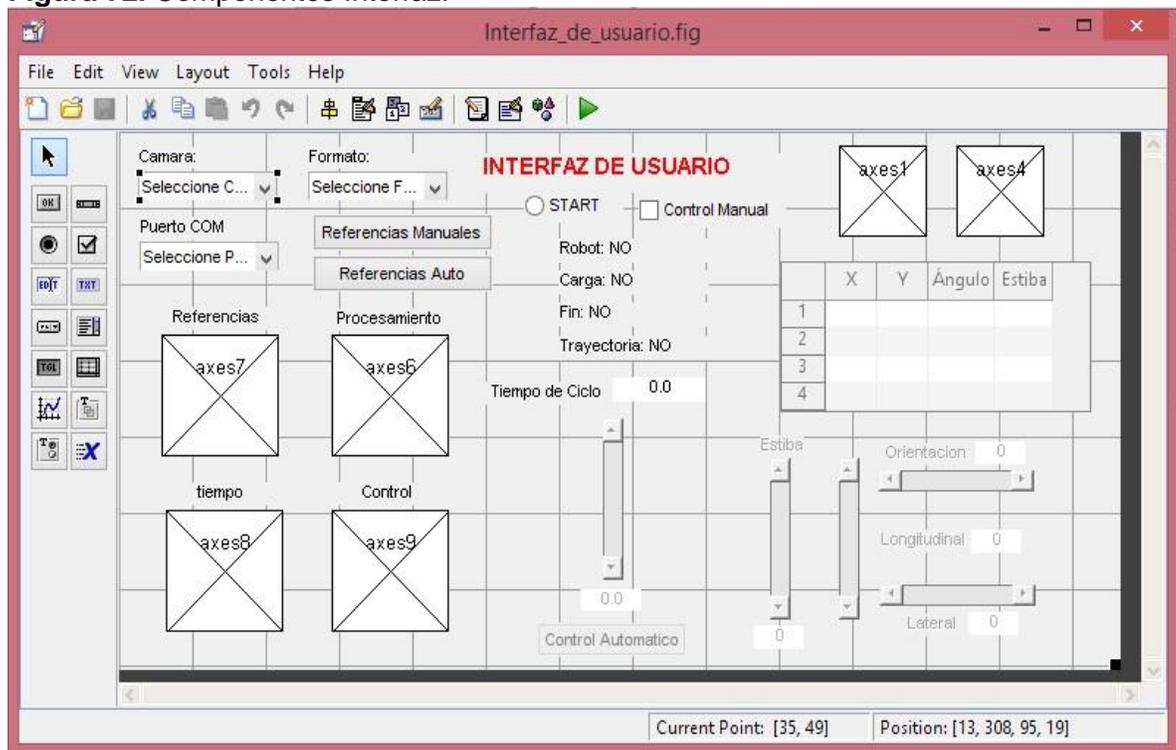
```

ANEXO D: MANUAL DE USUARIO DE LA INTERFAZ

La interfaz se desarrolló utilizando el software *MATLAB* bajo la licencia de la Universidad SurColombiana, este permite una programación orientada a objetos mediante un entorno visual *GUIDE*.

La interfaz gráfica está compuesta por los siguientes componentes (axes, button, push button, slider, entre otros. Mostrado en la Figura 72.

Figura 72. Componentes Interfaz.

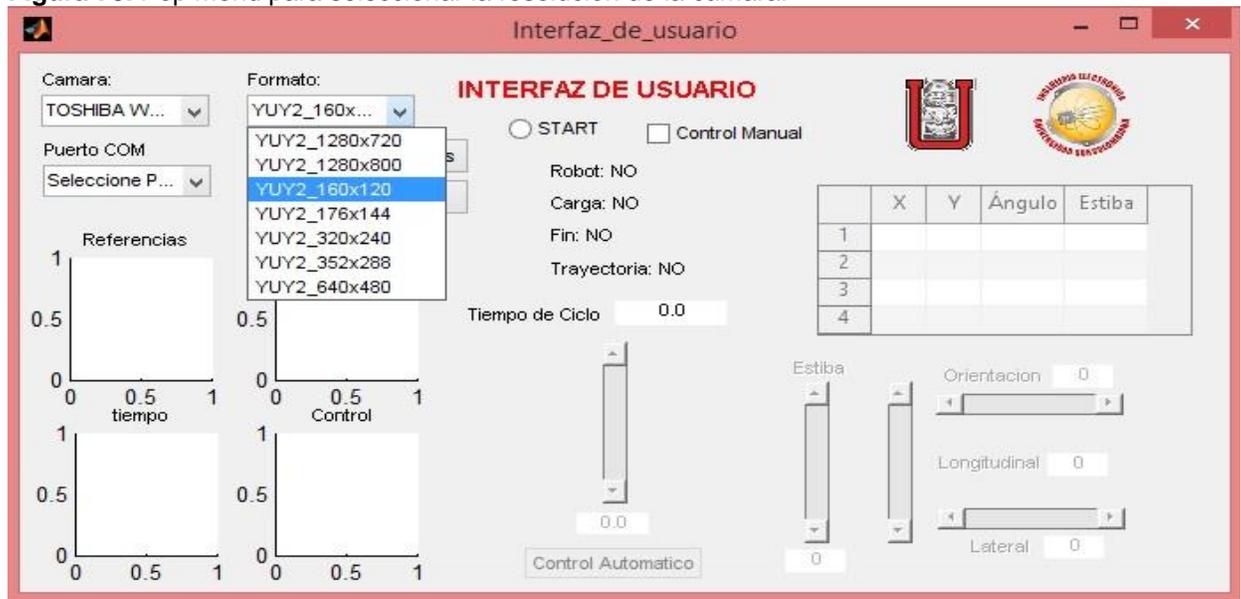


Fuente: autores.

Se procede a realizar la programación que incluye: declaración de las variables globales, tiempo de muestreo, variables de control, proceso (que sirven de banderas o indicadores), configuración de la imagen para que la interfaz quede en el centro, la recepción de la información de las cámaras y la colocación en el pop menú y la obtención los puertos seriales disponibles.

Se recomienda tener en cuenta las características que posee el computador donde se va a ejecutar el software a la hora de seleccionar el tipo de formato de video que va a utilizar la cámara, debido a que entre más pixeles utilizados mayor va a ser el requerimiento de procesador y el programa puede tornarse lento, mostrado en la Figura 73.

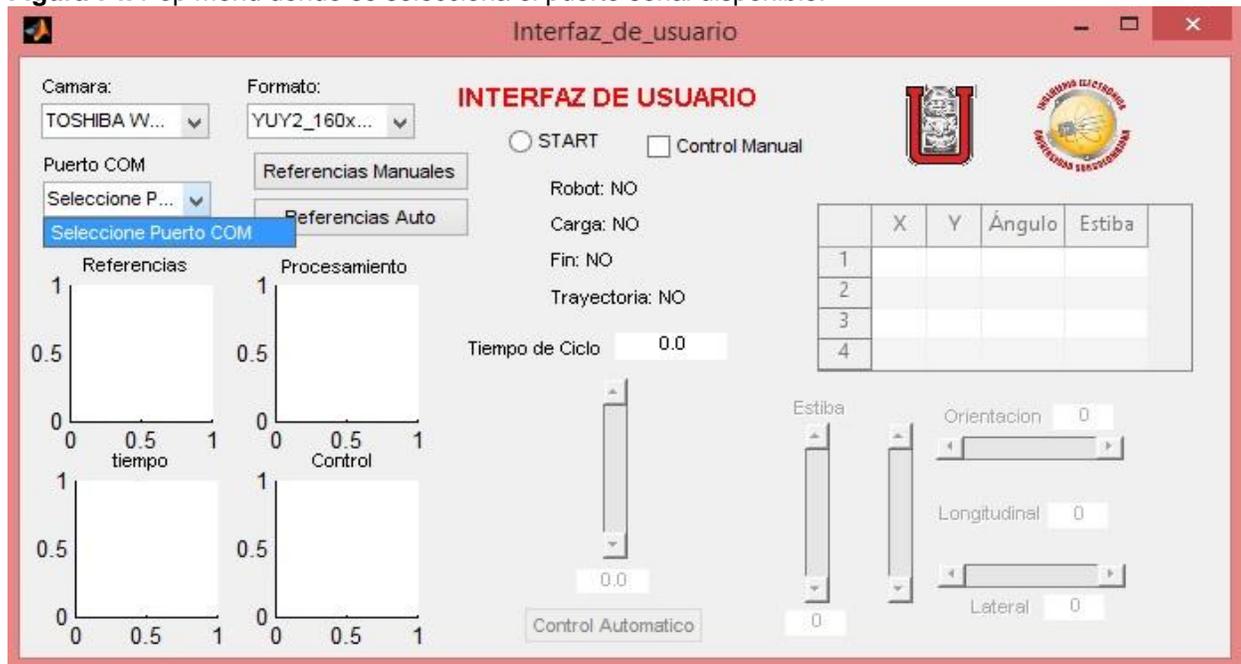
Figura 73. Pop menú para seleccionar la resolución de la cámara.



Fuente: autores.

Al seleccionar la cámara se define el puerto, el *Select* reconoce los puertos que utiliza el computador donde se conecta la tarjeta XBEE, junto al módulo Explorer Dongle., mostrado en la Figura 74.

Figura 74. Pop menú donde se selecciona el puerto serial disponible.



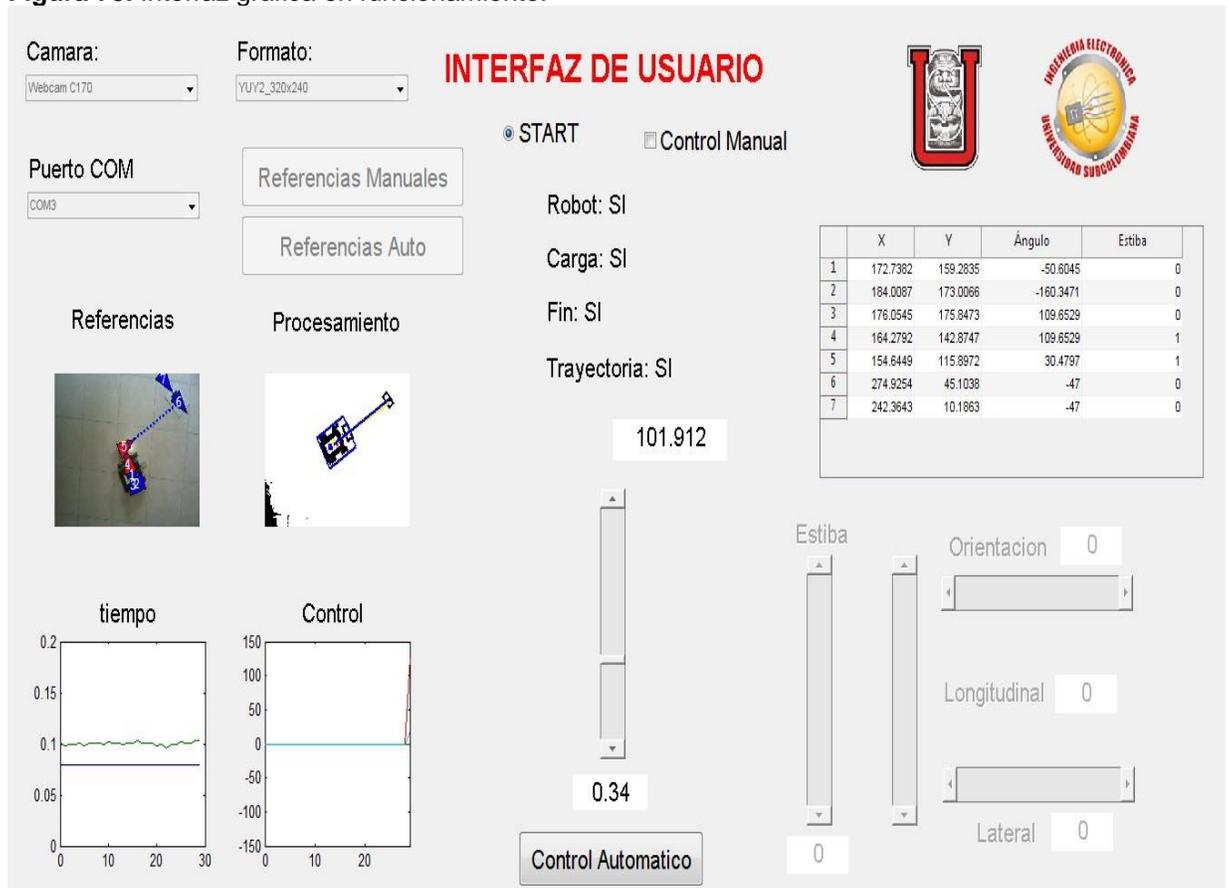
Fuente: autores.

Cuando se terminan de hacer las configuraciones previas, se tienen en cuenta las referencias para generar la trayectoria, la interfaz da dos opciones una manual y otra automática.

Las referencias automáticas son los puntos que el programa identifica para poder generar la trayectoria, y las referencias manuales el usuario las identifica y puede colocar los puntos que crea necesarios para que el prototipo pueda recoger la carga e ir a descargarla en un punto determinado.

Después de establecida la trayectoria, sea de manera automática o manual se le da en el radio button *Start*, el cual comienza a correr la programación del software de control para que el prototipo siga la trayectoria designada, mostrada en la Figura 75.

Figura 75. Interfaz gráfica en funcionamiento.



Fuente: autores.

Cuando se establecen las referencias ya sean de manera manual o automática, se puede observar que la interfaz muestra la imagen de la trayectoria que va a seguir el robot, esto se puede visualizar en el axis identificado como referencia, la trayectoria esta demarcada por líneas rectas de color azul y los puntos de ubicación

del prototipo están identificados con una figura geométrica triangular, donde la punta se muestra en la vista del usuario, e indica la orientación de desplazamiento del robot.

El axis establece la escala de los ejes con diferentes resultados efectuando un procesamiento incluido dentro de la interfaz gráfica, se utiliza para el análisis de identificación de los patrones definidos para el robot, la carga y el sitio de descarga. Entiéndase por patrón para este proyecto figuras geométricas definidas para la localización de los tres puntos anteriormente mencionados.