

**DISEÑO Y CONSTRUCCIÓN DE VEHÍCULOS AUTÓNOMOS PARA WAREHOUSING**  
**LAB. DE LA UNIVERSIDAD EAFIT**

**EDWIN FERNANDO GIRALDO ARISTIZÁBAL**

**Proyecto presentado para obtener el título de Magister en Ingeniería**

**Área de concentración: Logística**

**Asesor: MSc. Martín Alonso Tamayo Vélez - Universidad Eafit**

**UNIVERSIDAD EAFIT**

**MAESTRÍA EN INGENIERÍA**

**MEDELLÍN**

**2021**

## CONTENIDO

	Pág.
<b>1. MOTIVACION DEL PROYECTO .....</b>	<b>10</b>
<b>2. OBJETIVOS DEL PROYECTO.....</b>	<b>12</b>
2.1 OBJETIVO GENERAL.....	12
2.2 OBJETIVOS ESPECIFICOS .....	12
<b>3. ESTADO DEL ARTE .....</b>	<b>13</b>
3.1 AGV (AUTOMATIC GUIDED VEHICLE) .....	13
3.2 SISTEMAS DE NAVEGACIÓN DE AGVS.....	14
3.3 SISTEMAS DE SEGURIDAD .....	15
3.4 SISTEMA DE COMUNICACIÓN .....	17
3.5 ROBOTS AUTONOMOS PARA PICKING .....	17
3.6 AMAZON KIVA ROBOTS (KIVA SYSTEMS) .....	18
3.7 BUTLER (GREY ORANGE) .....	19
3.8 EFFIBOT (EFFIDENCE).....	20
3.9 LOCUSBOT™ (LOCUS ROBOTICS).....	21
3.10 I AM SWIFT™ (I AM ROBOTICS).....	23
3.11 PLATAFORMA MÓVIL (BLEUM) .....	26
3.12 AGV WEASEL® (SSI SCHAEFER).....	27
3.13 2PICK® (MOTUM).....	30
<b>4. DESARROLLO DEL PROYECTO .....</b>	<b>32</b>
4.1 SELECCIÓN DEL TIPO DE COMUNICACION .....	32
4.2 SELECCIÓN TIPO DE CONTROL.....	35
4.3 PLANOS MECANICOS .....	45
4.3.1 Plataforma .....	45
4.3.2 Armazón de carga .....	47
4.3.3 Acople entre el armazón de carga y plataforma .....	49
4.3.4 Estructura de rotación ultrasonido .....	50
4.4 PLANOS CIRCUITOS IMPRESOS .....	51

4.5 COMPONENTES DEL VEHÍCULO AUTONOMO .....	55
4.6 PROGRAMAS EN ARDUINO Y LABVIEW .....	58
4.7 PLANOS DE CONEXIÓN DEL VEHICULO.....	73
4.8 MANUAL DE USUARIO .....	76
4.9 FALLAS Y FORMAS DE SOLUCIONARLAS .....	80
<b>5. CONCLUSIONES Y SUGERENCIAS PARA TRABAJOS FUTUROS .....</b>	<b>82</b>
5.1 CONCLUSIONES.....	82
5.2 SUGERENCIAS PARA TRABAJOS FUTUROS .....	83
<b>BIBLIOGRAFÍA .....</b>	<b>84</b>

## LISTA DE TABLAS

	Pág.
Tabla 1. Características vehículo autónomo LocusBot <sup>TM</sup> (Locus Robotics) .....	22
Tabla 2. Especificaciones técnicas Vehículo I am SWIFT <sup>TM</sup> (I am Robotics).....	25
Tabla 3. Características vehículo AGV WEASEL® (SSI SCHAEFER).....	29
Tabla 4. Explicación de la trama api numero 10 (envío de datos).....	34
Tabla 5. Identificación de Mensajes .....	71
Tabla 6. Intercambio de mensajes entre el software, master y el vehículo .....	72
Tabla 7. Algunas fallas y formas de solucionarlo .....	80

## LISTA DE ILUSTRACIONES

	Pág.
Ilustración 1. Vehículo Betty-Bot Amazon .....	18
Ilustración 2. Características Betty-Bot Amazon .....	19
Ilustración 3. Vehículo autónomo BUTLER (Grey Orange) .....	20
Ilustración 4. Vehículo autónomo EffiBOT (Effidence) .....	21
Ilustración 5. Vehículo autónomo LocusBot <sup>TM</sup> (Locus Robotics) .....	22
Ilustración 6. Vehículo I am SWIFT <sup>TM</sup> (I am Robotics) .....	24
Ilustración 7. Vista general vehículo I am SWIFT <sup>TM</sup> (I am Robotics) .....	24
Ilustración 8. Plataforma móvil (Bleum) .....	26
Ilustración 9. Vista general AGV WEASEL® (SSI SCHAEFER) .....	28
Ilustración 10. Vehículo AGV WEASEL® (SSI SCHAEFER) .....	28
Ilustración 11. Vehículo 2Pick ® (Motum) .....	31
Ilustración 12. Vehículo 2Pick ® (Motum) .....	31
Ilustración 13. Diagrama de las fases del proyecto .....	32
Ilustración 14. Estructura general de una trama api .....	33
Ilustración 15. Composición de la trama api numero 10 .....	34
Ilustración 16. Diagrama de control PID vehículo .....	35
Ilustración 17. Base de Datos del Sistema .....	37
Ilustración 18. Diagrama de Bloques, Control con Ganancia Programable .....	38
Ilustración 19. Gráfica de la Base de Datos, Curva de Reacción del Sistema .....	39
Ilustración 20. Validación del Sistema de Primer Orden con Retardo .....	40
Ilustración 21. Diagrama de Bloques del Sistema, Control PID Motores Derecho	41
Ilustración 22. Respuesta del Sistema bajo Control PID, Motores Derecho .....	41
Ilustración 23. Diagrama de Bloques del Sistema, Control PID Motores Izquierdo .....	42
Ilustración 24. Respuesta del Sistema bajo Control PID, Motores Lado Izquierdo .....	42
Ilustración 25. Respuesta del Sistema bajo Control PID Ajustado, Motores Izquierdo .....	44
Ilustración 26. Vista trasera del vehículo .....	46

Ilustración 27. Vista delantera del vehículo .....	46
Ilustración 28. Vista inferior del vehículo .....	46
Ilustración 29. Plano armazón de carga .....	48
Ilustración 30. Vista general del vehículo .....	49
Ilustración 31. Vista articulación sensor ultrasonido .....	50
Ilustración 32. Esquemático del circuito arduino DUE .....	52
Ilustración 33. Board del circuito arduino DUE .....	53
Ilustración 34. Esquemático del circuito impreso arduino nano .....	54
Ilustración 35. Board del circuito impreso arduino nano .....	54
Ilustración 36. Identificación de partes del circuito del vehículo autónomo .....	55
Ilustración 37. Identificación de partes del vehículo autónomo .....	56
Ilustración 38. Identificación de parte delantera del vehículo autónomo .....	56
Ilustración 39. Conexiones externas al circuito impreso de interfaz de arduino ....	57
Ilustración 40. Máquina de estado finito 1 programa en labView .....	64
Ilustración 41. Máquina de estado finito 1 programa en labView .....	66
Ilustración 42. Maquina de estado finito 2 programa en labView .....	66
Ilustración 43. Maquina de estado finito 2 programa en labView .....	69
Ilustración 44a. Plano de conexiones vehículo autónomo 1 de 3.....	73
Ilustración 44b. Plano de conexiones vehículo autónomo 2 de 3.....	74
Ilustración 44c. Plano de conexiones vehículo autónomo 3 de 3.....	75
Ilustración 45. Descripción de partes del vehículo .....	76
Ilustración 46. Descripción parte trasera del vehículo .....	77
Ilustración 47. Parte delantera del master .....	78
Ilustración 48. Descripción panel frontal software de operación.....	79

## RESUMEN

Una de las tendencias en la actualidad es la que se ha denominado “Industria 4.0”, la cual busca, entre otras, la interacción masiva de sistemas y dispositivos digitales. Una de las aplicaciones es la utilización de vehículos autónomos que buscan incrementar la eficiencia, velocidad, confiabilidad y precisión en los procesos.

La universidad EAFIT, ha diseñado y construido un laboratorio para logística de almacenamiento que se llamado “Warehousing Lab.” en este se explican los diferentes sistemas de picking que existen comercialmente, mostrando las ventajas y desventajas de cada uno, además se realizan investigaciones de alternativas que puedan mejorar estos procesos logísticos.

Los vehículos autónomos son unos de los artefactos que muchas empresas han implementado, para disminuir costos, optimizar los recursos y aumentar la productividad en los procesos industriales y logísticos; y de esta forma han mejorado de manera significativa sus indicadores, ya que estos vehículos traen consigo una alta cantidad de beneficios.

Este trabajo muestra todas las etapas que se llevaron a cabo para el diseño y construcción de los vehículos autónomos para el Warehousing Lab de la Universidad EAFIT, en este proceso se realizaron actividades de levantamiento de planos mecánicos, eléctricos y electrónicos, tarjetas electrónicas, documentación y desarrollo de aplicaciones en las plataformas Labview y Arduino.

Se pretende que este informe sirva además como manual del equipo de tal manera que se pueda reproducir, mejorar o hacer mantenimiento correctivo y/o preventivo.

**Palabras Claves:** Vehículo autónomo, picking, labview, arduino, xbee, tramas api, comunicación serial, IoT, logística.

## ABSTRACT

Currently one of the trends in industries 4.0 is autonomous vehicles that increase efficiency, speed, reliability, and accuracy of processes.

The EAFIT university has a Warehousing laboratory. In it explains different picking up systems that commercially exist, showing up the advantages and disadvantages of them, as well as alternative investigations improving them.

Autonomous vehicles are one of the artifacts that many companies have implemented, to reduce costs, optimize resources and increase productivity in industrial and logistics processes; in this way, they have improved their indicators, since these vehicles bring with them a big amount of benefits.

This work shows all carried stages for construction and design of autonomous vehicles for EAFIT's Warehousing Lab and mechanical, electrical and electronic plans, electronic cards, Labview software documentation, and Arduino.

This report is intended to serve as a team manual. This manual could be reproduced, improved or made corrective and/or preventive maintenance.

**Keywords:** Autonomous vehicle, picking up, LabVIEW, Arduino, XBee, API frames, serial communication.

## 1. MOTIVACION DEL PROYECTO

La Universidad EAFIT cuenta con un laboratorio de almacenamiento (Warehousing Lab.) en el cual se analizan los diferentes sistemas de picking que existen comercialmente, mostrando las ventajas y desventajas de cada uno, además se investiga sobre alternativas que puedan mejorar estos procesos.

Este proyecto busca dotar al laboratorio con dos vehículos autónomos para realizar labores de picking asistido. Adquirir estos vehículos en el mercado tiene varias desventajas:

- El alto costo.
- Los vehículos comerciales de este tipo tienen tecnología cerrada y no permiten ajustarse fácilmente a las necesidades específicas de las empresas de nuestro medio, además que son de gran tamaño y no se acomodan al espacio del Warehousing de la universidad EAFIT.
- No permiten la apropiación de la tecnología con el fin de adaptarlo y reproducirlo en diferentes áreas.

Teniendo en cuenta lo anterior, se plantea la opción de diseñar, programar y poner en funcionamiento dos vehículos autónomos como proyecto de maestría, todo esto a través de una inversión de la universidad gestionada a través de una orden de trabajo del Centro de laboratorios.

Las órdenes de trabajo en el Centro de laboratorios de la universidad EAFIT se manejan como un proyecto, tienen unos fondos asignados y debe producir un equipo que se pueda activar en el inventario de la universidad.

## **2. OBJETIVOS DEL PROYECTO**

### **2.1 OBJETIVO GENERAL**

Desarrollar el sistema de control de navegación (con un sistema de posicionamiento) y la construcción de dos vehículos autónomos, para el almacén (Warehousing Lab) de la Universidad EAFIT, de tal manera que el sistema permita que ambos vehículos realicen su labor de manera simultánea o individual.

### **2.2 OBJETIVOS ESPECIFICOS**

- Diseñar y construir dos vehículos autónomos, con el presupuesto asignado.
- Implementar el protocolo de comunicación más adecuado entre el sistema de gestión del laboratorio y los vehículos.
- Implementar la comunicación entre los vehículos y el sistema de control de forma inalámbrica.
- Desarrollar el programa de control con LabView, que permita leer y almacenar el pedido de picking en archivos de texto.
- Desarrollar la documentación necesaria para la operación y reproducción de este sistema.

### 3. ESTADO DEL ARTE

#### 3.1 AGV (AUTOMATIC GUIDED VEHICLE)

Los Automatic Guided Vehicles (AGV), son vehículos de guiado automático o vehículos que no requieren conductor para llevar a cabo el transporte de mercancía en instalaciones.

Existen diversos tipos de vehículos de conducción autónomos, como apiladores automáticos, carretillas elevadoras, transpaletas, plataformas y robots colaborativos autónomos de transporte de bastidores pequeños como Kiva y Fetch.

La implementación de este tipo de vehículos provee ventajas como las siguientes:

- **Seguridad:** Evitan que los empleados del almacén realicen tareas que pueden causar lesiones o agotamiento físico, además del daño del producto. Pueden ser designados caminos seguros. Incluso si el camino es muy usado, el sensor AGV en el vehículo lo dirige a disminuir la velocidad o detenerse, ya que detecta su proximidad a objetos o personas.
- **Flexibilidad:** los vehículos autónomos se pueden integrar con sistemas de gestión de almacenes, sistemas automatizados de almacenamiento y recuperación, transportadores, etc. Además, su movilidad evita que sean un obstáculo permanente.
- **Costos reducidos:** puede reducir los costos de mano de obra con vehículos autónomos en el almacén. Pueden trabajar 24/7 y bajo demanda, eliminando

así la necesidad de capacitación y agotamiento en el caso de los seres humanos. Además, si bien proporcionan una velocidad constante, requieren un bajo mantenimiento.

- **Mayor productividad y precisión:** La capacidad de integración con AS/RS (Automated Storage and Retrieval System) y WMS (Warehouse Management System) y la tecnología de sensores hace que estos vehículos sean precisos. Trabajan a una velocidad constante, lo que hace que el manejo del material sea eficiente en el tiempo, consistente y menos propenso a errores.

### 3.2 SISTEMAS DE NAVEGACIÓN DE AGVS

Módulo responsable de posicionar el vehículo dentro de las rutas determinadas para llevar a cabo su función.

- **Filoguiado:** utilización de un cable enterrado en el suelo para marcar la ruta a seguir.
- **Magnético:** utilización de banda o tags magnéticos en el trazado de rutas.
- **Óptico:** trazado de rutas por banda pintada gracias a la instalación de una cámara a bordo del AGV.
- **Láser:** sistema de guiado por láser que mediante la instalación de reflectores en planta en lugares estratégicos se garantiza que en todos los puntos del recorrido se detecten tres reflejos del haz de luz emitido por el láser de tal forma que se conoce con precisión milimétrica su posición exacta.

- **Natural / por contornos:** utilización de la disposición de la planta para el diseño de mapas virtuales y el trazado de rutas.
- **Combinados:** porque hay situaciones en las que los recorridos atraviesan zonas muy diversas es necesario combinar distintos sistemas de navegación con el fin de proporcionar en todo momento el sistema de navegación óptimo para cada tramo. Estos sistemas pueden funcionar de manera conjunta en forma de sistemas duales de guiado, entrando a funcionar alternativamente en función del tramo de ruta que esté atravesando el vehículo.

### 3.3 SISTEMAS DE SEGURIDAD

Son los elementos que garantizan que los vehículos convivan de manera segura con el resto de elementos que les rodean.

- **Láser de seguridad:** dispositivo capaz de detectar obstáculos reduciendo la velocidad de la marcha del vehículo hasta su completa detección una vez alcanzada la distancia mínima de seguridad totalmente programable y configurable a las necesidades de cada proyecto.
- **Fotocelulas + bumper:** sistemas activos de seguridad que permiten la recogida segura de la carga. Instalados en el perímetro del vehículo pueden funcionar de sistema de seguridad que detiene el vehículo al detectar un obstáculo, los bumpers a su vez son sistemas de seguridad que se activan deteniendo al vehículo al contactar con un obstáculo.

- **Sistema de aviso acústico y luminoso:** sistema que mediante el uso de señales acústicas y luminosas informa al resto de usuarios en planta de la presencia del vehículo automático.
- **Parada de emergencia:** botón de parada del sistema que detiene el vehículo tras su activación ante un peligro inminente.
- **Sistema de baterías y recarga:** Sistema que garantiza que los vehículos cuenten con autonomía para moverse.

Los sistemas de AGVs se equipan con todo tipo de baterías:

- Ion litio
- Gel
- Plomo ácido

### 3.4 SISTEMA DE COMUNICACIÓN

Sistema que permite la comunicación entre los elementos en planta, tanto con software como entre máquinas. Recoge los datos y los gestiona para la generación de órdenes y el control de tráfico.

- **Sistema de captura de datos:** sistema que permite la recogida de información sobre el estado de los componentes en la instalación.
- **Host:** elemento central del sistema de comunicación. Recoge todas las señales que se generan en planta y genera las órdenes que ejecutarán los AGVs.
- **Sistema de gestión de tráfico:** sistema que gestiona las paradas de vehículos en aquellas instalaciones equipadas con más de un AGV. Este sistema evita paradas innecesarias contribuyendo a la optimización de la productividad.

### 3.5 ROBOTS AUTONOMOS PARA PICKING

Los robots autónomos son un tipo de AGVs que son parte de un sistema modular de picking de mercancía a persona que también incluye estanterías móviles para almacenamiento y puestos de preparación de pedidos. Cada robot puede mover una carga de aproximadamente 160 a 1.500 kg. De peso, llevándola a un puesto donde un trabajador gestiona hasta 48 pedidos simultáneamente. Todo el proceso está controlado por el sistema de gestión de almacenes y ayuda a los trabajadores

a minimizar el tiempo en distancias recorridas y los errores manuales de selección de productos.

### 3.6 AMAZON KIVA ROBOTS (KIVA SYSTEMS)

- Robots autónomos de transporte de bastidores.
- Modelos con capacidades de carga de hasta 250 lb, 1.000 lb y 3.000 lb
- Velocidad: 4 - 6 km/h

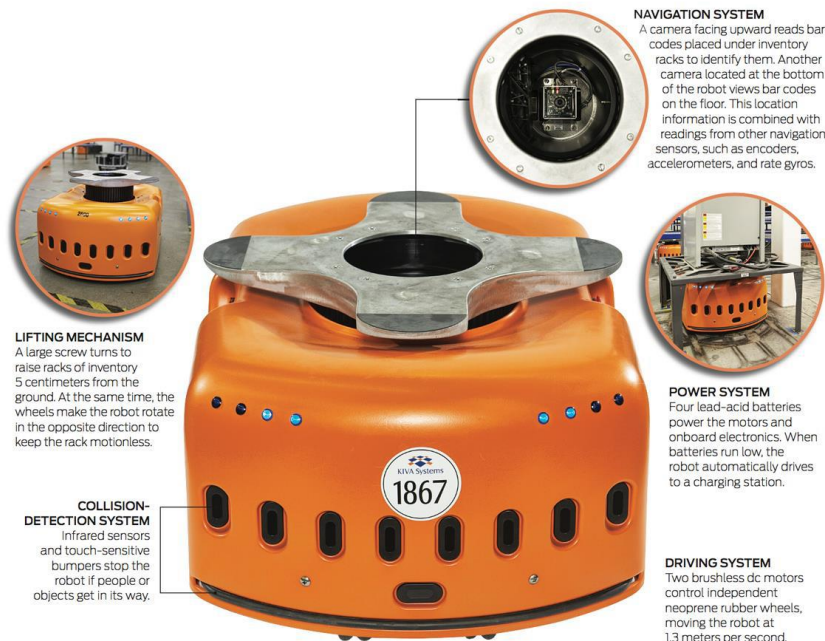
Ilustración 1. Vehículo Betty-Bot Amazon



✓ efectividad en 99.99 %

Fuente: (Polo, 2016)

## Ilustración 2. Características Betty-Bot Amazon



Fuente: (Brown, 2016, p. 12)

### 3.7 BUTLER (GREY ORANGE)

Grey Orange Butler es una solución robótica de bienes a persona para el almacenamiento y reabastecimiento de inventario automatizado, y para recoger y combinar pedidos en centros de distribución y cumplimiento. Habilitado por Inteligencia Artificial, su software se adapta en tiempo real a los perfiles de inventario cambiantes y a los patrones y picos de cumplimiento de pedidos, lo que resulta en una mayor productividad y precisión. Su arquitectura de sistema modular proporciona flexibilidad y escalabilidad para aumentar el volumen de

almacenamiento o el rendimiento de cumplimiento al agregar más bastidores o robots según sea necesario.

Los sistemas Grey Orange Butler se implementan en Hong Kong y la India para el comercio minorista y el comercio electrónico, en América del Sur y en Japón con Nitori Group, la cadena de muebles y muebles para el hogar más grande.

- Modelos con capacidades de carga de 600kg y 1400 kg.

### **Ilustración 3. Vehículo autónomo BUTLER (Grey Orange)**



Fuente: (The Standard Cio, 2017).

### **3.8 EFFIBOT (EFFIDENCE)**

Carretilla automatizada para la recolección y el traslado de materiales, que sigue a los operadores a lo largo del almacén y se ocupa de la mayor parte del trabajo físico.

La ventaja para el operario de picking es la opción de trabajar con las manos libres y no tener que empujar o tirar de carretillas pesadas.

- Capacidad de carga 300 Kg y velocidad 7 km/h

#### Ilustración 4. Vehículo autónomo EffiBOT (Effidence)



Fuente: (Institution of Mechanical Engineers, 2016).

### 3.9 LOCUSBOT™ (LOCUS ROBOTICS)

Se mueven de manera autónoma hacia las estaciones donde están los trabajadores, minimizando el desplazamiento de los trabajadores y eliminando el trabajo pesado.

LocusBots aprende automáticamente las rutas de viaje más eficientes a través de su almacén, para mejorar drásticamente la eficiencia y la productividad de los trabajadores sobre la selección de carros tradicional.

El diseño flexible de LocusBot permite que se configure para satisfacer una amplia gama de necesidades de recolección de contenedores y contenedores múltiples, que se adapta prácticamente a cualquier requisito de cumplimiento. Su diseño liviano les permite operar de manera segura cerca de los trabajadores.

**Tabla 1. Características vehículo autónomo LocusBot™ (Locus Robotics)**

### LocusBot Especificaciones

Peso	100 lbs (45kg) (descargado)
Carga útil	40 - 100 lbs (18 - 45kg)
Velocidad de viaje	1.4 metros / s
Cargando	110 o 220V
Operación	14 horas / carga
Recargar	60 minutos

Fuente: (Locus Robotics, 2018).

**Ilustración 5. Vehículo autónomo LocusBot™ (Locus Robotics)**



Fuente: (Locus Robotics, 2018).

### **3.10 I AM SWIFT™ (I AM ROBOTICS)**

Swift es un robot autónomo de recogida de materiales, que ofrece características como ninguna otra. Con la tecnología integrada de detección de obstáculos, Swift navega por múltiples pasillos para moverse con seguridad y precisión. Swift es capaz de seleccionar y transportar productos a velocidades humanas y le permite escalar las operaciones de manera más rentable. Swift puede trabajar solo o simultáneamente como una flota.

Cuenta con un brazo articulado que le permite recoger productos de manera rápida y precisa desde ambos lados del pasillo, en estantes desde el nivel del piso hasta siete pies de altura. Este brazo cuenta con tecnología RapidVision que permite a los robots ver y localizar objetos en 3D y en tiempo real. Usando sus sensores integrados, el robot reconoce los objetos basándose en la información que recopiló del escáner del producto Flash.

Swift tiene la capacidad de usar múltiples efectores finales para agarrar objetos de hasta 15 libras. El efector final más común es la pinza de vacío, que puede seleccionar una amplia gama de productos de consumo y SKU. El efector final también se puede cambiar para recuperar cualquier contenedor y convertir su infraestructura existente en un sistema automatizado de almacenamiento y recuperación.

Especificaciones del producto: Estante de trabajo superior de extensión máxima 85 in, carga útil máxima del carro 50lbs, peso máximo de picking 15 lbs

**Ilustración 6. Vehículo I am SWIFT™ (I am Robotics)**



Fuente: (IAM Robotics, 2017).

**Ilustración 7. Vista general vehículo I am SWIFT™ (I am Robotics)**



Fuente: (IAM Robotics, 2019).

**Tabla 2. Especificaciones técnicas Vehículo I am SWIFT™ (I am Robotics)**

TALLA Y PESO	DIMENSIONES	43.6 pulgadas x 74 pulgadas x 27.7 pulgadas (110.8 cm x 188.8 cm x 70.4 cm) 650 libras (294.87 kg)
VELOCIDAD Y RENDIMIENTO	ALTURA DE RECOGIDA DE ESTANTE SUPERIOR	84 en
	ALTURA DE RECOGIDA DE ESTANTE INFERIOR	2 en
	PICKING PICKLOAD CON EFECTOR ESTÁNDAR	Hasta 15 libras
	MODELO DE BRAZO	Fanuc 200 iD 7L
	ASCENSOR INTEGRADO	100 libras
	SUSPENSIÓN	Pasivo
	EXACTITUD POSICIONAL	+ / - 20 mm (0.78 in)
	RADIO DE GIRO	0 mm
	SENSORES DE SEGURIDAD	Detección de obstáculos hacia adelante y hacia atrás - 9 pies
BATERÍA Y SISTEMA DE ALIMENTACIÓN	TECNOLOGIA DE BATERIA	Fosfato de litio y hierro (LiFePO4, también llamado LFP)
	DURACIÓN DE LA BATERÍA	Supera las 10 horas por carga, 3000 ciclos de carga.
	BATERÍA INTERCAMBIABLE RÁPIDA	Soporte para turnos múltiples, 2 baterías buenas para 3 turnos
	TIEMPO DE CARGA	7 horas
	VOLTAJE Y CAPACIDAD DE LA BATERÍA	55 V / 100 Ah
COMUNICACIÓN DE INTERFAZ	ACCESORIOS	Escáner de flash, SwiftLink Server, SwiftLink Viewer, Segunda batería, Servicio al cliente 24/7, Estación de carga, Controlador manual de la unidad
	INTERFAZ DE PANTALLA TÁCTIL	Interfaz de usuario LCD
	COMUNICACIÓN	Wi-Fi 802.11 ac / a / b / g / n 2.4 GHz, 5 GHz
	LA SEGURIDAD	4 botones de parada de emergencia y sensores que detienen la producción cerca de la interacción humana.

Fuente: (IAM Robotics, 2017).

### 3.11 PLATAFORMA MÓVIL (BLEUM)

Cuenta con motor de accionamiento para movimiento y un motor para accionar el mecanismo de elevación y un motor para manejar la rotación contraria cuando el robot necesita girar, pero no la plataforma.

Por seguridad, la parte frontal del robot tiene un sensor ultrasónico que lo detendrá si hay un obstáculo en el camino e informará que continúe una vez que el obstáculo haya desaparecido.

El sistema también tiene dos cámaras. Uno que apunta hacia abajo para propósitos de navegación y otro que apunta hacia arriba para asegurar que el robot esté levantando y moviendo el estante deseado.

#### Ilustración 8. Plataforma móvil (Bleum)



*Robot de servicio regular*

Peso máximo de transporte	1, 100 lbs. o 500kg
Máxima velocidad	5 pies / seg o 1.5 M / seg
Duración de la batería a la carga máxima	8 horas

*Robot de trabajo pesado*

Peso máximo de transporte	2,200 lbs. o 1,000 KG
Máxima velocidad	3 pies / seg o 1 M / seg
Duración de la batería a la carga máxima	6.5 horas

Fuente: (Bleum, 2017).

### **3.12 AGV WEASEL® (SSI SCHAEFER)**

Estos AGVs pueden equiparse para operación manual, semiautomática o totalmente automatizada, según sus requisitos. Los empleados pueden suministrar una estación de transferencia manual por adelantado, por ejemplo. El controlador de la flota se encarga de enviar el WEASEL ® a la estación correspondiente.

El WEASEL ® no requiere numerosos sensores o un sistema de control complejo, sino que navega a lo largo de un carril designado ópticamente. Este carril se puede aplicar rápida y fácilmente, y también puede modificarse en cualquier momento para cumplir con sus requisitos.

El software de control estandarizado se puede instalar rápida y fácilmente, lo que permite que el sistema se pueda ampliar o reducir a corto plazo. Los pedidos de transporte se generan mediante comandos de radio activados manualmente (solución independiente), sistemas de terceros como máquinas de producción con un PLC o su sistema de flujo de material. Un controlador de flota gestiona los pedidos y los asigna a los vehículos correspondientes.

La energía se suministra a través de paquetes de baterías sin mantenimiento que se pueden recargar manualmente en un gabinete de carga automáticamente a través de los contactos del piso. El COMADREJA ® puede lograr tiempos de hasta 16 horas de funcionamiento con una carga de batería.

**Ilustración 9. Vista general AGV WEASEL® (SSI SCHAEFER)**



Fuente: (SSI Schaefer, 2017).

**Ilustración 10. Vehículo AGV WEASEL® (SSI SCHAEFER)**



Fuente: (SSI Schaefer, 2017).

**Tabla 3. Características vehículo AGV WEASEL® (SSI SCHAEFER)**

<b>Key system figures</b>	
Transport weight	up to 35 kg
Vehicle dimensions	up to 810 mm x 420 mm x 180 mm (l x w x h)
Acceleration/braking	up to 2 m/s <sup>2</sup>
Turning radius	min. 600 mm
Traveling speed	up to 1 m/s
Data transfer between the fleet controller and WEASEL occurs via WLAN (5 Ghz).	
<b>Environment</b>	
Incline climbing ability	loaded: 10 % empty: 20 %
Ambient temperature range	2 - 50 °C, non-condensing
<b>Energy supply</b>	
Manual via charging cabinet or automatic via ground-contact charging stations	
Power supply	battery 24 V
Charging time	8 hours
Operating time	up to 16 hours
<b>Lift</b>	
Lift speed	up to 3 m/s
Number of entries/exits per lift	up to 10
Height of the lift	up to 13.4 m

Fuente: (SSI Schaefer, 2017).

### **3.13 2PICK ® (MOTUM)**

Este AGV transporta contenedores, paletas y otros transportadores de carga para recibir los artículos seleccionados. 2Pick ® navega de forma autónoma a las distintas posiciones del almacén y espera allí los artículos seleccionados. Una vez finalizado el pedido, los portadores de carga se conmutan automáticamente. Los procedimientos sin papel, las bajas tasas de error en la selección y las rutas optimizadas son solo algunas de las ventajas de este proceso semiautomático con soporte de AGV al compararlo con la selección manual.

Utilizan baterías de litio 2Lite. Como resultado, 2Pick ® puede funcionar las 24 horas del día sin ningún problema.

Un controlador de flota gestiona y optimiza todos los pedidos de transporte. El sistema puede recibir los pedidos de cada sistema de gestión de almacén.

## Ilustración 11. Vehículo 2Pick® (Motum)

### 2Pick pp (Piece Picking)



#### Technical Information

■ Load Handling	Special cantilever system
■ Max. Payload	200 kg total (cartons, totes)
■ Lift Height	130 mm
■ Overall Length	1,740 mm
■ Overall Width	975 mm
■ Aisle Width	1,350 mm
■ Max. Travel Speed	2.0 m/s
■ Acceleration	0.5 m/s <sup>2</sup>
■ Travel Drive	AC 4 kW
■ Lift Drive	AC 1.4 kW
■ Battery	24 V, AGM or 2Lite
■ Guidance	Laser navigation
■ Controls	MASS by MoTuM

Fuente: (SSI Schaefer, 2017, p. 12).

## Ilustración 12. Vehículo 2Pick® (Motum)

### 2Pick cp (Case Picking)



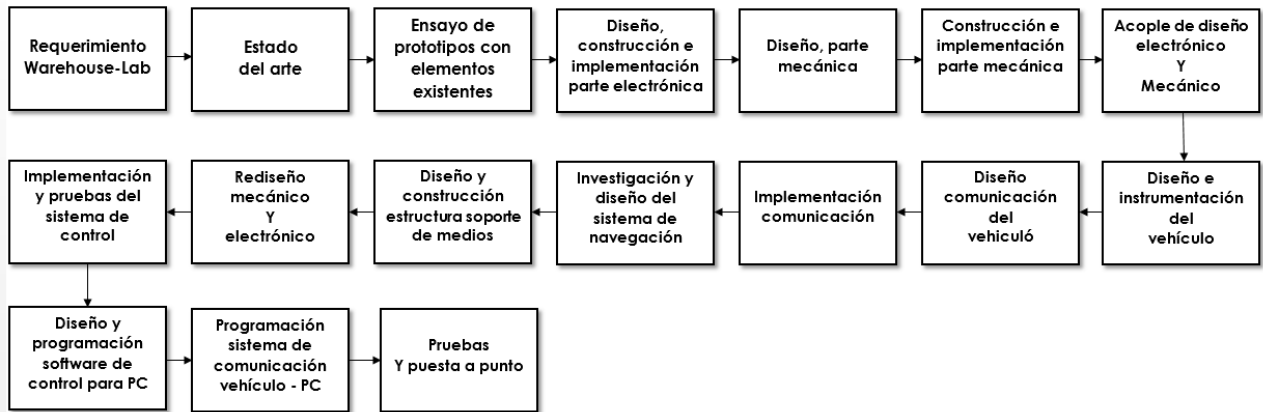
#### Technical Information

■ Load Handling	Fork
■ Max. Payload	2,000 kg
■ Lift Height	130 mm
■ Overall Length	3,350 mm
■ Overall Width	900 mm
■ Aisle Width	1,300 mm
■ Max. Travel Speed	2.0 m/s
■ Acceleration	0.5 m/s <sup>2</sup>
■ Travel Drive	AC 4 kW
■ Lift Drive	AC 1.4 kW
■ Battery	24 V, AGM or 2Lite
■ Guidance	Laser navigation
■ Controls	MASS by MoTuM

Fuente: (SSI Schaefer, 2017, p. 12).

## 4. DESARROLLO DEL PROYECTO

Ilustración 13. Diagrama de las fases del proyecto



Fuente: Elaboración propia.

### 4.1 SELECCIÓN DEL TIPO DE COMUNICACION

Después de analizar los diferentes tipos de comunicación inalámbrica disponibles en el mercado se definió usar el protocolo de comunicación Zigbee a través de tramas api, usando los dispositivos xbee.

Zigbee es un protocolo de comunicaciones inalámbrico basado en el estándar de comunicaciones para redes inalámbricas IEEE\_802.15.4. Creado por Zigbee Alliance.

Los xbee son dispositivos de bajo consumo que permiten comunicarse en forma inalámbrica usando comunicación serial y puede comunicarse en modo transparente o en modo api.

Con los dispositivos xbee se pueden formar redes de hasta 65535 dispositivos que se pueden conectar en red.

Las tramas api (Application Programming Interface) es una manera de transmitir los datos de una manera segura y estructurada. La información que envía y recibe cada xbee tiene que seguir una estructura específica y debe estar empaquetado en tramas (frames), que definen operaciones y eventos dentro del módulo.

La estructura de una trama api está compuesta de la siguiente manera:

#### Ilustración 14. Estructura general de una trama api



Fuente: Elaboración propia.

El delimitador es el byte de inicio, siempre comienza con 7E (en hexadecimal)

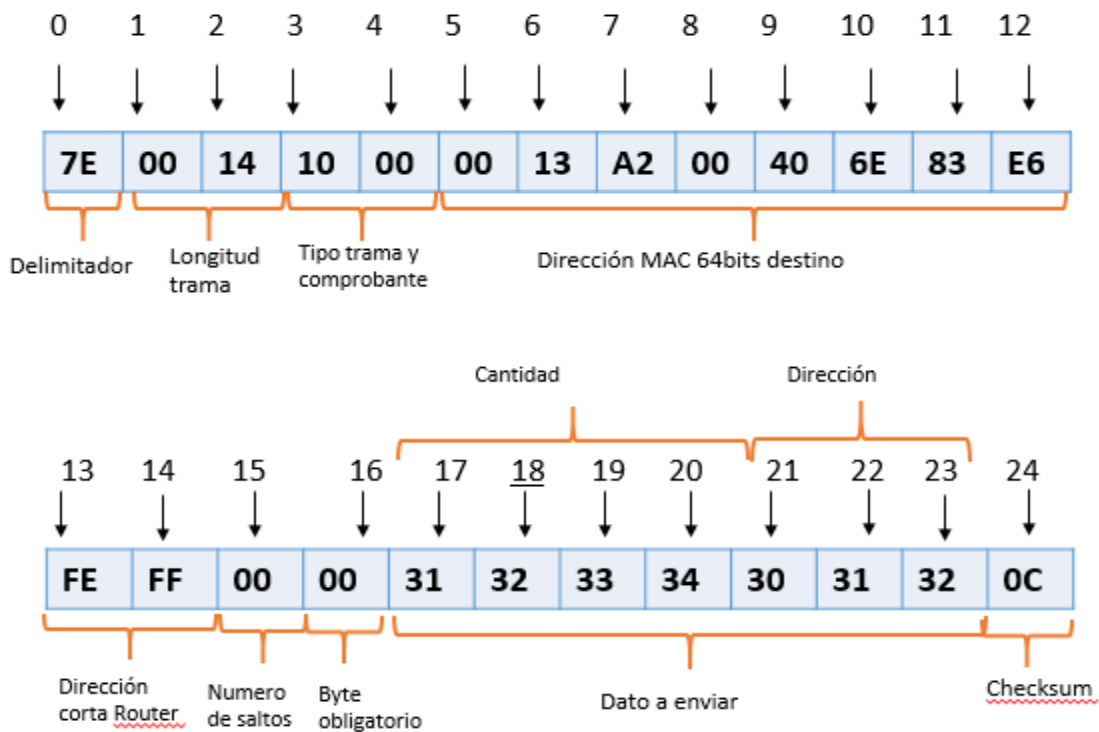
La longitud de la trama ocupa 2 bytes de la estructura. Indica cuantos bytes van a ser transmitidos.

La estructura API contiene la dirección mac del dispositivo destino, la información a transmitir y características de la comunicación.

El checksum es el byte de verificación de la trama enviada.

Las tramas enviadas en el proyecto corresponden a la trama numero 10 (Envío de datos) y está compuesta de la siguiente estructura:

### Ilustración 15. Composición de la trama api numero 10



Fuente: Elaboración propia.

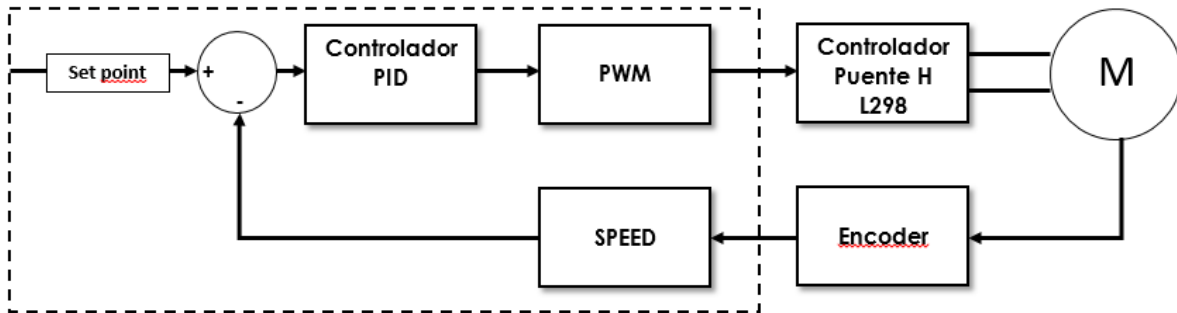
**Tabla 4. Explicación de la trama api numero 10 (envío de datos)**

Byte	Descripción
0	Delimitador de inicio de trama
1 y 2	Longitud de la trama, a partir del 10 y sin contar el checksum
3 y 4	Tipo de trama y configuración de confirmación. (00 no recibe confirmación)
5 al 12	Dirección mac del dispositivo destino
13 y 14	Dirección mac corta del dispositivo destino, si no se conoce poner FF FE
15	Numero de saltos desde el coordinador hasta el router
16	Byte obligatorio en ceros
17 al 23	Información enviada. En este caso cantidad y dirección
24	Byte de checksum (Verificación de la información enviada)

Fuente: Elaboración propia.

## 4.2 SELECCIÓN TIPO DE CONTROL

Ilustración 16. Diagrama de control PID vehículo



Fuente: Elaboración propia.

Para desarrollar este tipo de control se puede emplear tanto técnicas de control avanzadas como convencionales, en este caso particular y analizando entre diferentes opciones de controladores se decidió implementar un algoritmo PID. La razón por la que se decide realizar la técnica de control convencional PID es debido a que dicha técnica de control se puede establecer en sistemas de procesamiento que requiera de un alto procesamiento debido a las limitaciones del hardware utilizado, basados en la simulación del sistema se obtiene una respuesta deseada garantizando exactitud, velocidad de respuesta y estabilidad del sistema con los parámetros evaluados para controlar el sistema.

Se implementaron cuatro controladores PID para cada uno de los 4 motores que permiten el movimiento del carro diseñado para Warehouse Lab de la Universidad EAFIT, con el propósito de disminuir desgastes en ellos, y así, permitir que la vida útil de los servomotores logre cumplir el ciclo de vida para el cual fueron diseñados,

disminuyendo los costos por recambio de partes o componentes. Además, dichos algoritmos de control resultan ser muy efectivos debido a que son optimizados en relación con su desempeño y estrategia de control, basándose directamente en el comportamiento del sistema, analizando su respuesta dinámica en diferentes puntos o zonas de trabajo dentro del Warehouse Lab.

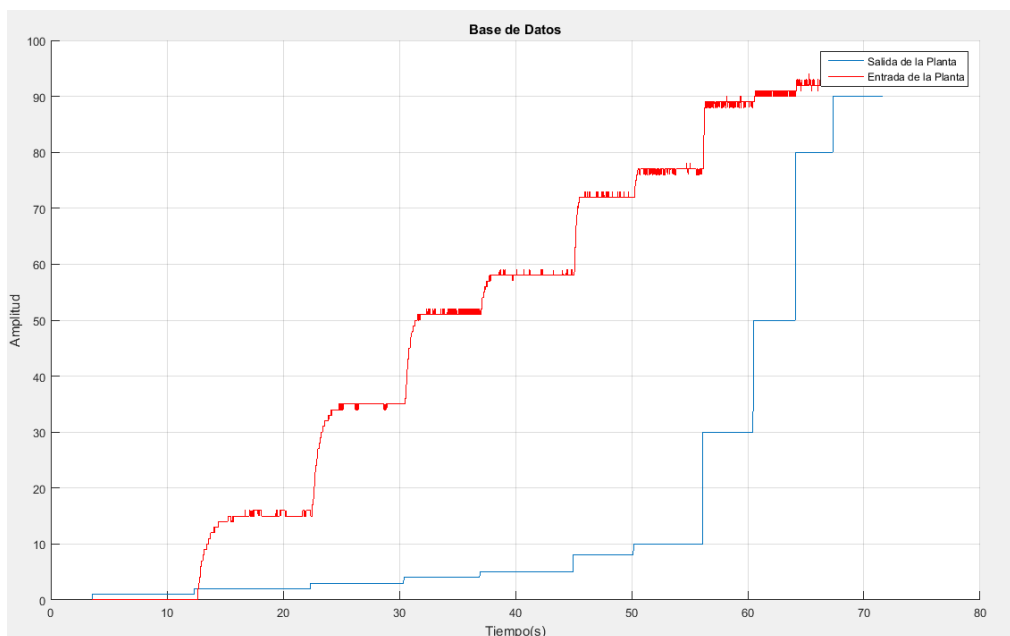
Con la implementación del control, se espera obtener una respuesta del sistema con un tiempo de establecimiento ( $t_s$ ) menor a 1 segundo, un sobre impulso ( $M_p$ ) menor o igual al 10% de la salida, y un error en estado estable ( $e_{ss}$ ) igual a cero (0) o lo más próximo a cero. Por otra parte, la salida de control se satura al 100%, tanto en la simulación como en la implementación, para así de esta forma evitar errores de procesamiento o desbordamiento de la información leída a cada instante de tiempo.

Los controladores con ganancia programable (Gain Scheduling), son utilizados en procesos donde se hace necesario estabilizar el sistema alrededor de un punto de operación seleccionado, teniendo presente que el sistema analizado, puede ser lineal o no lineal. Para controlar sistemas no lineales, es común el uso de controladores lineales previamente obtenidos mediante un punto de operación establecido para satisfacer el control requerido en diferentes puntos de operación del sistema.

Como el objetivo de estos controladores es mantener el control del sistema de una forma óptima, se realiza un diseño del controlador variando una variable que intervenga en el proceso y la cual se establece como variable de programación,

pues con dicha variable se obtienen diferentes controladores para cada punto de operación seleccionado, proporcionando un controlador diferente para cada punto del proceso analizado, lo que permite una mejor eficiencia del control sobre el proceso, garantizando un correcto funcionamiento del mismo, pues la gran ventaja de este tipo de control, es que el sistema es analizado en diferentes puntos de la salida y la función de transferencia se calcula para cada salida del sistema, lo que permite al controlador desarrollar un mejor desempeño del control de la variable en cada dinámica analizada e identificada.

### Ilustración 17. Base de Datos del Sistema

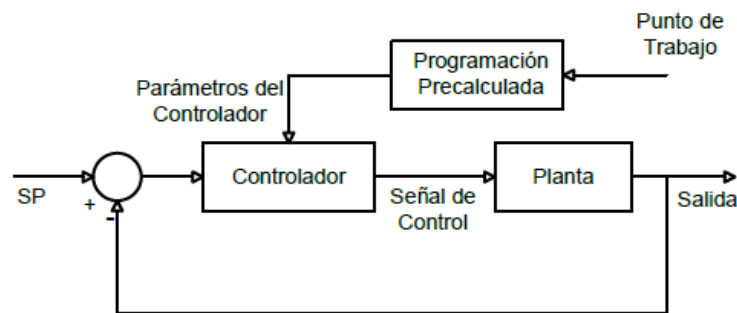


Fuente: Elaboración propia.

Así, en cada punto de operación seleccionado se realiza una linealización de la(s) variable(s) de interés, en este caso será la velocidad, obteniendo una gama de set-points para cada punto de operación seleccionado de acuerdo a la señal enviada por el seguidor de línea dispuesto en el vehículo.

En conclusión, los parámetros de los controladores se van ajustando al proceso mediante la variación de los valores que deben tomar las variables para cada punto de operación o de trabajo previamente determinado.

### Ilustración 18. Diagrama de Bloques, Control con Ganancia Programable



Fuente: Elaboración propia.

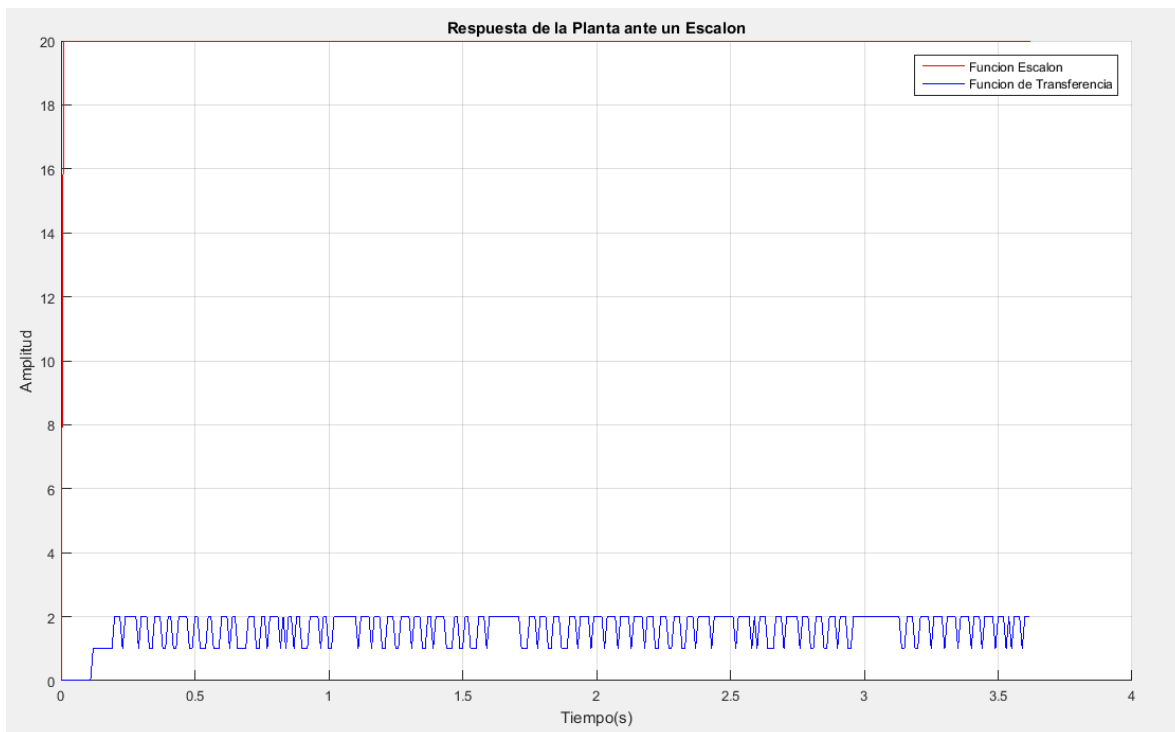
Para el diseño de este tipo de controladores no se tiene definido un orden estricto o estructura matemática a seguir; sin embargo, los pasos que se tuvieron en cuenta para el diseño de este controlador, fueron los siguientes:

- Determinar los valores de set-point según la dinámica del recorrido.
- Obtener el modelo del proceso para diferentes puntos de operación.
- Calcular los parámetros del controlador para la correcta operación.

Cuando se realizó la caracterización, se evaluaron las respuestas del sistema se en cada uno de los tramos de la pista ante diferentes estímulos, lo que permitió caracterizar el sistema y saber su respuesta dinámica en cada estado.

Los controladores implementados fueron Controladores PID, obtenidos mediante la curva de reacción de los Motores DC instalados en el vehículo, implementando el método de Ziegler-Nichols para tal fin.

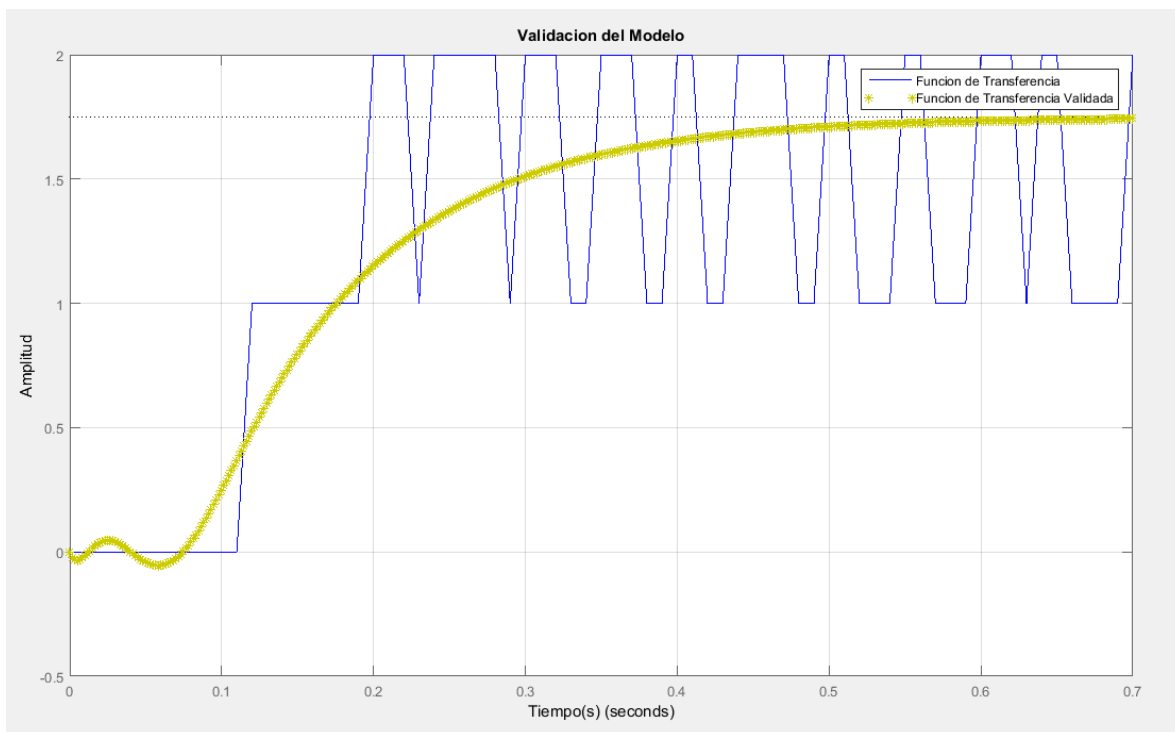
### Ilustración 19. Gráfica de la Base de Datos, Curva de Reacción del Sistema



Fuente: Elaboración propia.

De esta respuesta dinámica del sistema se realiza el modelamiento matemático establecido por Ziegler Nichols, y se obtiene una validación matemática del sistema, la cual arroja una función de transferencia de primer orden con retardo que valida la respuesta en el dominio del tiempo del sistema analizado.

### Ilustración 20. Validación del Sistema de Primer Orden con Retardo

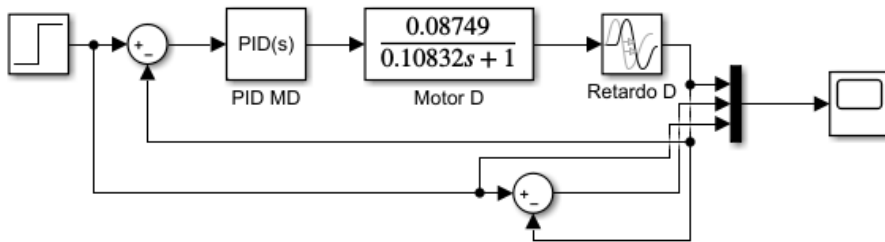


Fuente: Elaboración propia.

La función de transferencia validada de la curva de reacción del sistema obtenida es la siguiente:

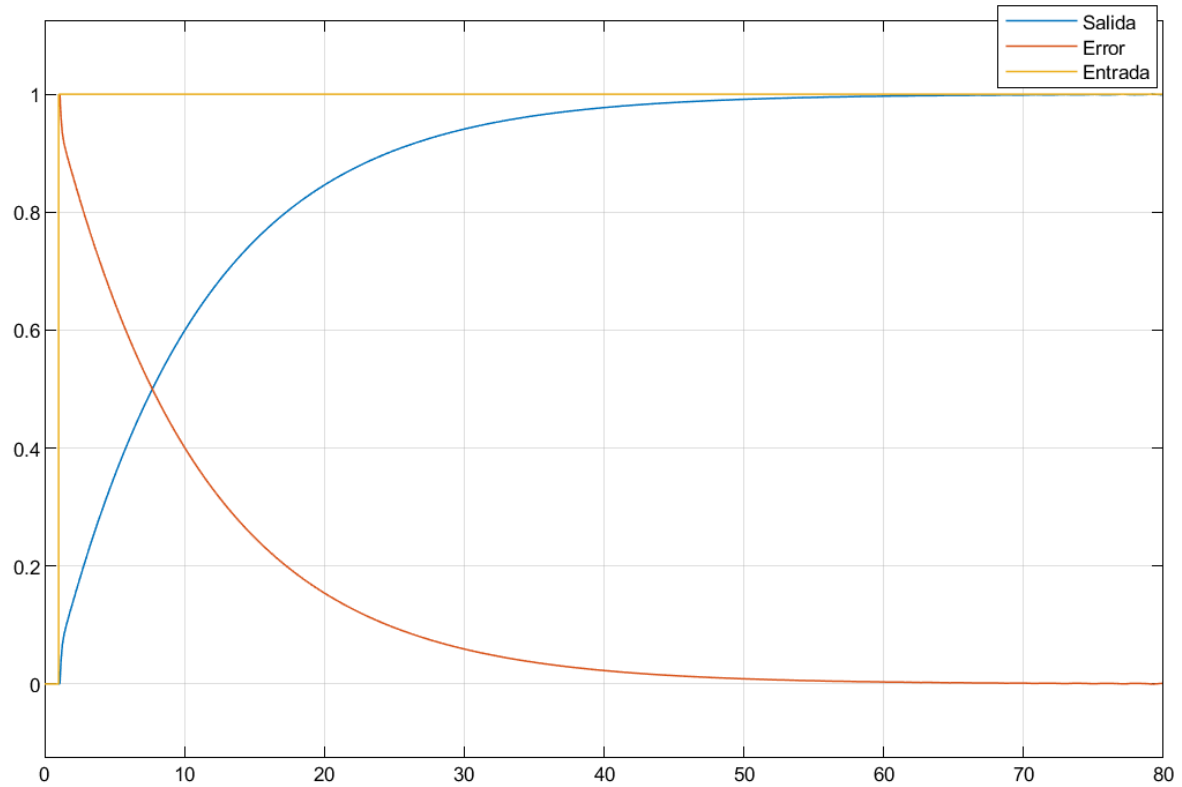
$$G(s) = \frac{0.087499e^{-0.08379s}}{0.108322s + 1}$$

**Ilustración 21. Diagrama de Bloques del Sistema, Control PID Motores Derecho**



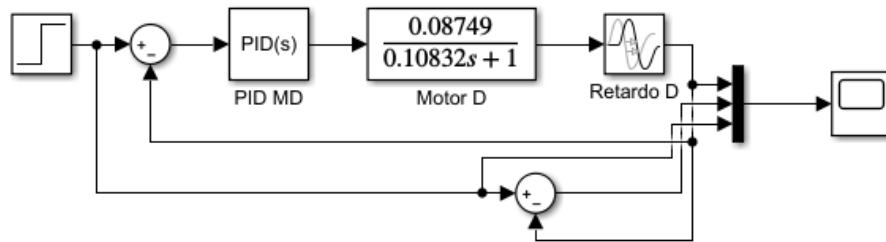
Fuente: Elaboración propia.

**Ilustración 22. Respuesta del Sistema bajo Control PID, Motores Derecho**



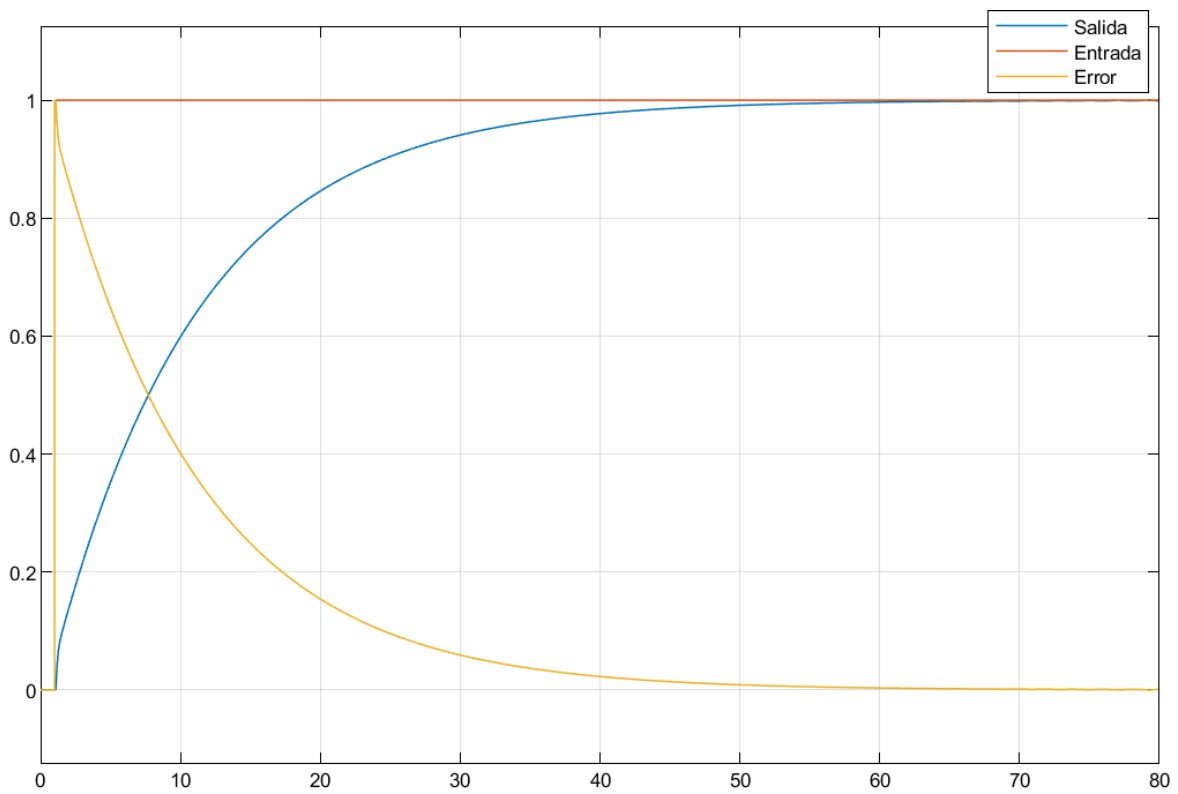
Fuente: Elaboración propia.

**Ilustración 23. Diagrama de Bloques del Sistema, Control PID Motores Izquierdo**



Fuente: Elaboración propia.

**Ilustración 24. Respuesta del Sistema bajo Control PID, Motores Lado Izquierdo**



Fuente: Elaboración propia.

Al momento de realizar la simulación del sistema, se logra observar que el tiempo muerto es despreciable, y la velocidad de respuesta del sistema, como es característico de los motores de DC, fue alta.

Se implementó el controlador en el proceso real, y al estar en ejecución, por las características mecánicas del sistema, en el desempeño de los motores de cada lado se generaba una alteración respecto a la estabilidad del mismo, debido a que la corrección que realizaba el PID del motor derecho trasero, incidía en la salida del controlador PID del motor derecho delantero y a su vez se repetía el mismo problema en los motores del lado izquierdo.

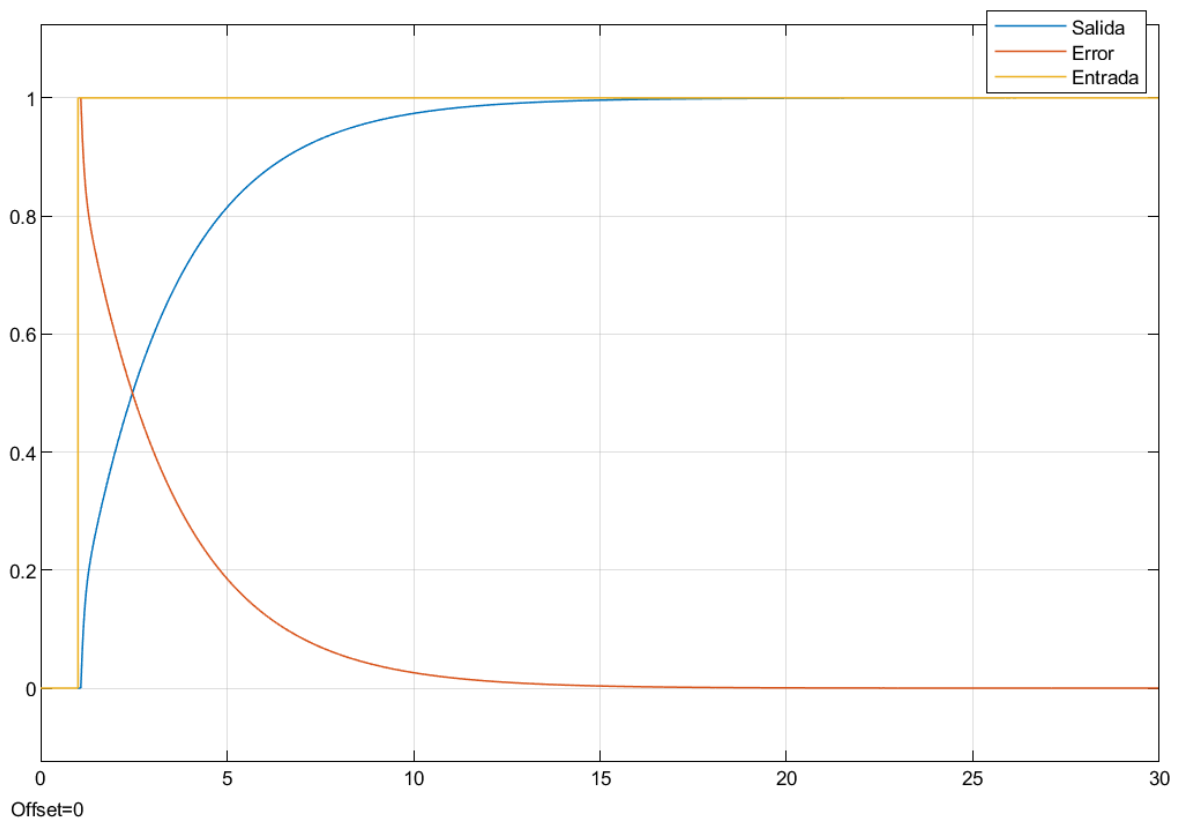
Para corregir este problema, se verificó mediante simulación en Simulink, la salida de los controladores de cada motor, lo que permitió observar que los controladores al trabajar en cadena en cada lado, lo que hacían era incrementar el orden del sistema, generando inestabilidad, dada su disposición física en el vehículo.

Por tal razón, fue necesario realizar el diseño de dos controladores PID, los cuales permitieran controlar cada uno de los lados de tracción del vehículo, teniendo en cuenta la variable Velocidad en ambas llantas, es decir que la tracción del lado derecho, tanto el motor delantero como trasero se convierten en una sola entrada para el algoritmo de control PID que rige ambos motores. Esto se replicó en el lado opuesto.

Luego de la corrección implementada, el sistema ya entregaba una salida con menos oscilaciones en la variable controlada, lo cual permitió iniciar la sintonización de los controladores calculados para su control. Se realizaron ajustes finos en

campo, ayudando a la respuesta de la variable controlada, en términos de velocidad de respuesta, tiempo de establecimiento y error en estado estable. Esto generó un control más óptimo, el cual termino comportándose dinámicamente como un sistema de primer orden.

### Ilustración 25. Respuesta del Sistema bajo Control PID Ajustado, Motores Izquierdo



Fuente: Elaboración propia.

## **4.3 PLANOS MECANICOS**

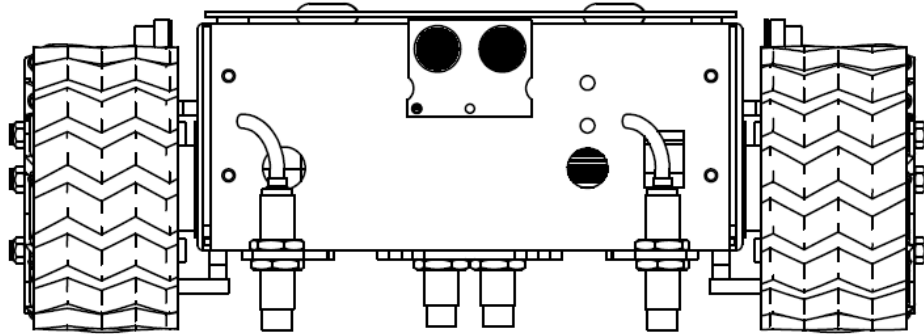
### **4.3.1 Plataforma**

El vehículo autónomo cuenta con tracción en las cuatro ruedas con motores y cajas reductoras con una relación de transmisión 64:1, para poder generar las diferentes marchas por bloques o lados y así generar cambios de dirección y distribuir la carga en las cuatro ruedas, para mayor control y tracción se incorporan dos ruedas traseras macizas de 100mm de diámetro, para darle tracción y estabilización del sistema, en la parte delantera se encuentran dos ruedas omnidireccionales para reducir la fricción en el momento del giro y minimizar el momentum en la estructura total del vehículo, generando así mejor tiempo de respuesta al encontrar una curva o corregir sobre la línea guía, suavizando los movimientos del vehículo, además reduciendo el esfuerzo sobre los motores, esto se logra con la configuración de dos ruedas macizas traseras que estabilizan los movimientos bruscos de las ruedas delanteras y ruedas omnidireccionales delanteras minimizan la fricción sobre el vehículo en especial el esfuerzo sobre los ejes.

Las llantas omnidireccionales se ubican de tal manera que el ángulo de los rodillos esté entre ellos con un desfase de 90 grados y con la cara interna de los rodillos atrás, buscando con ello minimizar el subviraje, ya que las omnidireccionales según la dirección de giro generan movimientos axiales, para ir en línea recta es necesario ubicar estas en dirección contraria para cancelar el movimiento axial generado por la rueda contraria, en el momento de tomar una curva las omnidireccionales deben

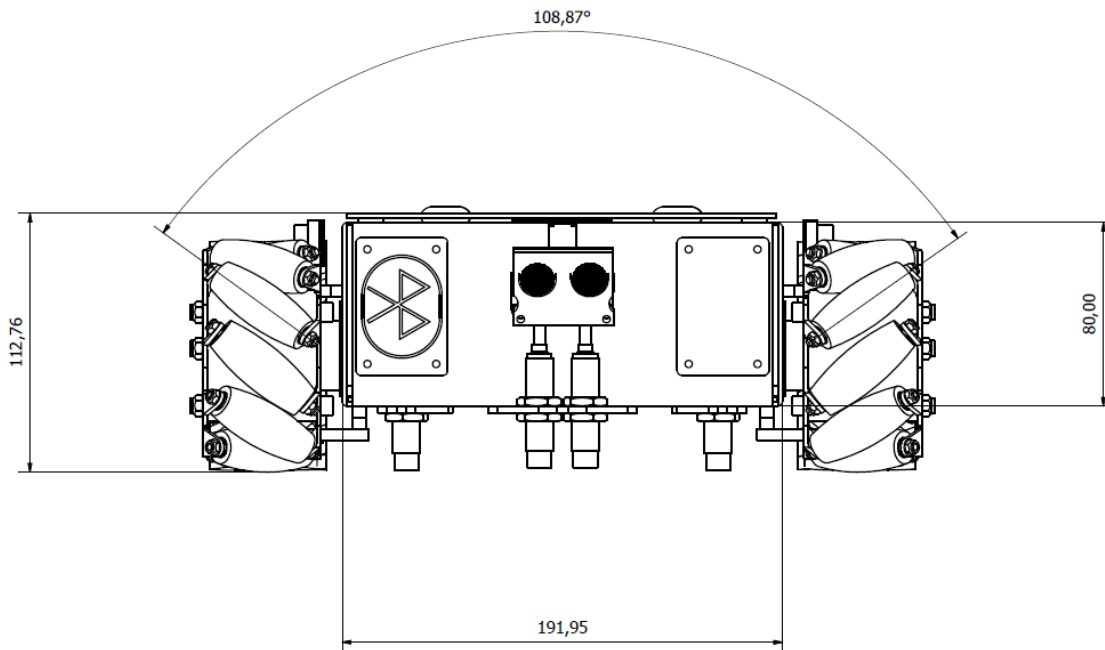
aumentar la velocidad en el costado contrario al giro y reduciendo la velocidad en el lado que se va a girar, en esto las omnidireccionales son importante para reducir la fricción sobre el suelo, estabilizar el vehículo y reducir esfuerzos sobre la estructura.

**Ilustración 26. Vista trasera del vehículo**



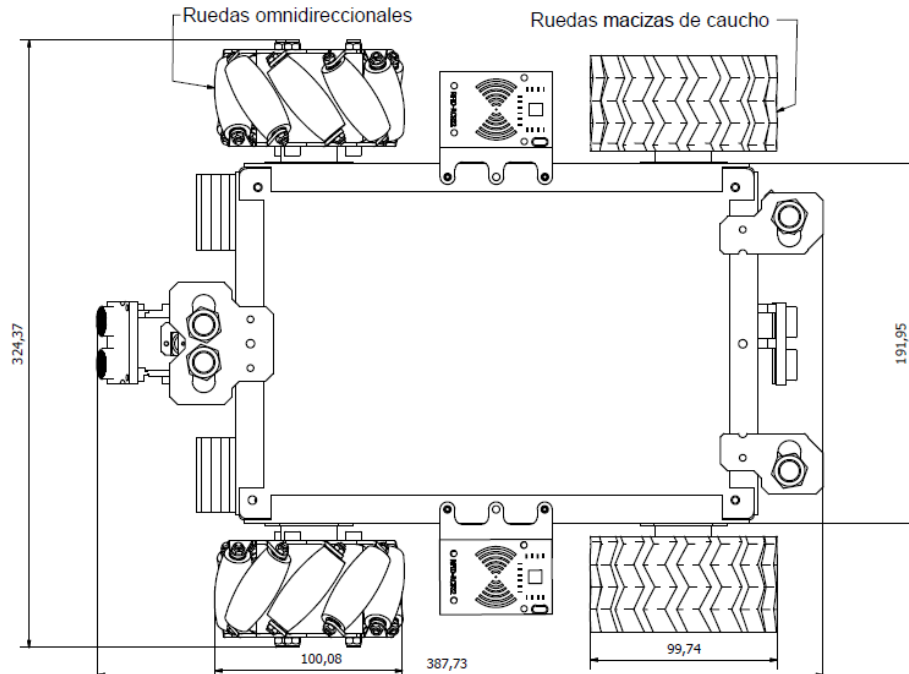
Fuente: Elaboración propia.

**Ilustración 27. Vista delantera del vehículo**



Fuente: Elaboración propia.

## Ilustración 28. Vista inferior del vehículo



Fuente: Elaboración propia.

### 4.3.2 Armazón de carga

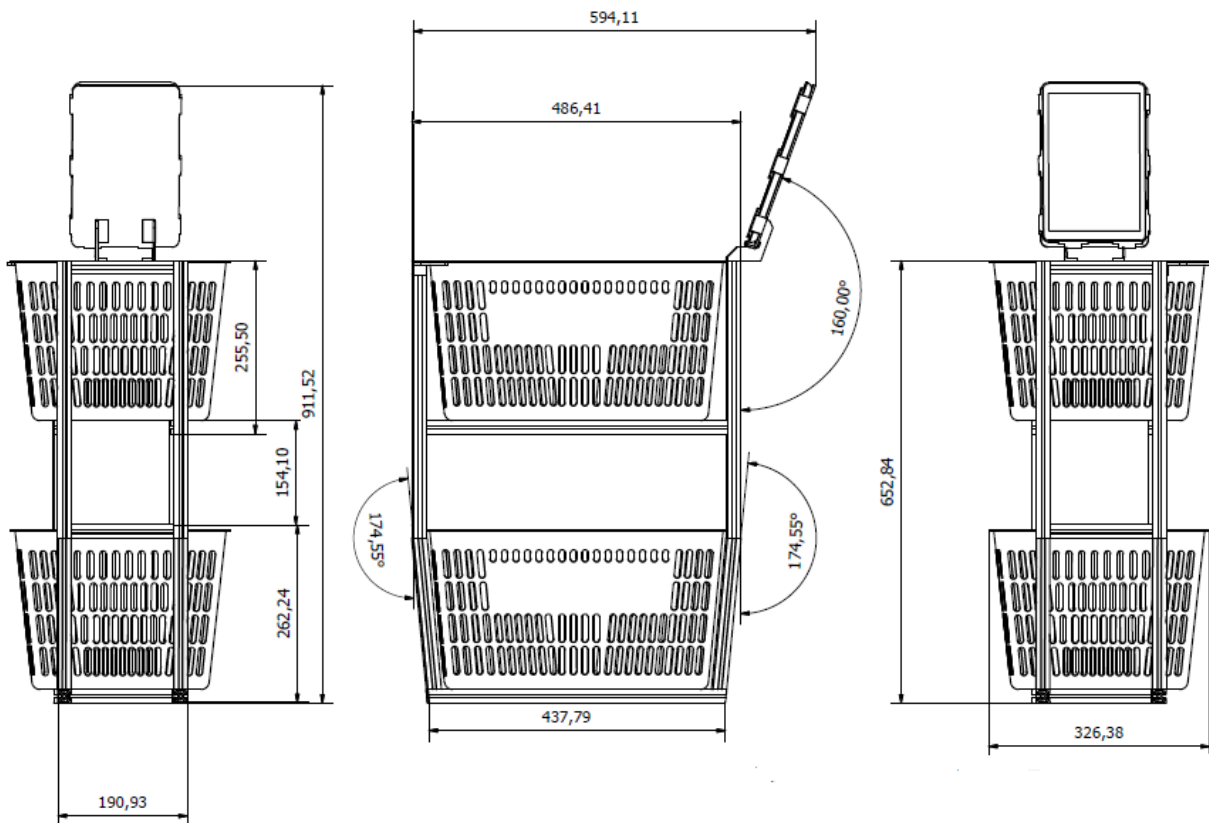
Está diseñado con perfiles en aluminio de 20x20 para reducir el peso, está estructurado con uniones angulares para reducción de vibraciones y estabilización de la misma; además, en la parte inferior la estructura cuenta con una reducción de 30 grados hacia la base con el fin de canalizar los esfuerzos generados en el centro de gravedad hacia la plataforma.

La estructura se diseñó con el fin de soportar dos canastas para la recolección de los pedidos que existían previamente en el laboratorio warehouse, de tal manera que puedan ser puestas o retiradas de manera fácil y minimizar los tiempos en el

receso, esto se consigue con la implantación de cuatro uñas de soporte 2 inferiores y 2 laterales.

En la parte superior de la estructura está ubicado el soporte para la Tablet que permite la interacción entre operación y el sistema del vehículo autónomo, ubicada en la parte delantera de éste, permitiendo la interacción y facilitando el movimiento del mismo.

**Ilustración 29. Plano armazón de carga**



Fuente: Elaboración propia.

### 4.3.3 Acople entre el armazón de carga y plataforma

El armazón cuenta con cuatro soportes en L que se unen a la estructura con tornillos, a su vez los soportes en L tienen en la parte inferior 4 amortiguadores de caucho con tornillo pasante entre placa y soporte, para soportar los esfuerzos y evitar la vibraciones, la placa superior de la plataforma cuenta con cuatro tornillos con bujes en T y O-ring para dar flexibilidad a la estructura de la plataforma, esto es necesario para que las ruedas puedan estar en contacto con el piso a pesar de las irregularidades de este.

### Ilustración 30. Vista general del vehículo

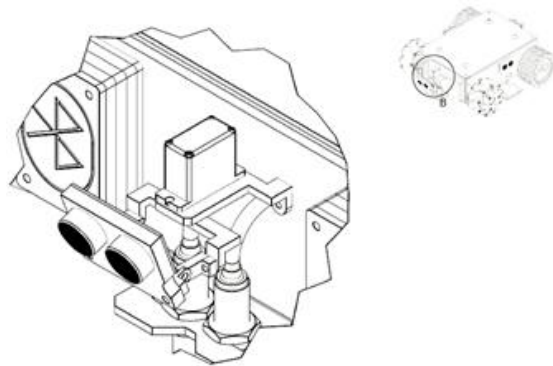


Fuente: Elaboración propia.

#### 4.3.4 Estructura de rotación ultrasonido

Cuenta con un servomotor de 9 gramos con giro de 180 grados, acoplado a la estructura con una pieza de aluminio. En la transmisión del servomotor se acopla una pieza de aluminio que soporta la articulación de ajuste para el ángulo de inclinación del ultrasonido.

**Ilustración 31. Vista articulación sensor ultrasonido**

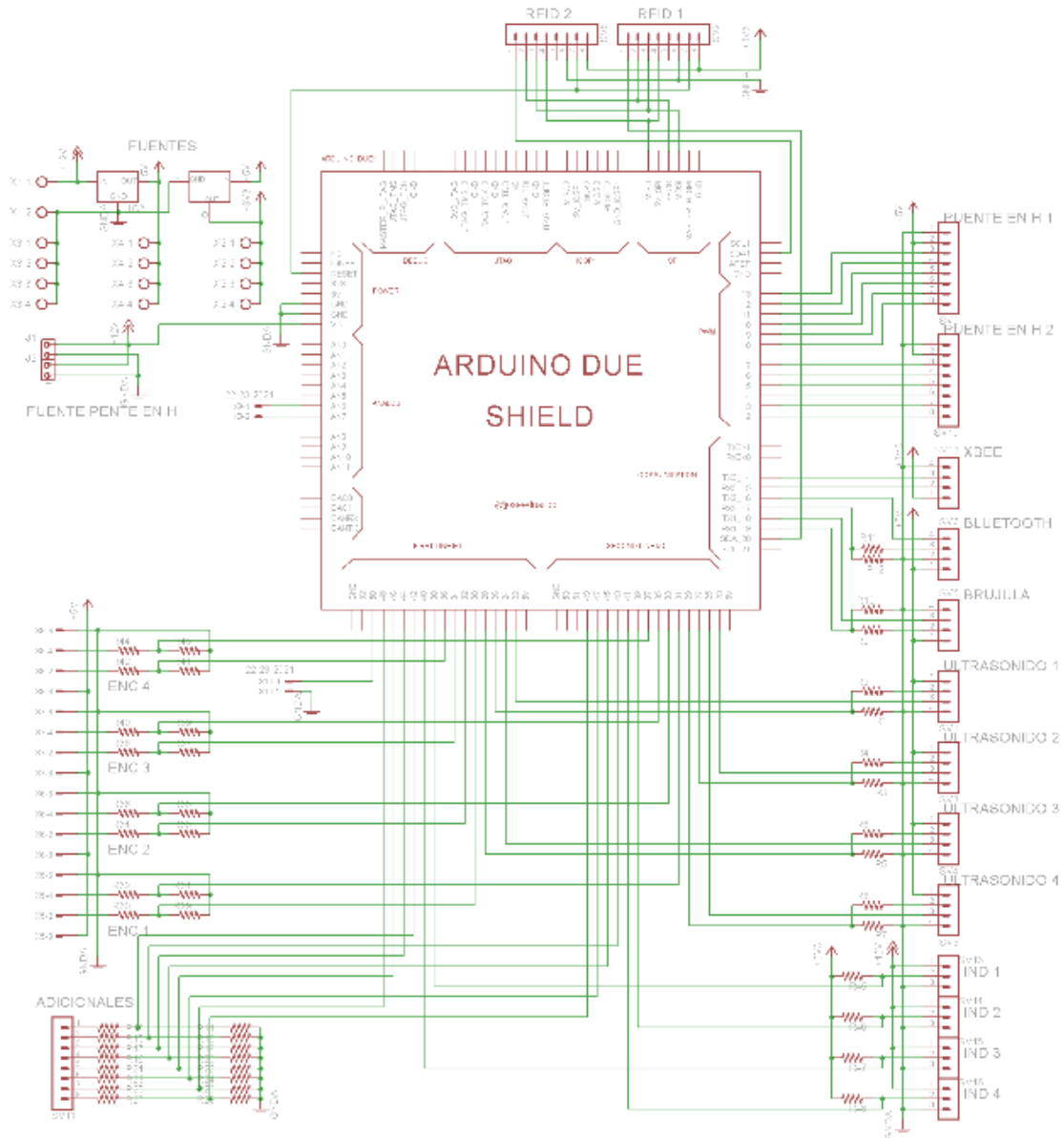


Fuente: Elaboración propia.

#### **4.4 PLANOS CIRCUITOS IMPRESOS**

- A continuación, se presenta los planos de los circuitos impresos que se diseñaron durante el desarrollo del proyecto.
- El primer circuito diseñado fue el circuito de control y comunicación del vehículo, este posee arduino Due encargado de las señales de control a cada uno de los motores de acuerdo a la información recibida por cada de las entradas que está conformada por sensores inductivos, infrarrojos, RFID, también es el encargado de la comunicación tanto con la Tablet a través del cual es el intermediario entre el software del PC y los vehículos comunicación

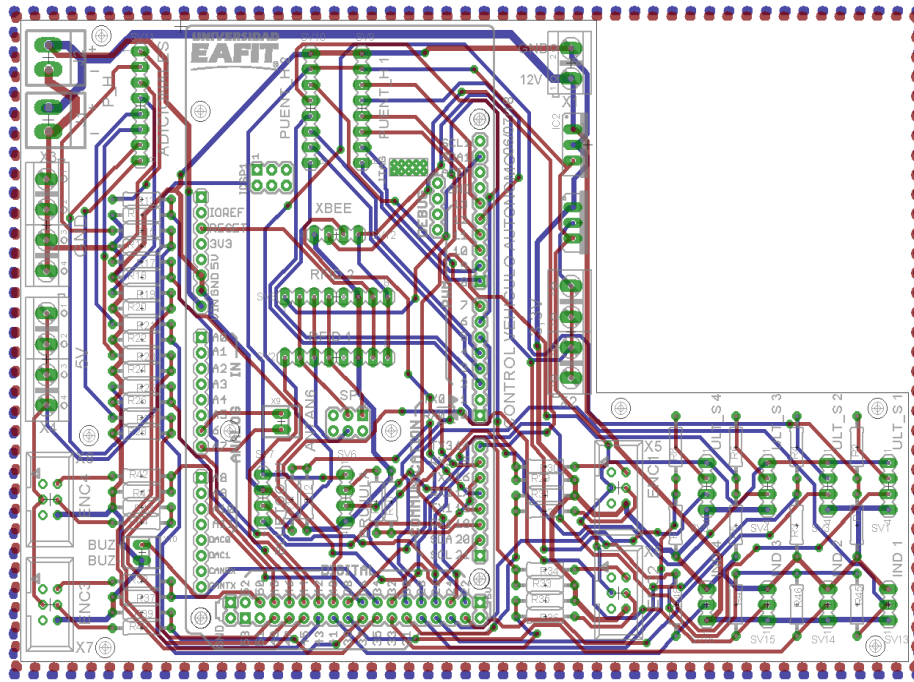
Ilustración 32. Esquemático del circuito arduino DUE



Fuente: Elaboración propia.

A continuación, se muestra la board de la tarjeta:

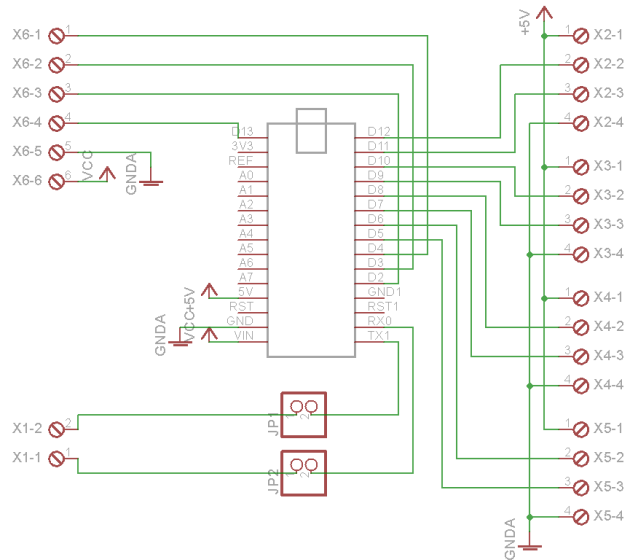
### Ilustración 33. Board del circuito arduino DUE



Fuente: Elaboración propia.

Explicación del circuito impreso de los ultrasonidos: Debido al tiempo que tardan los sensores de ultrasonidos en responder fue necesario implementar un microcontrolador que manejara esta parte y se comunicara de forma paralela y serial con el sistema del DUE

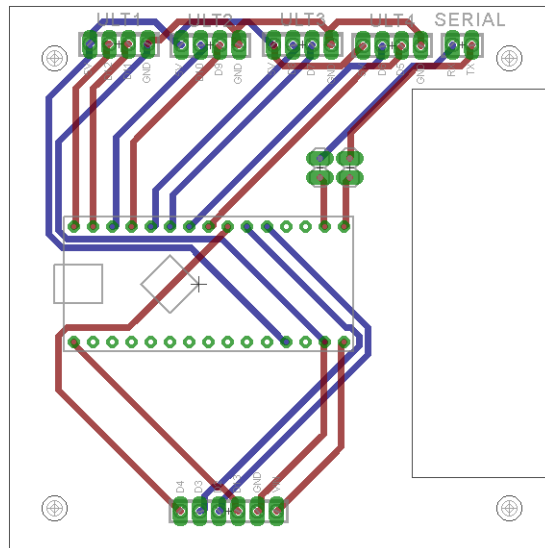
**Ilustración 34. Esquemático del circuito impreso arduino nano**



Fuente: Elaboración propia.

A continuación, se muestra la board del circuito impreso.

**Ilustración 35. Board del circuito impreso arduino nano**

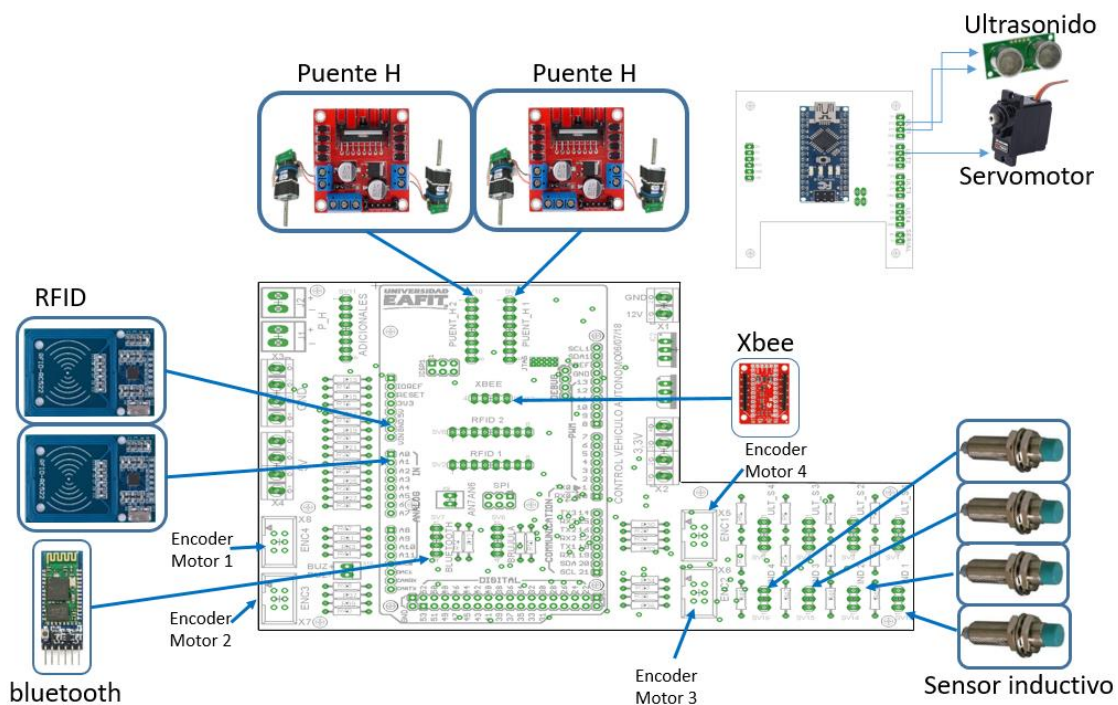


Fuente: Elaboración propia.

## 4.5 COMPONENTES DEL VEHÍCULO AUTÓNOMO

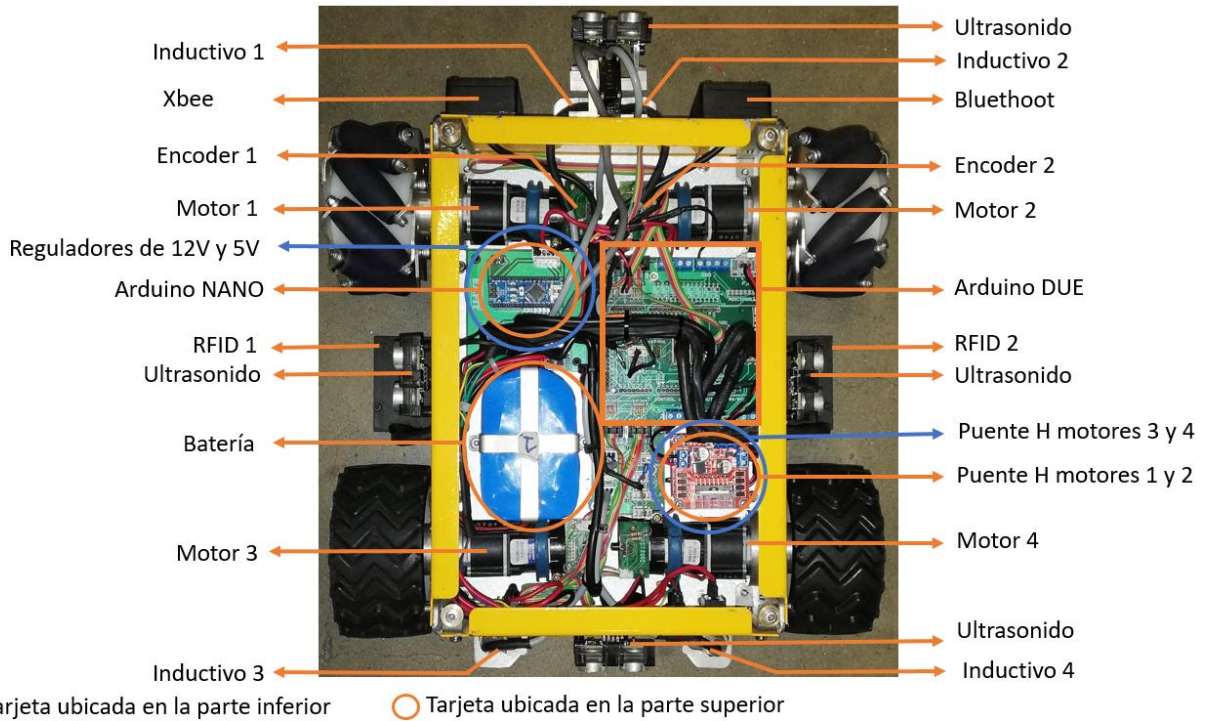
Conexión modular del vehículo autónomo:

Ilustración 36. Identificación de partes del circuito del vehículo autónomo



Fuente: Elaboración propia.

### Ilustración 37. Identificación de partes del vehículo autónomo



Fuente: Elaboración propia.

En la figura mostrada anteriormente se identifica cada uno de las partes del vehículo con su respectiva identificación el cual permite la interacción con los diferentes elementos del sistema, bien sea la comunicación o movimiento.

### Ilustración 38. Identificación de parte delantera del vehículo autónomo



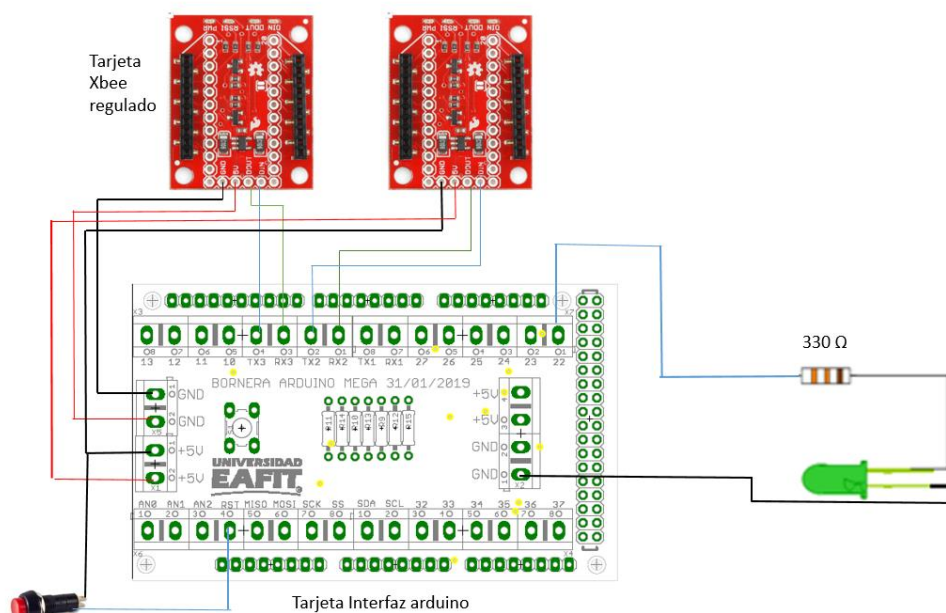
Fuente: Elaboración propia.

Explicación del circuito impreso de interfaz con el arduino por modulo:

El circuito impreso de interfaz con el arduino de cada módulo si requiere conectarse a elementos externos para que cumpla con su función de control.

Las conexiones externas a este circuito impreso son las siguientes:

### Ilustración 39. Conexiones externas al circuito impreso de interfaz de arduino



Fuente: Elaboración propia.

Explicación de las conexiones externas a la tarjeta de interfaz: El led indica que el arduino del módulo está en el estado cero y listo para recibir datos. El pulsador conectado al pin de reset inicializa el arduino, los xbee conectados a los seriales Serial3 permiten la comunicación con cada uno de los vehículos. El xbee conectado

al Serial2 permite usar un programa para encontrar posibles errores en la comunicación de cada módulo. El arduino se comunica con LabView a través del cable de programación y usa el Serial.

## **4.6 PROGRAMAS EN ARDUINO Y LABVIEW**

### **Programa vehículo autónomo:**

Funciones del arduino DUE que controlan el funcionamiento del vehículo autónomo:

```
void doEncode1() {
```

```
void doEncode2() {
```

```
void doEncode3() {
```

```
void doEncode4() {
```

Estas cuatro funciones son las encargadas de generar el conteo de los encoder teniendo en cuenta el sentido de los mismos, maneja cuatro contadores ISRCOUNTER (conteo acumulado del encoder en el recorrido), recorrido (para generar los conteos de pulsos en el recorrido de la línea), cp (para calcular la velocidad de los motores para el PID), Eu (para promediar los pulsos de los cuatro encoders).

```
void apagarmotores() {
```

```
void apagarmotoresretorno() {
```

Esta función realiza el apagado de los motores realizando una rampa de desaceleración buscando suavizar las detenciones en el recorrido.

**void estadosind() {**

Cada vez que se genera un cambio en los sensores inductivos se realiza una interrupción que llama esta función indicando en cuál de los 5 estados posibles se encuentran.

**void corregir() {**

Controla la potencia y dirección de los motores de acuerdo al estado indicado por los sensores inductivos.

**void iniciorampa(int val1, int val2) {**

Valor inicial de la rampa para iniciar los motores

**void rampa(int val1, int val2, int val3) {**

Rampa de aceleración de motores

**void retorno() {**

Función para retornar el vehículo cuando finaliza un recorrido y quedar nuevamente en posición para el siguiente.

**void promedioencoder() {**

Genera el promedio de los encoders de los cuatro motores descartando los valores que este más alejados.

**void leerRFID() {**

Función que se invoca cuando se llega algún tag de RFID y se lee el número correspondiente de la tarjeta

**void verificarRFID() {**

Verifica el valor leído por el RFID y le asigna una bandera de acuerdo al valor, con la cual se hacen las verificaciones de la MEF.

**void adelante() {**

Generar las direcciones en los puentes H para que el vehículo rote hacia adelante.

**void hallarpulsos(float distancia) {**

Dada una distancia, esta función calcula el número de pulsos que se deben dar para cubrir esta distancia.

**void hallardistancia(float pulsos) {**

Dado un número de pulsos calcular la distancia recorrida.

**void pitar() {**

Genera un zumbido con el buzzer.

**void salircurvaderecha() {**

Garantiza que el vehículo salga hacia la derecha en las bifurcaciones de la pista.

**void seguirderecha() {**

Continúa en la línea dando prioridad a seguir hacia la derecha.

**void salircurvaizquierda() {**

Garantiza que el vehículo salga hacia la izquierda en las bifurcaciones de la pista.

**void seguirizquierda() {**

Continúa en la línea dando prioridad a seguir hacia la izquierda.

**void separadatos() {**

Separa los datos que llegan en un String separados por comas, en un arreglo con solo los datos.

**void chat() {**

Dada una cadena se genera la trama del Xbee para poder ser enviada por este.

**void router() {**

Dirección del Xbee al que se va a enviar la información desde el chat, además genera la suma de esta para el checksum.

**long msg(String valor) {**

Envía el String de valor por el xbee y devuelve el tamaño del string.

**long checksum(long receiver, String text) {**

Genera el valor de checksum de toda la cadena enviada por el xbee.

**long convertir\_mensaje(String text) {**

Halla la suma de text convertidos en char para el checksum.

**void enrutar() {**

Esta función no se usa.

**void tag() {**

Esta función no se usa.

**void calcularpid() {**

Calcula el PWM de salida de los motores de acuerdo al PID y a la velocidad de los motores.

**void calcular() {**

Calcula de acuerdo a los pulsos del motor en un tiempo la velocidad de los motores que servirá para el cálculo del PID.

**void ultra() {**

Esta función no se usa.

**void setup() {**

Configuración de entradas y salidas, velocidad de los cuatro puertos seriales, inicialización de los RFID, inicialización de estado de sensores inductivos, habilitación de interrupciones de encoders, sensores inductivos, configuración inicial de los PID.

**void loop() {**

Máquina de estados.

**void serialEvent3() {**

Evento de recepción de información recibida por Xbee al vehículo.

**void serialEvent2() {**

Evento de recepción de datos del bluetooth.

### **MEF arduino DUE:**

Funciones del arduino nano:

**void setup() {**

**void loop() {**

**void serialEvent(){**

**void lectura(){**

Funciones del arduino Master:

**void router() {**

**long msg(String valor) {**

**long checksum(long receiver, String text) {**

**long convertir\_mensaje(String text) {**

**void chat() {**

**void serialEvent()**

```
void serialEvent3()
```

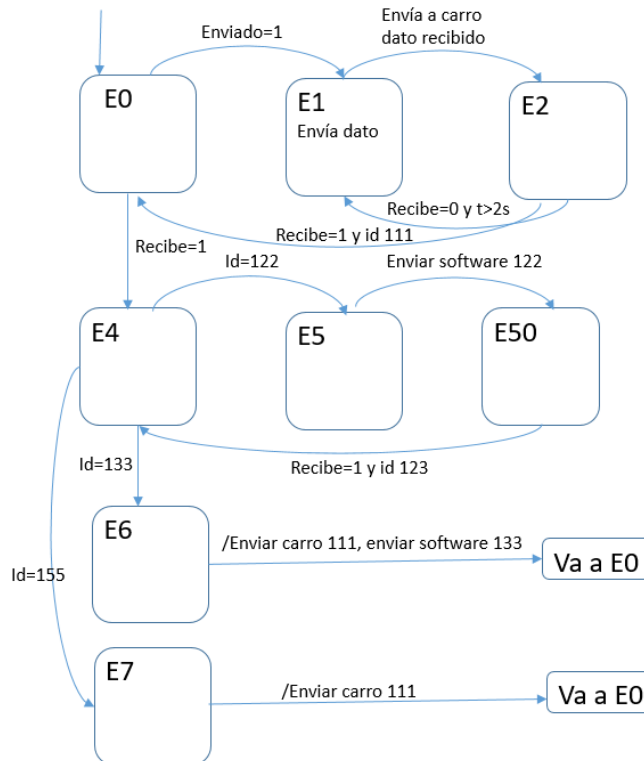
```
void setup() {
```

```
void loop () {
```

### MEF del master de comunicación Labview - vehículo autónomo:

El master es el encargado de manejar la comunicación entre labview y los vehículos a través de los Xbee, permitiendo liberar procesamiento a ambos, además este tiene un canal de comunicación independiente con los vehículos y labview.

### Ilustración 40. Máquina de estado finito 1 programa en labView



Fuente: Elaboración propia.

Estado 0 (E0): Espera a recibir un dato desde el software o el vehículo.

Estado 1 (E1): Si labview manda un dato se envía este al vehículo.

Estado 2 (E2): Espera confirmación del vehículo de dato enviado si no confirma en 2 segundos vuelve a enviar.

Estado 4 (E4): Si recibe un dato del vehículo, verifica el dato y de acuerdo a esto va al estado correspondiente.

Estado 5 (E5): Si recibe del carro 122, envía este dato al software y va a E50.

Estado 50 (E50): Espera respuesta del software de segmento libre y lo envía al vehículo.

Estado 6 (E6): Envía el 133 al software y envía la confirmación al vehículo.

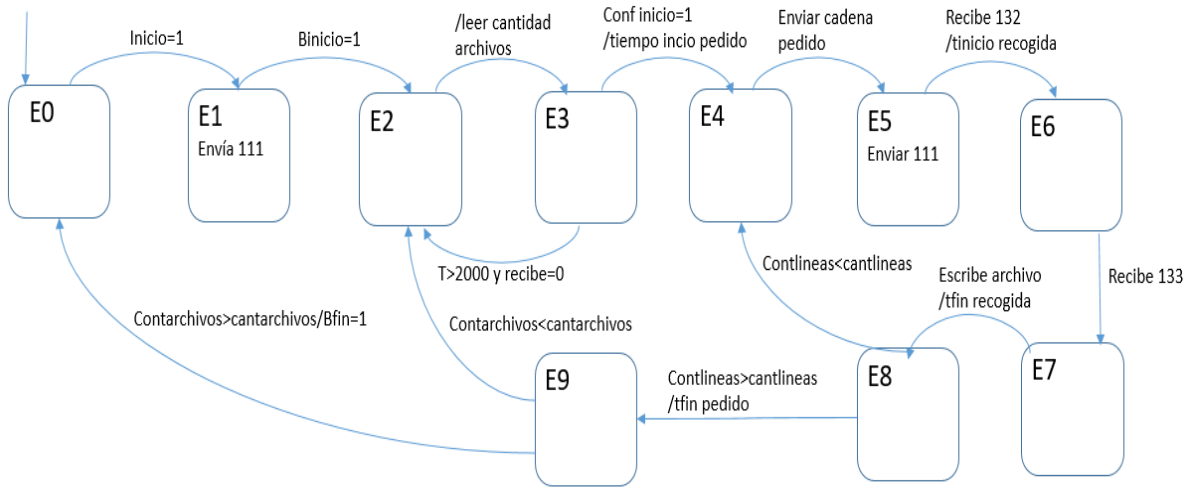
Estado 7 (E7): Envía confirmación al vehículo 111.

### **Máquina de estado finito del programa en LabView:**

La programación de los vehículos autónomos está hecha con LabView como programa que se comunica con cada vehículo, lee y actualiza los archivos y permite la interfaz con el usuario.

MEF 1: Manejo de datos y archivos con el vehículo

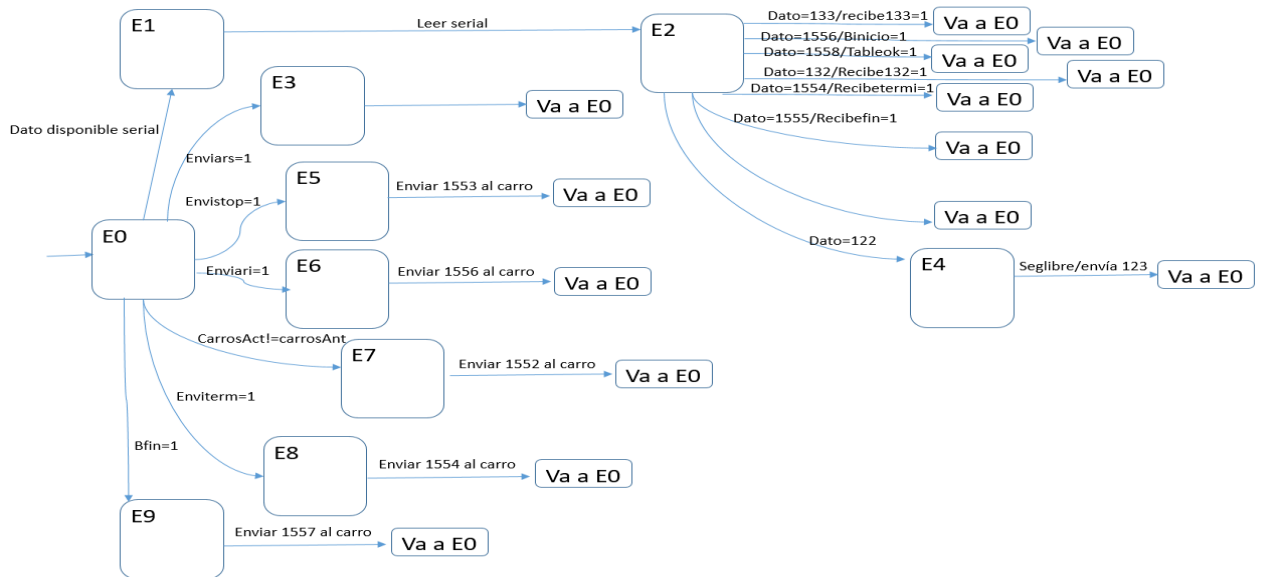
**Ilustración 41. Máquina de estado finito 1 programa en labView**



Fuente: Elaboración propia.

MEF 2: Comunicación serial

**Ilustración 42. Máquina de estado finito 2 programa en labView**



Fuente: Elaboración propia.

## **Explicación de la MEF en LabView:**

### **MEF1:**

Esta se encarga del manejo de los archivos y controlar el flujo de información que debe ir al carro o el control de acuerdo a la información que envía el vehículo.

Estado 0 (E0): Espera inicio para hacer el recorrido del pedido.

Estado 1 (E1): Activa la bandera Binicio para la MEF2, y confirma recepción 111.

Estado 2 (E2): Lee el archivo con la lista de archivos para ejecutar los pedidos.

Estado 3 (E3): Espera confirmación de inicio del vehículo.

Estado 4 (E4): Envía cadena del pedido a realizar.

Estado 5 (E5): Espera el 132, indicando que el vehículo llegó al punto del pedido.

Estado 6 (E6): Espera confirmación del operario de toma del pedido. Recibe 133.

Estado 7 (E7): Escribe en el archivo el pedido ejecutado, con cantidad confirmada.

Estado 8 (E8): Verifica si las líneas del pedido ya se ejecutaron la al E9 o si por el contrario falta va al E4.

Estado 9 (E9): Verifica si la cantidad de pedidos término va al E0 o va al E2, si tiene más archivos con pedidos para ejecutar.

### **MEF 2:**

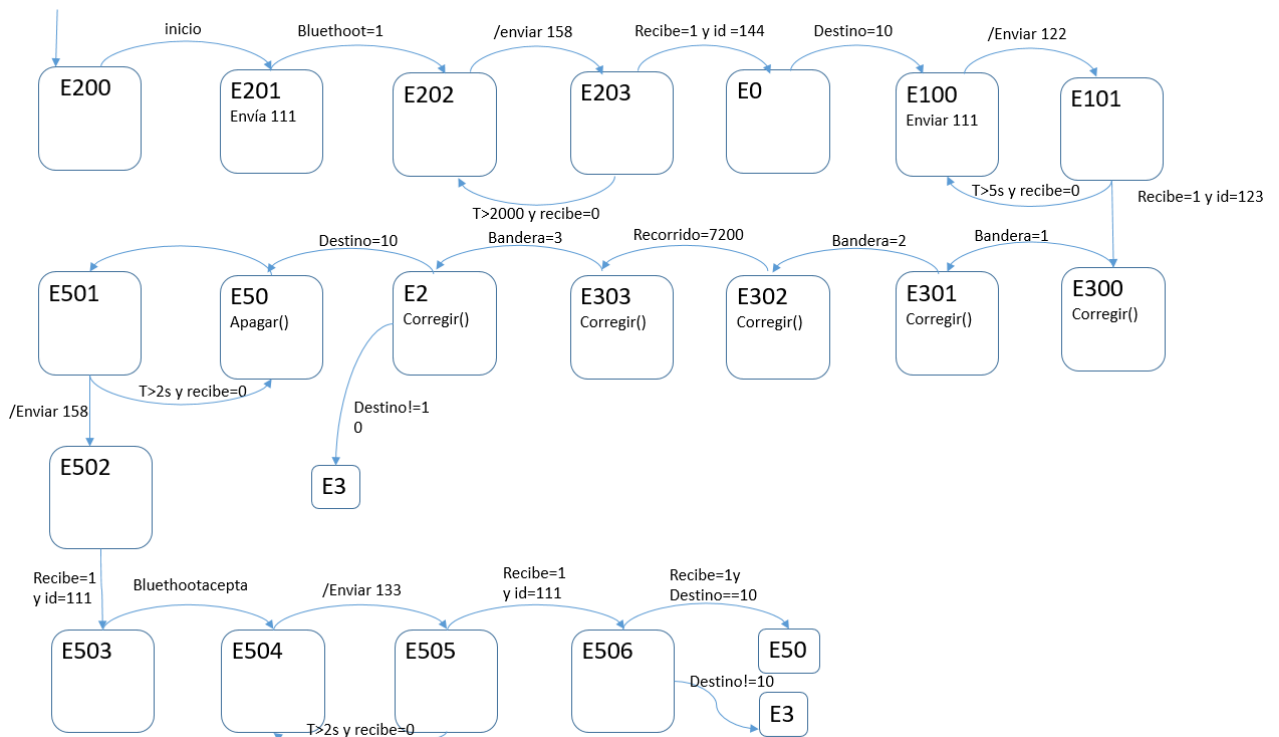
Esta MEF se encarga de manejar la comunicación serial con el master, tanto para enviar los datos como para recibir, al igual que la confirmación.

- Estado 0 (E0): Espera a recibir un dato serial o que alguna de las banderas de comunicación sea activadas.
- Estado 1 (E1): Lee el dato recibido por el puerto serial y pasa al E2.
- Estado 2 (E2): Verifica el valor y activa la bandera respectiva para la MEF 1, si el dato recibido es 122 va al E4.
- Estado 3 (E3): Envía cadena del pedido a realizar en el vehículo.
- Estado 4 (E4): Verifica si el segmento al que el vehículo va a entrar esta libre y de ser así envía un 123.
- Estado 5 (E5): Cuando se da stop se envía 1555 mensaje de stop al vehículo.
- Estado 6 (E6): Cuando se da inicio se envía 1556 mensaje de inicio al vehículo.
- Estado 7 (E7): Cuando hay dos carros trabajando se envía 1552 a los vehículos.
- Estado 8 (E8): Cuando se da terminar, envía 1554 mensaje de terminar al vehículo.
- Estado 9 (E9): Cuando se da fin se envía 1557 mensaje de fin al vehículo.

## Máquina de estado finito Vehículo:

- La función principal de la MEF del vehículo es corregir, controlar la comunicación que recibe por serial de forma segura evitando errores en la comunicación a través de protocolo seguro y evitar las colisiones, a continuación, se muestra parte de la MEF la cual atiende el primer segmento ya que las demás son exactamente iguales para los demás segmentos, se incluirá como anexo.

**Ilustración 43. Máquina de estado finito 2 programa en labView**



Fuente: Elaboración propia.

### **Explicación MEF vehículo autónomo:**

Inicia en el E200, cuando recibe la orden de inicio del master pasa al estado 1 confirma y espera orden de la Tablet, este envía orden de inicio en el E203 y espera el primer pedido, en el E100 y E101 pregunta si el segmento esta libre, al recibir el 123 con esta confirmación pasa al E300 que arranca hacer el recorrido hasta llegar al primer punto de condición E2, sigue la corrección hasta llegar al punto E50, apaga motores, envía al master que llego al punto de detención 158 E501, espera confirmación E502, en el E503 espera confirmación de operario E504, y al confirmar envía 133 con la cantidad al master, recibe confirmación E506, recibe siguiente trama si esta es en el mismo punto vuelve al E50 y si es diferente va al E3 para seguir en el otro segmento, este proceso se repite para todos los segmentos hasta terminar.

A continuación, se presenta una tabla con los mensajes que se envían entre el software central, el master y el vehículo para hacer el control de este.

**Tabla 5. Identificación de Mensajes**

<b>Código</b>	<b>Descripción</b>
111	Confirmación de recepción de una trama valida
122	Pregunta si el segmento esta libre cuando hay dos vehículos
123	Confirma y autoriza paso cuando el segmento esta libre
132	Confirma llegada a punto de detención
133	Confirmación de toma de pedido y cantidad
144	Trama de pedido
1552	Confirmación de dos (2) carros en ruta
1553	Mensaje de Stop, detiene toda la operación
1554	Mensaje de terminar el pedido actual ir a inicio y seguir con el pedido siguiente.
1555	Mensaje de Fin, va a inicio y no continua con los pedidos
1557	Mensaje indicando que termino de recoger todos los pedidos
1558	Mensaje de inicio dado por la tablet
1561	Iniciar comunicación con 1 vehículo activo
1562	Iniciar comunicación con 2 vehículo activos

Fuente: Elaboración propia.

A continuación, la tabla con los intercambios de mensajes entre los sistemas:

**Tabla 6. Intercambio de mensajes entre el software, master y el vehículo**

<b>Software</b>	<b>Master</b>	<b>Vehículo</b>
Envía 1561	Envía 1561	Recibe 1561 y Confirma 111
Recibe 158	Envía 158 y confirma 111	Envía 158
Envía 144	Envía 144	Recibe 144 y confirma 111
Recibe 132	Envía 132 y confirma 111	Envía 132
Recibe 133	Envía 133 y confirma 111	Envía 133
Envía 1557	Envía 1557	Recibe 1557 y confirma 111
Recibe 122	Envía 122 y confirma 111	Envía 122
Envía 123	Envía 123	Recibe 123 y confirma 111
Envía 1553	Envía 1553	Recibe 1553 y confirma 111
Envía 1554	Envía 1554	Recibe 1554 y confirma 111
Envía 1555	Envía 1555	Recibe 1555 y confirma 111
Envía 1557	Envía 1557	Recibe 1557 y confirma 111

Fuente: Elaboración propia.

## 4.7 PLANOS DE CONEXIÓN DEL VEHICULO

Ilustración 44a. Plano de conexiones vehículo autónomo 1 de 3

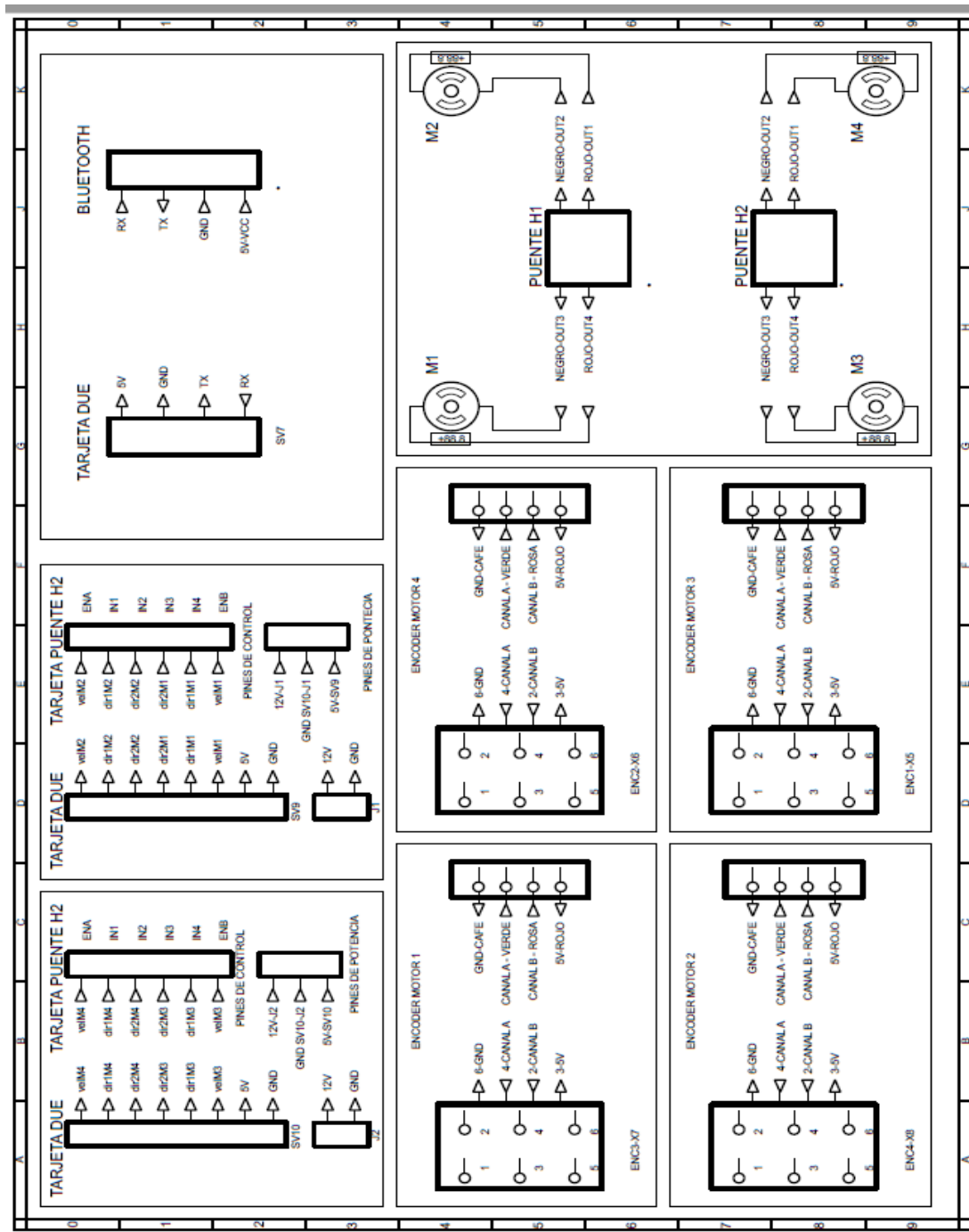


Ilustración 44b. Plano de conexiones vehículo autónomo 2 de 3

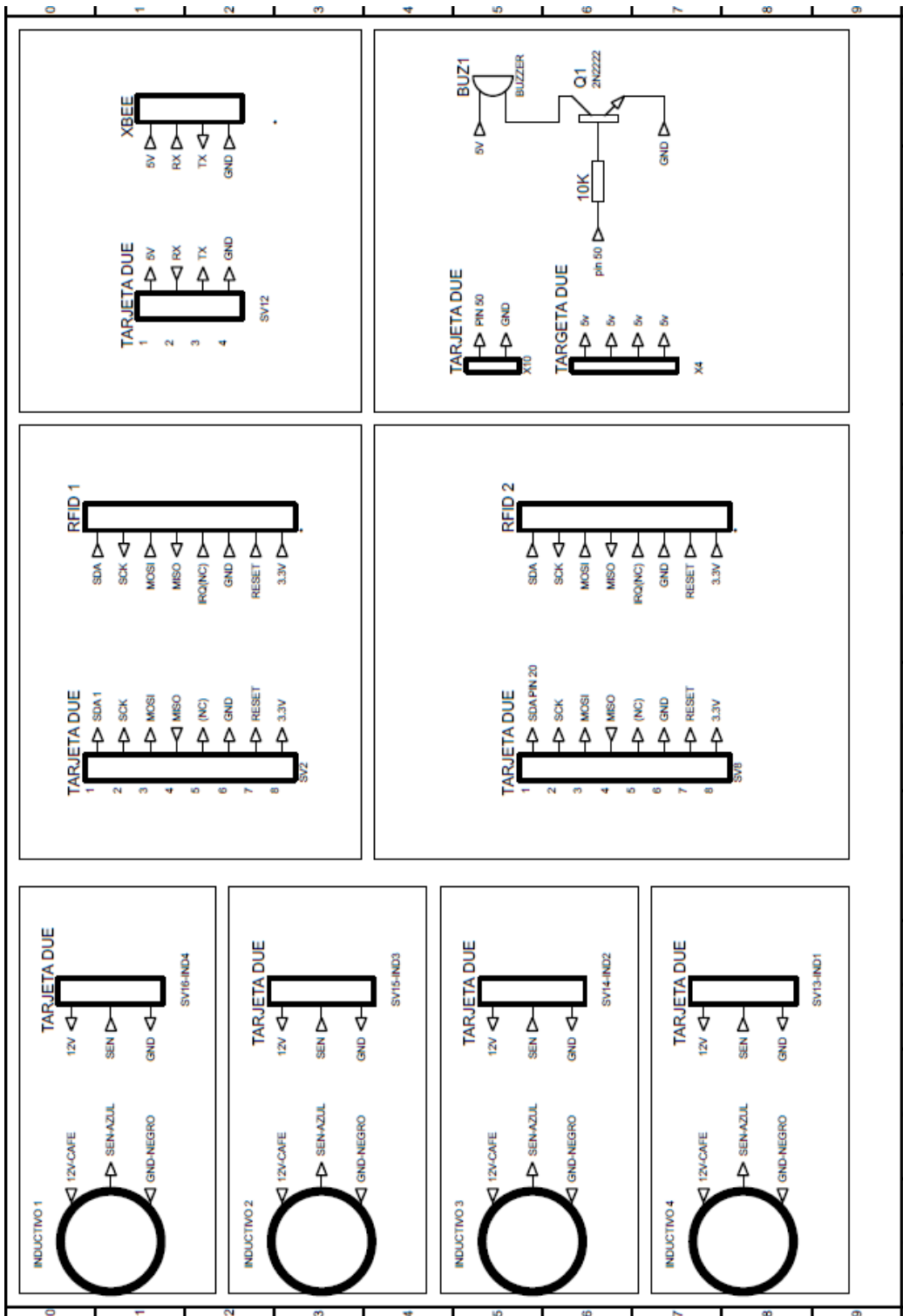
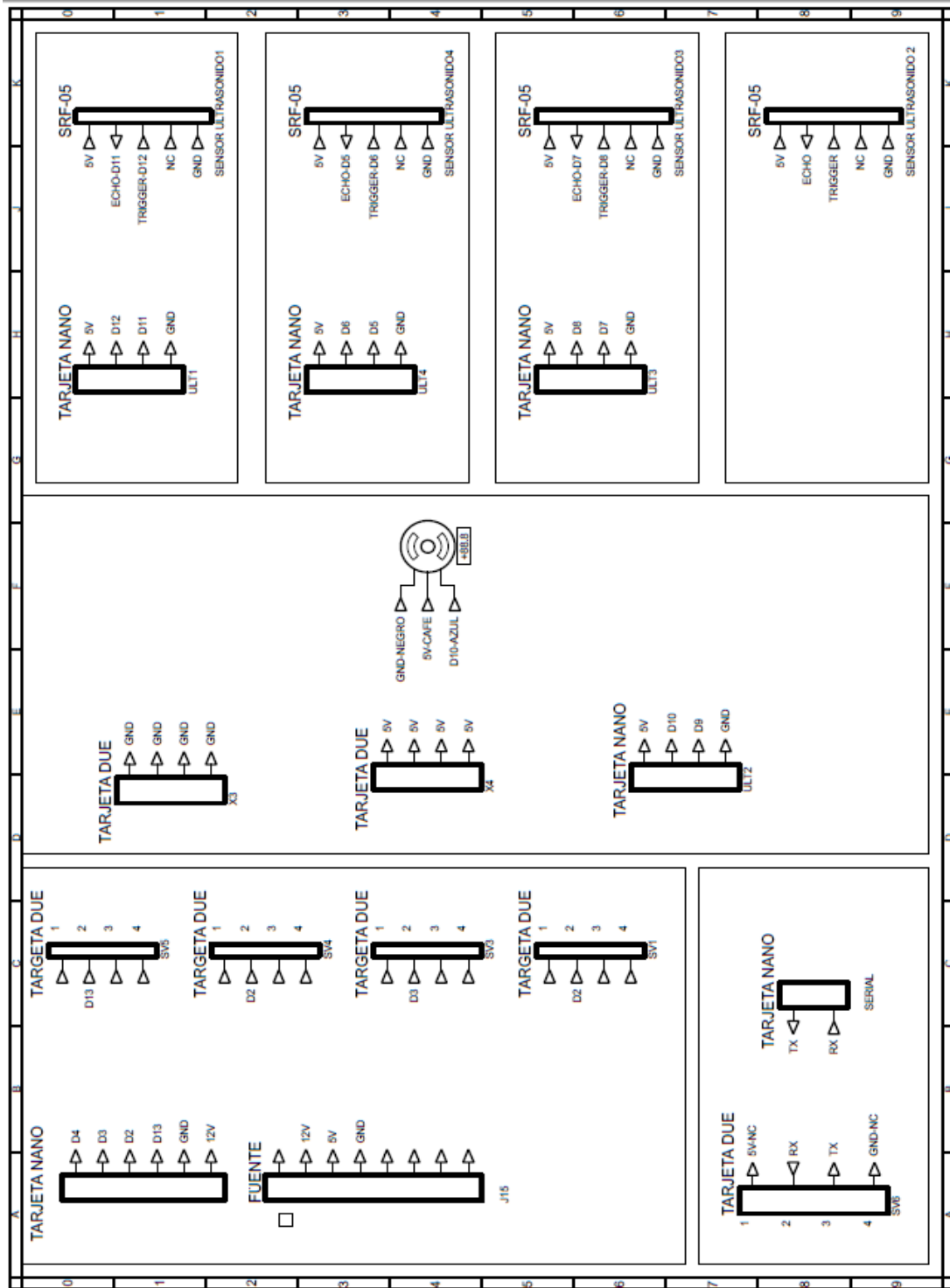


Ilustración 44c. Plano de conexiones vehículo autónomo 3 de 3

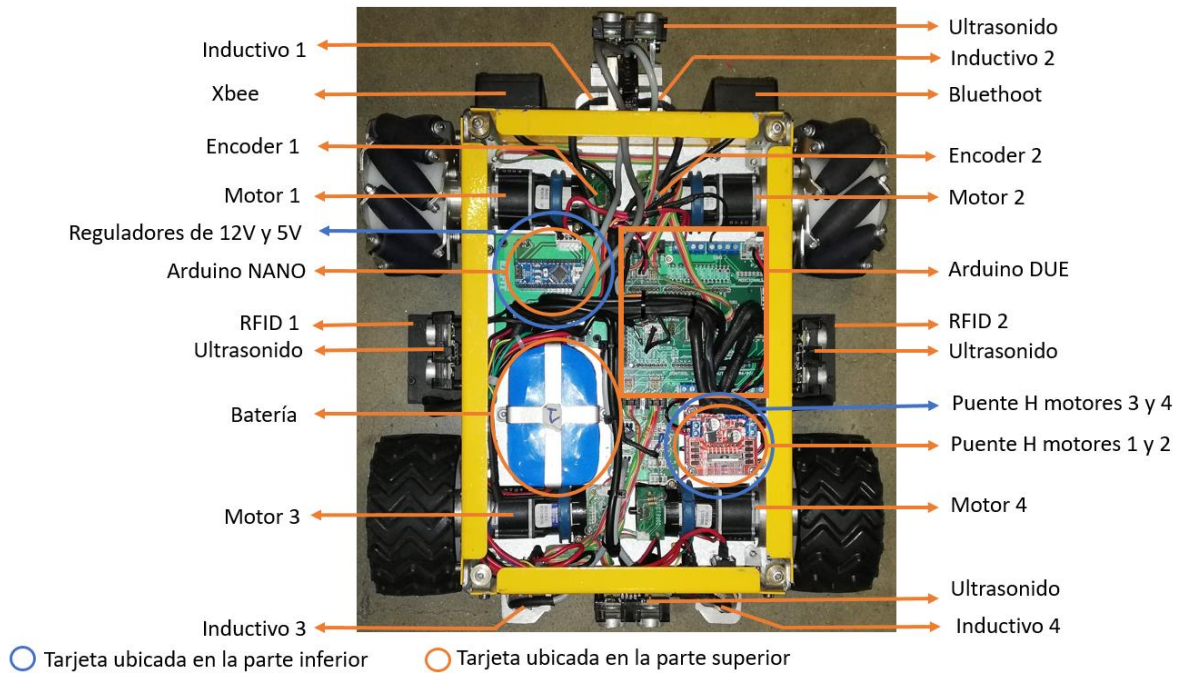


Fuente: Elaboración propia.

## 4.8 MANUAL DE USUARIO

### Sistema de vehículos autónomo:

#### Ilustración 45. Descripción de partes del vehículo



Fuente: Elaboración propia.

Para en ensayar y probar el sistema se deben seguir lo siguientes pasos en estricto orden:

- Verificar la posición de los sensores inductivos de la parte delantera, que estos se encuentren a la altura y posición correcta. Este paso es fundamental ya que estos son los ojos del vehículo y si estos se encuentran en forma incorrecta generan que el vehículo se pierda e incluso pueda generar una colisión dañando el vehículo y los que se encuentre a su paso.

- Ubicar el vehículo de tal manera que los sensores delanteros queden ubicados sobre la línea metálica.
- Encender el vehículo del suiche ubicado en la parte trasera del mismo.

#### Ilustración 46. Descripción parte trasera del vehículo

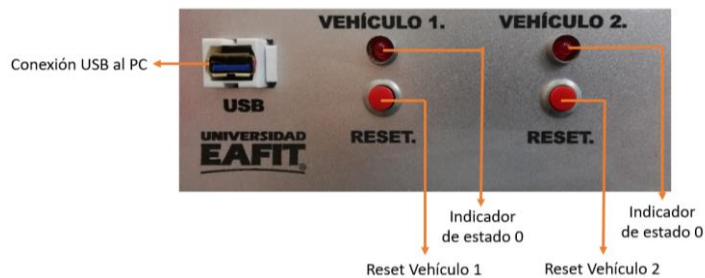


Fuente: Elaboración propia.

- Proceder a conectar la Tablet con el vehículo a través del bluetooth ingresando al programa vehículo autónomo de la Tablet, en la pantalla inicial aparece un botón CONECTAR, presionar este y de la lista de conexiones seleccionar la del carro que se tenga ubicado listo para iniciar (Carro 1 o Carro 2), si la conexión es correcta pasa a la siguiente pantalla sin ningún mensaje de error, de aparecer algún mensaje dar en desconectar y repetir el procedimiento.
- En la pantalla debe aparecer el mensaje esperando inicio, las cual se debe seleccionar solo después de ejecutar el programa en Labview.

- Antes de ejecutar el programa se debe garantizar que los leds indicadores de “estado cero” de cada módulo estén encendidos, si alguno de ellos está apagado, simplemente se oprime el pulsador de reset ubicado en la parte inferior de este.

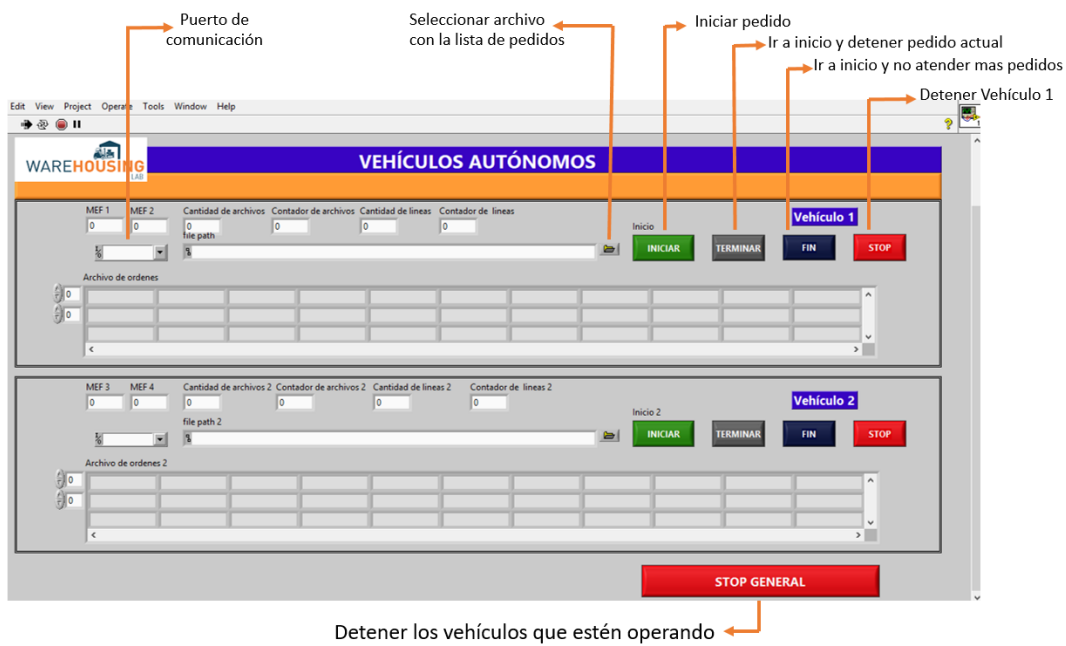
#### Ilustración 47. Parte delantera del master



Fuente: Elaboración propia.

- Abrir el archivo en LabVIEW “Vehiculos autonomos” que se encuentra ubicado en la siguiente ruta *C:\Users\warehousinglab\Desktop\Sistema.....*
- Conectar la caja del master al puerto USB (siempre se debe conectar al mismo puerto) para garantizar que los puertos USB usados en el programa sean los mismos.
- Cargar la ruta correspondiente a la lista de pedidos

## Ilustración 48. Descripción panel frontal software de operación



Fuente: Elaboración propia.

- Ejecutar el programa de LabVIEW y realizar la recogida de los pedidos.
- Labview tiene cuatro pulsadores, iniciar como su nombre lo indica es el que arranca el sistema y la comunicación, Terminar que detiene el pedido que se está realizando y va al principio del recorrido y continua con el siguiente pedido, fin que va al principio del recorrido y detiene todos los pedidos, y stop que detiene la operación en cualquier punto.

#### 4.9 FALLAS Y FORMAS DE SOLUCIONARLAS

En la siguiente tabla se presentan las fallas más comunes que puedan presentarse con su correspondiente posible solución.

**Tabla 7. Algunas fallas y formas de solucionarlo**

<b>Falla</b>	<b>Diagnóstico</b>	<b>Posible solución</b>
El vehículo se pierde en la línea y se sale hacia los lados.	Los sensores del vehículo tienen problemas.	Revisar el correcto funcionamiento de los sensores con el indicador de la parte superior y verificar la altura.
Al dar inicio desde el software el vehículo no arranca.	El vehículo no está encendido. El vehículo no tiene comunicación.	Se debe encender y reiniciar el master y el software. Apagar y encender de nuevo el vehículo, reiniciar master y software
El vehículo no lee los tag RFID	El vehículo avanza por la línea pero no se detiene en ningún lado	Revisar que los tag este en la posición correcta, ya que estos son muy importantes en la ubicación del vehículo.

Fuente: Elaboración propia.

## **PRUEBAS Y VALIDACION**

Durante la etapa de desarrollo del proyecto se hicieron diferentes pruebas para confirmar el funcionamiento de los programas y modificar algunas fallas que se presentaron. En cada prueba participaron varias personas validando el funcionamiento y sugiriendo mejoras. Antes de hacer pruebas los programas fueron probados por el desarrollador del sistema.

## **5. CONCLUSIONES Y SUGERENCIAS PARA TRABAJOS FUTUROS**

### **5.1 CONCLUSIONES**

Se implementó el software de control de los vehículos autónomos en el software Labview este se encarga de la comunicación y lectura de archivos de pedidos con los vehículos, permitiendo la interacción entre el usuario y el vehículo.

Se construyó un master intermediario entre el software de control y el vehículo para descargarle procesamiento tanto al vehículo, como al software de control.

Se logra la construcción de dos vehículos autónomos, capaces de seguir la línea del Warehousing Lab y las rectas de acuerdo a las órdenes dadas por el sistema central.

Se implementó la comunicación con dispositivos Xbee, usando tramas API las cuales permiten el envío de información de forma segura, siempre y cuando se tenga un protocolo de confirmación desde el dispositivo que transmite o recibe la información. Esta función no se puede dejar solo en el Xbee, y es precisamente la que se implementó en el master.

Al terminar la ruta los vehículos, los archivos planos quedaran actualizados en los cuales se entregan los tiempos del pedido general, el tiempo de cada pedido y los mensajes de acuerdo a la operación de los mismo.

## 5.2 SUGERENCIAS PARA TRABAJOS FUTUROS

Como posibles trabajos futuros o mejoras a este proyecto se plantean lo siguiente:

- Pasar el programa en LabView a Python que es un lenguaje libre y por lo tanto de fácil distribución.
- Implementar una cámara para seguir la línea a través de un sistema económico, liviano y rápido para el procesamiento de imágenes.

## BIBLIOGRAFÍA

Bleum (2017). *Two models of mobile shelf*. Disponible en:

<http://www.bleum.com/robot-models/>

Brown, R. (2016). *Multi-agent robotic systems in logistics*. Disponible en:

[https://tams.informatik.uni-hamburg.de/lectures/2016ws/seminar/ir/doc/slides/RobertBrown-Multi-Agent\\_Robotic\\_Systems\\_in\\_Logistics.pdf](https://tams.informatik.uni-hamburg.de/lectures/2016ws/seminar/ir/doc/slides/RobertBrown-Multi-Agent_Robotic_Systems_in_Logistics.pdf)

Caprile, S. R. (2009). *Desarrollo de aplicaciones con comunicación remota basados en módulos Zgbee y 802.15.4*. Argentina.

Carrasco, R. & Ponce, E. (2008). *Mejora de la eficiencia de una central logística mediante el rediseño del reaprovisionamiento de la zona de picking*.

Correa, R. A. & Gómez, R. (2009). *Tecnologías de la información en las cadenas de suministro*.

Chard, R. (2016). *Tendencias tecnológicas al servicio de la gestión de almacenes*.

Díaz, C. & Arias, R. (2009). *Una propuesta tecnológica basada en radiofrecuencia para procesos de picking en los cuartos fríos de industrias Zenú S.A.S.*

Errasti, A., Chackelson, C. & Arcelus, M. (2009). *Estado del arte y retos para la mejora de sistemas de preparación en almacenes*. Estudio Delphi.

GreyOrange (s.f.). *Butler goods to person system*. Disponible en:

<https://www.greyorange.com/butler-goods-to-person-system>

IAM Robotics (2017). *I AM Robotics awarded patent for warehouse robot*.

Disponible en: <https://roboticsandautomationnews.com/2017/09/28/iam-robotics-awarded-patent-for-warehouse-robot/14314/>

IAM Robotics (2017). *Technical Specifications*. Disponible en:

<https://www.iamrobotics.com/products/swift/>

IAM Robotics (2019). *IAM Robotics Redesigns, Expands Swift System for Mobile Fulfillment*. Disponible en:

<https://www.roboticsbusinessreview.com/events/iam-robotics-redesigns-expands-swift-system-for-mobile-fulfillment/>

ICMD (2020). *Automatizacion procesos vehiculos intralogistica*. Disponible en:

<https://www.icemd.com/digital-knowledge/articulos/automatizacion-procesos-vehiculos-intralogistica/>

Institution of Mechanical Engineers (2016). *DHL runs warehouse pilot with collaborative robots*. Disponible en: <https://fourbythree.eu/dhl-runs-warehouse-pilot-with-collaborative-robots/>

Interempresas (2010). *AGV, los vehículos industriales inteligentes*. Disponible en:

<http://www.interempresas.net/Logistica/Articulos/38400-AGV-los-vehiculos-industriales-inteligentes.html>

Locus Robotics (2018). *There's No More "I" in ROI*. Disponible en:

<https://locusrobotics.com/about-us/blog/page/3/>

Montoya, G. (2013) *Programación de microcontroladores PIC + ZIGBEE*. Colombia

Polo, D. (2016). *El caso Amazon: la gestión de almacenes automatizada*.

Disponible en: <https://www.gestionar-facil.com/el-caso-amazon-gestion-de-almacenes-automatizada/>

SSI Schaefer (2017). *Caso práctico Bachmann*. Disponible en: [https://www.ssi-](https://www.ssi-schaefer.com/es-br/sectores/caso-practico-bachmann--371090)

[schaefer.com/es-br/sectores/caso-practico-bachmann--371090](https://www.ssi-schaefer.com/es-br/sectores/caso-practico-bachmann--371090)

SSI Schaefer (2017). *MOTUM, 2PICK® Highest Efficiency in Picking*. Disponible

en: <https://www.ssi-schaefer.com/resource/blob/>

[79774/2ab421726cd66785f8b062e412cde22a/2pick-dam-download-en-2140--data.pdf](https://www.ssi-schaefer.com/resource/blob/79774/2ab421726cd66785f8b062e412cde22a/2pick-dam-download-en-2140--data.pdf)

SSI Schaefer (2017). *WEASEL, vehículo de guiado automático (AGV)*. Disponible

en: <https://www.ssi-schaefer.com/>

[resource/blob/350556/891996c4b96fa9b212c1769e1dba60af/agv-weasel-veh%C3%ADculo-de-guiado-autom%C3%A1tico-para-el-transporte-interno-de-mercanc%C3%ADas-dam-download-es-3980--data.pdf](https://www.ssi-schaefer.com/resource/blob/350556/891996c4b96fa9b212c1769e1dba60af/agv-weasel-veh%C3%ADculo-de-guiado-autom%C3%A1tico-para-el-transporte-interno-de-mercanc%C3%ADas-dam-download-es-3980--data.pdf)

The Standard Cio (2017). *Conozca los robots que DHL está utilizando en sus*

*almacenes*. Disponible en: <https://thestandardcio.com/2017/04/11/robots-dhl-en-almacenes/>

XPO Logisthic (s.f.). *Xpo-logistics utilizara 5000 robots colaborativos en sus almacenes de norteamerica y europa*. Disponible en:  
<https://es.xpo.com/es/news/xpo-logistics-utilizara-5000-robots-colaborativos-en-sus-almacenes-de-norteamerica-y-europa>