

Construcción de un prototipo para la gestión de almacén, dirigido a pequeñas y medianas empresas usando un brazo robótico antropomórfico y un robot AGV.

Daniel Fernando Rivera Rincón
Duvan Leonardo Fernandez Chaparro
Johan Camilo Espitia Camacho

Universitaria Agustiniana
Facultad de Ingenierías
Programa de Ingeniería Mecatrónica
Bogotá D.C.
2022

Construcción de un prototipo para la gestión de almacén, dirigido a pequeñas y medianas empresas usando un brazo robótico tipo puma y un robot AGV.

Daniel Fernando Rivera Rincón
Duvan Leonardo Fernández Chaparro
Johan Camilo Espitia Camacho

Director

Wilson Javier Serrano Puerto

Trabajo de grado para optar al título de Ingeniero Mecatrónico

Universitaria Agustiniiana
Facultad de Ingenierías
Programa de Ingeniería Mecatrónica
Bogotá D.C.

2022

Tabla de Contenidos

Agradecimientos.....	6
Resumen.....	7
Abstract.....	8
1. Introducción.....	9
2. Problema de investigación.....	10
2.1 Antecedentes.....	10
2.2 Descripción del problema.....	12
2.3 Árbol de problemas.....	14
2.4 Pregunta problema.....	14
3. Objetivos.....	15
3.1 Objetivo general.....	15
3.2 Objetivos específicos.....	15
4. Justificación.....	16
5. Marco referencial.....	18
5.1 Marco teórico.....	18
5.1.1 Cinemática en robots industriales.....	18
5.1.2 Dinámica de un robot.....	19
5.2 Marco conceptual.....	19
5.2.1 Sistemas de locomoción.....	19
5.2.2 Sistemas de percepción.....	20
5.2.3 Sistemas de comunicación.....	20
5.2.4 Lenguajes de programación.....	20
5.2.5 Software.....	20
5.2.6 Sistemas de vacío.....	21

5.3 Marco legal	21
6. Marco metodológico.....	26
6.1 Tipo de investigación.....	26
6.2 Variables del problema	26
6.3 Fuentes de información.....	26
6.3.1 Fuentes primarias:	26
6.3.2 Fuentes terciarias:.....	26
6.4 Instrumentos de recolección de información	27
6.5 Cronograma	28
7. Diseño del prototipo	31
7.1 Cálculos matemáticos.	31
7.1.1 Grados de libertad del brazo robótico.	31
7.1.2 Planos mecánicos brazo robótico y robot AGV.	32
7.1.3 Caracterización y peso del brazo robótico y del robot AGV.	49
7.1.4 Selección de materiales y componentes.	52
7.1.5 Pesos máximos.	60
7.1.6 Cadena cinemática.	62
7.1.7 piñón-cremallera.	66
7.2 Planos electrónicos.	70
7.2.1 Circuito brazo robótico.	70
7.2.2 Circuito robot AGV.....	74
7.3 Otros planos.	76
8. Programación y comunicación.	81
8.1 Diagramas de flujo y conexión.	81
8.2 Programación brazo robótico.....	86

8.3 programación del Robot AGV	95
8.4 Programación interfaz de usuario.	100
9. Construcción y pruebas.	113
9.1 Brazo robótico.....	113
9.2 Construcción robot AGV.	120
9.3 Errores y soluciones hallados en pruebas.	127
10. Parámetros de implementación.....	130
10.1 Recomendaciones para implementación.....	130
10.1.1 Brazo robótico.....	130
10.1.2 Estanterías.	130
10.1.3 Comunicación.	130
10.1.4 Robot AGV.	130
10.1.5 Neumática.	131
10.1.6 Ambiente de trabajo.	131
10.2 Costo de construcción prototipo a escala.....	131
10.3 Costo-beneficio.	134
Recomendaciones.	140
11. Conclusiones.....	141
12. Referencias	145

Agradecimientos.

Agradecemos a nuestros padres por el apoyo que nos han brindado durante todo nuestro proceso académico, levantando el ánimo frente a tropiezos y caídas, proporcionando todo recurso económico necesario para completar los estudios y por ser un soporte en general para todos nosotros.

Agradecemos también a nuestro tutor de trabajo de grado, Wilson Javier Serrano, por compartir su conocimiento a lo largo de nuestra formación profesional, por siempre estar disponible para resolver las dudas generadas en el lapso de este proyecto, sin importar los tiempos y por su buena disposición siempre.

Resumen.

En el presente documento se tuvo como objetivo la construcción de un prototipo a escala para la gestión de almacenes, dirigido a pequeñas y medianas empresas, utilizando un brazo robótico y un robot AGV. Como punto de partida se realizaron varias investigaciones de temas relacionados, además de recaudar información de utilidad para completar el objetivo del proyecto, se siguió la metodología de diseño design thinking, para diseñar los planos, programar el prototipo, construirlo y finalmente ponerlo a prueba, así mismo, se establecieron algunos parámetros para su debida implementación en un entorno real de almacén; el prototipo funcionó adecuadamente durante las pruebas, donde a su vez, dejó ver posibles mejoras en su diseño que se dejan a disponibilidad de futuros proyectos como la adición de un grado de libertad extra, así como realizar la construcción del sistema y probarlo en entornos reales, además de realizar una debida comparación entre almacenes en los que se implemente contra otros que no lo hayan hecho.

Palabras clave: Robot, brazo, almacenes, AGV, pedido, prototipo, eslabones, sistema.

Abstract.

In this document, the objective was to build a scale prototype for warehouse management, aimed to small and medium-sized companies, using a robotic arm and an AGV robot. As a starting point, several investigations of related topics were carried out, in addition to collecting useful information to complete the objective of the project, the design thinking methodology was followed, to make the plans, program the prototype, build it and finally test it. Likewise, some parameters were established for its proper implementation in a real warehouse environment; the prototype worked properly during the tests, where it also revealed possible improvements in its design that are left available for future projects such as the addition of an extra degree of freedom, as well as building the system and testing it in real environments, in addition to making a comparison between warehouses in which it is implemented against others that have not done it.

Key words: Robot, arm, warehouses, AGV, order, prototype, link, system.

1. Introducción

Por medio de este documento se plantea la construcción de un prototipo para facilitar la logística en almacenes de pequeñas y medianas empresas. El prototipo consiste en un brazo robótico encargado de organizar los paquetes solicitados por un operario sobre un robot AGV que se encargará del transporte y entrega de dichos paquetes. Así mismo, ambas partes son capaces de recibir y enviar información entre ellas, para garantizar un trabajo logístico de calidad dentro del almacén.

Para llevar a cabo este trabajo se realizó una búsqueda previa de documentos donde se abarcan temas con relación a la gestión de almacenes empleando robótica industrial o robots móviles tipo AGV, es entonces que, al evidenciar que estas máquinas están mayormente dirigidas a grandes empresas y hay una falta de opciones existentes para las pequeñas y medianas empresas, se decidió abordar esto mismo como el problema central de este documento y por consiguiente, se dio el planteamiento de una pregunta problema que fue utilizada para proponer el objetivo del presente trabajo de investigación y los pasos necesarios para cumplirlo.

Una vez planteado lo anterior, y con el fin de obtener unas buenas bases teóricas para la construcción del prototipo, se realizó una investigación acerca de los fundamentos teóricos, los conceptos clave y los aspectos legales que rodearán la investigación.

2. Problema de investigación

2.1 Antecedentes

A continuación se presentan los antecedentes relacionados a la construcción de prototipos de sistema para la gestión de almacén, donde se encuentran aspectos a tener en cuenta para la construcción de robots, como los sistemas de navegación. Los siguientes trabajos hacen énfasis en la logística en almacenes por medio de robots industriales de tipo AGV (Automated Guided Vehicle) y brazo robótico. En los siguientes párrafos, se presentarán proyectos relacionados a la investigación.

En primer lugar, se puede encontrar el trabajo realizado por Guillen, que se enfocó en demostrar que el proceso de picking es de vital importancia, ya que, este indica el inicio del suministro de productos por parte de la empresa, con ello representa un flujo constante de ingresos, para ello, se buscó la manera más eficaz para la automatización de este proceso productivo, Guillen (2020) propone que la solución se divide en 3 subsistemas: el de desplazamiento, el de extracción de productos y el de registro y programación de pedidos. Donde se especifica la importancia de un buen diseño a la hora de plantear un prototipo para la excelente realización de la tarea a la que fue planeado.

En segundo lugar, se toma como referencia la investigación realizada por Puerta y Rodríguez (2021), los cuales realizaron un paralelismo exponiendo su planteamiento sobre el cómo pequeñas y medianas empresas afrontan los problemas vividos en tiempos de pandemia sin haber contado con sistemas de gestión, por otra parte, relataban el cómo las grandes empresas, que ya contaban con dichos sistemas lograron sobrellevar esta crisis. Con relación a las macroempresas, estas utilizaban la tecnología para mejorar en aspectos relacionados a la productividad y la eficiencia de los sistemas de gestión de almacenes, pues al implementar tecnologías de comunicación se genera un gran gasto a modo de inversión, pero a su vez, trae consigo ventajas evidentes, además, a largo plazo, el éxito de una empresa dependerá de cómo se gestionan sus almacenes y mercancía. Con relación a lo anterior, este aspecto tomó más relevancia a causa de la pandemia vivida durante el año 2020, el uso de la transmisión de información vía internet agilizó los procesos y procuró el cuidado de los empleados.

Posteriormente, al conocerse más a fondo sobre el tema, se justifica la utilidad de esta tecnología en diferentes prácticas de la industria donde los sistemas de control en vehículos AGV mayormente son usados en dos ámbitos, por una parte, empleados en el transporte de materia prima al interior

de plantas y fábricas, en segundo lugar, son utilizados en líneas de ensamblaje, gracias a su eficacia y fácil manejo, así mismo, la industria automotriz es líder en el uso de robots AGV, no obstante, esto no significa que no se utilicen en otras industrias, ya que son igualmente usados en centros de distribución, almacenes, así como en la industria del papel, impresión y siderúrgica (García, 2019).

Eventualmente se observó, desarrolló y analizó su funcionalidad a través de un modelado matemático donde se demostró la complejidad de este sistema y la eficacia de un modelo ya comercializado. Con base a esto, se busca la simplificación del sistema para una mejor implementación en pequeñas y medianas empresas.

Ahora bien, al realizar una investigación a nivel nacional, se observó que no se han realizado investigaciones y avances respecto a la implementación de robótica industrial y tecnología AGV en el área de automatización de almacenes y bodegas de paquetería, con la excepción de Arbeláez (2021) que propone un modelo de vehículo AGV controlado con arduino inspirado en un modelo pre-existente, que a su vez podrá ayudar a futuros estudiantes proporcionándoles un punto de partida donde estos logren realizar la construcción de vehículos auto guiados.

2.2 Descripción del problema

Colombia es un país cuyo Producto Interno Bruto (PIB), según el Ministerio de Trabajo (2019), está constituido en un 35% por las pequeñas y medianas empresas, así mismo, estas son las que tienen un menor nivel de desarrollo en cuanto a automatización se refiere, y por ende, tal como lo mencionan Rojas y Urián (2020), las que presentan una menor eficiencia en los procesos productivos, en especial en la gestión del almacenamiento, por lo que son comunes los casos donde los productos son entregados con desperfectos, incompletos o incluso no hayan sido entregados tal y como lo menciona revista Semana (2020), entre el 20 de marzo y el 20 de julio del 2020 se reportaron aproximadamente 3400 denuncias a causa de retrasos en entregas, condiciones contractuales, entre otros. En consecuencia, se observó un bajón de confianza en las empresas y por lo tanto, una menor obtención de ganancias.

No obstante, algunas empresas, en lugar de realizar una actualización completa en las operaciones, deciden emplear ciertos grados de automatización con el fin de reducir costos, incrementar la seguridad y productividad dentro del almacén, además de lograr una mayor satisfacción por parte de los clientes (Mecalux, 2019). Sin embargo, al implementar estos sistemas se generan varios costos indirectos que las PYMEs (Pequeñas Y Medianas Empresas) no suelen tener en cuenta, ocasionando que algunos sistemas de automatización no sean rentables para las mismas, entre los costos extras, se encuentran: el soporte o mantenimiento que se debe prestar a la maquinaria cada cierto periodo de tiempo, actualizaciones a los sistemas electrónicos y programaciones, mejoras constantes que debe recibir el equipo utilizado, entre otras (Hernández, 2019).

De manera similar, en Colombia, al existir poca demanda por parte de las PYMEs en la implementación de tecnologías para la automatización, su oferta en el campo de las máquinas y los sistemas es poca y por ende, el precio de las mismas puede aumentar. Pues, mientras que en países como Japón, Corea o Alemania existen desde 300 hasta 450 robots industriales por cada 10.000 trabajadores, en promedio y con relación a lo anterior, Colombia cuenta con sólo un robot o menos dentro de su logística de trabajo, viéndose que para el año 2017 sólo existían 2 ensambladoras con menos de 55 robots (Metalmecánica, 2017). Es debido a esto que, en la encuesta realizada por PWC (por sus siglas en inglés PriceWaterhouseCoopers) en el año 2021, aproximadamente el 74% de los directores ejecutivos en la nación, han mostrado su preocupación frente a la velocidad del cambio tecnológico (PWC, 2021).

Así mismo, el poco uso de estos sistemas en el país, evidencia la poca información disponible que se tiene de estos sistemas, pues, así como lo mencionan Carrero y Sastre (2021), en Colombia las pequeñas y medianas empresas no implementan tecnologías para realizar control de sus inventarios por falta de conocimientos, metodologías guías y estudios que demuestren el retorno de inversión.

2.3 Árbol de problemas.

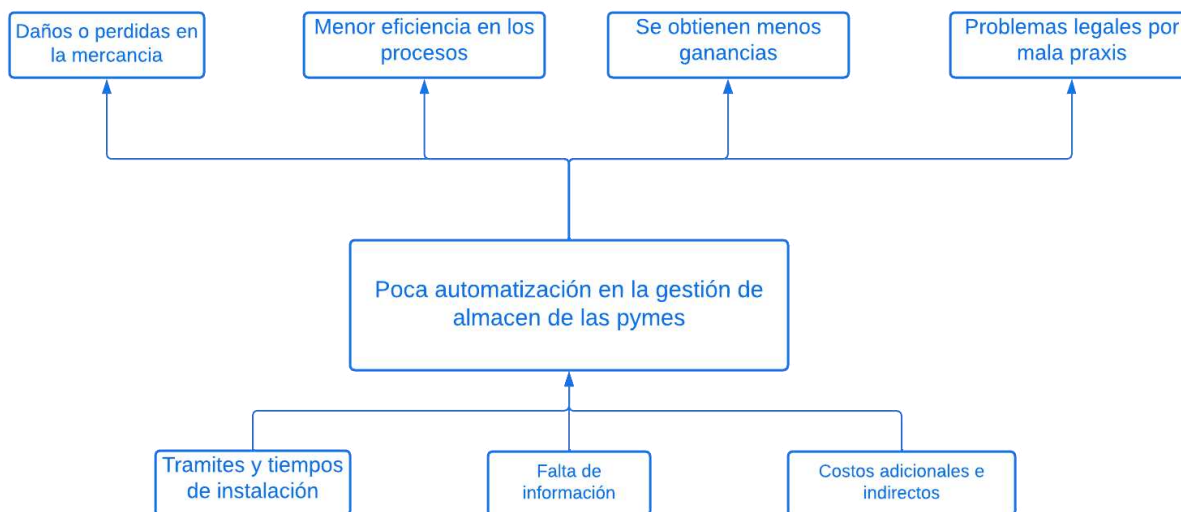


Figura 1. Árbol de problemas. Elaboración propia (2022).

2.4 Pregunta problema

¿Cómo construir un prototipo asequible que contribuya a la automatización y gestión de almacenes para pequeñas y medianas empresas utilizando un brazo robótico y un robot AGV?

3. Objetivos

3.1 Objetivo general

- Fabricar un prototipo a escala, funcional y asequible para la gestión de almacenes de pequeñas y medianas empresas usando un brazo robótico antropomórfico y un robot AGV.

3.2 Objetivos específicos

- Realizar los planos mecánicos y electrónicos de un prototipo de gestión de almacenes con base en un robot antropomorfo y un robot AGV para tener una vista previa de la construcción de ambos robots.

- Desarrollar la programación mediante lenguajes de programación que sean favorables para la comunicación y funcionamiento del prototipo.

- Construir el modelo a escala en base a los diseños y el programa realizado, para desestimar errores en el diseño y programación previos mediante diversas pruebas.

- Establecer parámetros especificando las condiciones de operación y requisitos mínimos para su debida implementación en pequeñas y medianas empresas.

4. Justificación

Este proyecto se realiza con el objetivo de construir un prototipo para la gestión de almacenamiento que al implementarse permita a las pequeñas y medianas empresas aumentar su eficiencia y productividad en el desarrollo de actividades de acción repetitiva, como lo son la clasificación y transporte de productos en el interior de almacenes, pues el 95 % de las empresas que optaron por la automatización robótica de procesos (RPA por sus siglas en inglés *Robotic Process automation*) afirman que se ha visto una mejoría en su productividad, el 93% ha evidenciado un incremento en su cumplimiento, el 81% presentan una reducción de costos y el 77% muestran un mejor manejo de la información (Hernandez, 2019).

Es entonces que, al hacer uso del prototipo en la logística de los almacenes, estos podrán disminuir gastos por productos en mal estado o extraviados a causa de mala praxis o agotamiento físico del personal, además de posibles problemas legales que surjan a raíz de accidentes laborales como caídas, cortes, golpes o atropellos, entre otros (Mecalux, 2019). Como ejemplo de lo anterior, entre el 20 de marzo y el 20 de julio del año 2020 surgieron alrededor de 3.400 denuncias en la Superintendencia de Industria y Comercio de Colombia, por fallos en las entregas debido a problemas relacionados con la logística de varias empresas (Revista semana, 2020).

De la misma manera el uso del prototipo planteado en este documento ayudará a minimizar casos, como lo acontecido en la bodega de Mercado libre en México, donde diversos vendedores aseguraron que las ganancias disminuyeron debido a las pérdidas de sus productos en dicha bodega, ya que al tratarse del centro logístico Tepetzotlán, donde trabajan aproximadamente 1.400 personas, dicho incidente dio paso a una amplia cantidad de errores humanos y por consiguiente a quejas y reclamos dirigidas a mercado libre (Saldaña, 2021).

En síntesis, al implementar este proyecto en las PYMEs, se minimizarán los errores humanos dentro de los procesos de logística, al igual que las cargas de trabajo sobre el personal y los riesgos que corren al realizar sus labores, gracias a esto no sólo se generaría una mejora en los tiempos de transporte de mercancía al interior del almacén, sino que a su vez, los gastos y pérdidas económicas se reducirán y las ganancias de la empresa aumentarán.

Finalmente, cabe recalcar que para llevar a cabo la construcción del prototipo propuesto en este proyecto, se necesitan de diversas habilidades, como lo son el diseño en 3D, la construcción mecánica, el control, la programación y la electrónica, por lo que se hace indispensable que el proyecto sea desarrollado por un Ingeniero Mecatrónico, pues este debe contar con el conocimiento

de cada área anteriormente mencionadas y a su vez, la capacidad de desempeñarse adecuadamente en las mismas.

5. Marco referencial.

5.1 Marco teórico

5.1.1 Cinemática en robots industriales.

Al construir un robot industrial es importante conocer la capacidad de movimiento que su estructura le permite ya que actualmente estas máquinas suelen tener una estructura compleja, según Kumar (2008) “Para que un robot ejecute una tarea específica, primero que nada, deberá establecerse la posición y la orientación del efector final, es decir, su posición o configuración en relación con su base. Esto es esencial para resolver problemas de posicionamiento” (p.112).

Aquí es donde la matemática interviene por su capacidad de ilustrar de manera entendible de cómo funciona el movimiento de estos, tal y como Kumar (2008) comenta:

En el análisis de posición, se encuentra una relación entre las coordenadas cartesianas, es decir, la posición de un punto en el efector final y su orientación con los ángulos de las articulaciones. Aquí existen dos tipos de problemas: la cinemática directa y la inversa (p.112).

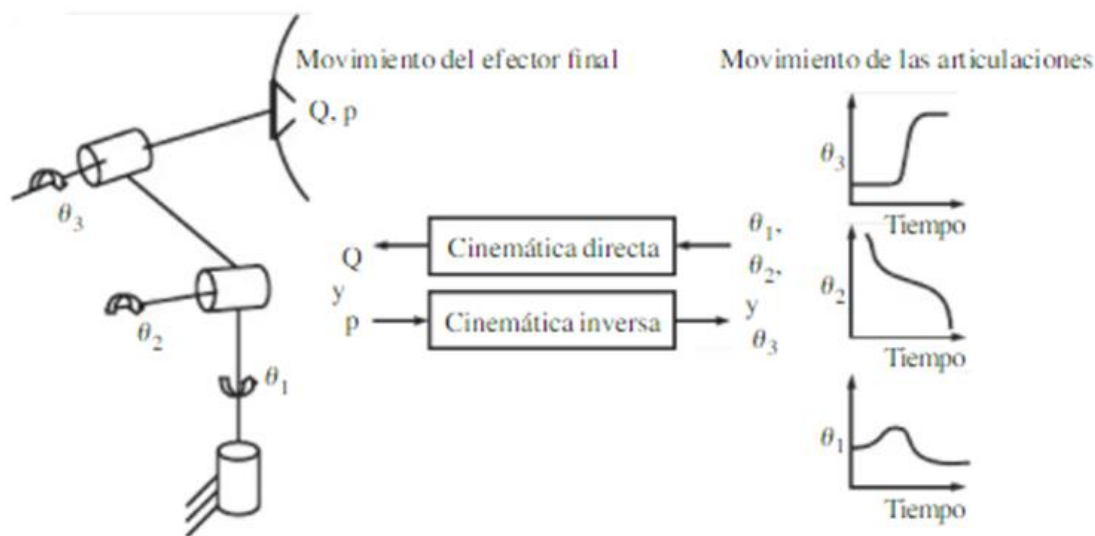


Figura 2. Cinemática directa e inversa. (Kumar, 2008, p.113).

5.1.1.1 Notación de Denavit y Hartenberg. Para establecer la cinemática existente entre los eslabones de un brazo robótico, es necesario tener en cuenta y aplicar el método de Denavit y Hartenberg que para Villanueva (2018):

La notación de Denavit y Hartenberg, proporciona un método sistemático para asignar un sistema de referencia cartesiano trirrectángulo y dextrógiro a cada barra que forma parte de brazo robot. Gracias a ello se podrán describir las relaciones cinemáticas entre elementos del manipulador mediante una serie de transformaciones homogéneas como ya se ha expresado anteriormente. Dicho procedimiento hace

uso de un número mínimo de parámetros para definir completamente las relaciones cinemáticas, así mismo evita indefiniciones en estos casos de cadenas cinemáticas abiertas. (p.15)

5.1.2 Dinámica de un robot.

La dinámica de robots, ayuda a establecer una relación entre el movimiento del robot y las fuerzas que actúan en el cuerpo del mismo (Ramírez, 2012, p.2), en el caso de un brazo robótico, al establecer su dinámica, se puede identificar las fuerzas que actuaran en cada una de las secciones del brazo dependiendo de los movimientos que estas hagan, por otra parte, el modelado dinámico de un robot de uno o dos grados de libertad no resulta ser complejo, pero, a medida que se aumente el número de grados de libertad del robot, la complejidad de su modelo dinámico también aumentará generando la necesidad de usar métodos computacionales para resolver sus ecuaciones matemáticas (Legarreta y Martínez, 2017, p.3).

5.1.2.1 Formulación Lagrange-Euler. Este método permite realizar el modelado dinámico de robot teniendo en cuenta la energía cinética y potencial almacenada, ya que para el método de Lagrange-Euler se considera al robot como una caja negra (Legarreta y Martínez, 2017, p.4).

$$\begin{aligned} \text{Langrangiano } L(q, \dot{q}) &= K(q, \dot{q}) - U(q) \\ \text{Ecuación de movimiento } \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} &= \tau \end{aligned} \quad (1)$$

5.1.2.2. Formulación Newton-Euler. Con este método se puede realizar el modelado dinámico de un robot efectuando un balance de fuerzas y momentos, para esto es necesario formular las ecuaciones del movimiento lineal y el angular de cada uno de los eslabones del robot (Legarreta y Martínez, 2017, p.4).

$$\begin{array}{cc} \text{Newton} & \text{Euler} \\ \sum F = m\dot{v} & \sum T = I\dot{\omega} + wx(I\omega) \end{array} \quad (2)$$

5.2 Marco conceptual

5.2.1 Sistemas de locomoción

Los sistemas de locomoción, son un conjunto de elementos que le brindan estabilidad y a su vez hacen posible el movimiento de robots móviles por determinados entornos. En este sentido, el objetivo de la locomoción es estabilizar el robot mientras se desplaza, reaccionando a fenómenos y propiedades como la fuerza gravitatoria, la inercia y las cargas de trabajo, por último, para que todo esto sea posible el sistema debe estar adaptado para el terreno en el que se desplazará la máquina (Barquero, 2020, p.8-9).

5.2.2 Sistemas de percepción

Son sistemas que toman información por medio de sensores (exteroceptivos y propioceptivos), esto es recolectado tanto del entorno exterior como de los sistemas interiores que posee el robot, para luego poder ser convertido en datos que el robot pueda comprender, de esta forma, la máquina puede tomar decisiones de manera autónoma con respecto al entorno y a su propio estado, tales como detenerse, cambiar o modificar su ruta, potenciar sus sistemas de ventilación o simplemente, almacenar o transmitir los datos que percibe (Rico y Escobar, 2021, p.35).

5.2.3 Sistemas de comunicación

Debido al gran número de tareas críticas o cotidianas que actualmente realizan los robots como son el mapeo, limpieza y transporte, esto ha generado la necesidad de hacer que los robots puedan comunicarse entre sí para realizar dichas labores de una manera más rápida al trabajar en equipo y de forma sincronizada; actualmente existen varios sistemas de comunicación, entre los cuales se pueden encontrar: comunicación por radiofrecuencia, comunicación ultrasónica y comunicación láser. De todos los sistemas de comunicación existentes en la actualidad, cada uno “tiene bondades y limitantes propias de las características del canal, pero todos han demostrado su viabilidad en distintas situaciones” (Fields, 2016, p.22).

5.2.4 Lenguajes de programación

La programación en robótica no es muy distinta a la usada en otros campos, “normalmente los lenguajes utilizados son genéricos, es decir, se usan en otras aplicaciones informáticas, y la parte específica de robótica se encapsula en bibliotecas u objetos particulares” (Cañas, 2004, p.10). No obstante, Cañas (2004) considera que ha habido intentos para utilizar lenguajes de programación exclusivamente para robótica como pueden ser: Task Description Language (TDL) de Simmons o Reactive Action Packages (RAP) de Friby (p.10).

5.2.5 Software

Es el conjunto de programas que hacen parte del sistema operativo de una máquina, los mismos, según sea su función, realizan una o más acciones, así como tomar datos de sensores y controlar actuadores, esto según Cañas (2004):

Una colección de drivers con rutinas para leer datos de los sensores y enviar consignas a los actuadores. En este contexto, el programa de aplicación leía las medidas obtenidas por los sensores y escribía los órdenes de movimiento a los motores invocando directamente las funciones (p.7).

5.2.6 Sistemas de vacío

Los sistemas de vacío son circuitos neumáticos utilizados en procesos donde se requiera agarrar un objeto o una pieza para transportarla a alguna otra parte de dicho proceso, para esto comúnmente se utilizan ventosas que, gracias a una diferencia de presiones entre el interior y el exterior de la ventosa, esta se adhiere fácilmente al objeto o pieza que se necesite. Para lograr que se genere una diferencia de presiones, es necesario contar con un generador de vacío, ya que este es el responsable de succionar todo el aire existente en el interior de la ventosa (Revista de Robots, 2022).

5.2.6.1 Ventosas de vacío. Las ventosas de vacío son las encargadas de la sujeción de las piezas que se quieran manipular durante el proceso. Para asegurar que la ventosa no dañe o rompa la pieza al momento de sujetarla, se debe calcular la fuerza de sujeción de la misma (Schmalz, s.f).

$$f = \Delta p * A \quad (3)$$

f = Fuerza de retención.

Δp = Diferencia entre presión ambiental y la presión del sistema.

A = Superficie de aspiración efectiva (superficie efectiva, cargada de vacío, de una ventosa)

5.2.6.2 Generador de vacío. El generador de vacío es el componente encargado de generar el vacío entre la ventosa y la pieza que se esté manipulando dentro del sistema neumático, para esto el generador de vacío succiona el aire presente entre la ventosa y la pieza, permitiendo que la ventosa se adhiera firmemente. Además los generadores de vacío pueden ser neumáticos o eléctricos, donde los neumáticos destacan por ser más compactos y ligeros, y los eléctricos destacan por su mayor potencia de aspiración (Schmalz, s.f).

5.2.6.3 Electro válvulas. Las electroválvulas se encargan de controlar el aire comprimido o el vacío, estas válvulas funcionan gracias a señales eléctricas de conmutación (Schmalz, s.f).

5.3 Marco legal

A continuación se presenta la tabla que contiene las normativas y estándares que se deben de seguir y tener en cuenta a la hora de diseñar y fabricar tanto robots industriales como el brazo puma como robots tipo AGV.

Tabla 1.

Marco legal

Jerarquía	Número	año	Título	Artículos que aplican	Alcance de la norma
Comisión electrotécnica internacional	IEC60204-1	2016	Seguridad en equipo electrónico	Parte 5, 6, 7, 9, 13, 14, 15, 17, Anexo C, Anexo D	Conexiones de alimentación, protección contra choques eléctricos, protección del equipo, funciones de mando, conductores y cables, cableado, motores y equipos asociados, señales de emergencia, protección contra la sobre intensidad de los conductores y cables en el equipo, explicación de las funciones de operaciones de emergencia

Asociación española de normalización	UNE-EN ISO 3691-4	2020	Carretillas de manutención: carretillas industriales sin conductor y sus sistemas	Anexo A, Anexo B, Anexo C, Anexo D, Anexo E	Requisitos para la preparación de las zonas de operación, lista de peligros significativos, determinación de capacidad nominal, operaciones de transferencia de carga, verificación de los requisitos esenciales de salud y seguridad
	UNE-EN ISO 3691-2	2016	Carretillas de manutención: carretillas autopropulsadas de alcance variable	Parte 4.1, 4.2, 4.3, 4.4, 4.6, 4.8, Anexo A, Anexo B	Requisitos de seguridad o medidas de protección, movimiento, frenos, control manual, sistemas de alcance, estabilidad, determinación de capacidad nominal, lista de peligros significativos

UNE-EN ISO 3691-6	2013-2016	Carretillas de manutención: carretilla transportadoras de carga y de personas	Parte 4, 6, Anexo A	Requisitos de seguridad, información para el usuario, lista de peligros significativos
UNE-EN 1175-1	2011	Seguridad de las carretillas de manutención: requisitos generales para carretillas alimentadas por baterías	Parte 5, 7, Anexo A, Anexo B, Anexo ZA	Requisitos generales, información para la utilización, conectores para baterías de tracción, motores eléctricos, reglas de potencia y ensayo, capítulos de la norma europea relacionados con los requisitos esenciales.

	UNE-EN ISO 9283	2003	Robots manipuladores industriales, criterios de análisis de prestaciones y métodos de ensayo relacionados	Parte 6, 7, 8, 9, 10, 11, Anexo A, Anexo B	Características principales de funcionamiento, especificaciones para la realización de los test que debe aprobar el robot industrial, recomendaciones para asegurar que el robot cumpla con los test, ejemplos prácticos de informes de resultados.
--	--------------------	------	---	---	---

Nota. Elaboración propia (2022).

6. Marco metodológico

6.1 Tipo de investigación

La investigación que se llevará a cabo será de tipo cuantitativo ya que se trabajará con variables numéricas como lo son las dimensiones de los productos que cargarán el robot AGV y el brazo robótico, así como su peso. También se tendrá en cuenta la distancia que recorrerá el robot y la gama de color que ayudará a guiarlo a su destino; además, tendrá un alcance explicativo ya que se construirá un prototipo con el cual se harán pruebas para medir su funcionamiento.

La estrategia a seguir será observacional ya que se carece de un ambiente real de trabajo en el cual recolectar datos de funcionamiento en campo, al recolectar datos de varias variables en un corto periodo de tiempo el trabajo de investigación seguirá una secuencia temporal transversal.

6.2 Variables del problema

Mediante la siguiente tabla se presentará la operacionalización de las variables que maneja el prototipo.

Tabla 2.

Operacionalización de variables

Valores	Indicadores	Valores finales	Tipo
Dimensiones	Largo, Ancho, Alto	6x6x6 cm	Continua
peso	Peso	3Kg	Continua
Distancia	Distancia	5m	Continua
Color	Gama de colores	12R, 22G, 33B	Discreta

Nota. Elaboración propia (2022).

6.3 Fuentes de información

A continuación, se presentarán las fuentes de información:

6.3.1 Fuentes primarias:

Las fuentes de información serán observacionales pudiendo evidenciar las dimensiones de los elementos que maneja el prototipo, así como también su peso; también se observarán la distancia y la gama de colores que definirán la posición dentro de la trayectoria del prototipo.

6.3.2 Fuentes terciarias:

A la hora de emplear e investigar acerca de datos teóricos y/o conceptuales en el proyecto, se tomarán como referencia documentos, informes, libros, revistas, entre otras fuentes externas de información.

6.4 Instrumentos de recolección de información

Con el fin de conocer la capacidad de carga del robot industrial puma y del robot AGV, los objetos de prueba que ambos robots transportarán serán medidos y pesados, eso se hará empleando un flexómetro y una báscula (marcas de los instrumentos por definir). Así mismo, para conocer la longitud de los trayectos que el robot AGV transitará, se empleará también un flexómetro. Finalmente, con el fin de conocer el color, tanto de los objetos a separar y transportar, como de las marcas en el suelo que guiarán al robot AGV, se emplearán sensores de color tcs3200, conectados a placas controladoras que enviarán la información a bases de datos en la nube.

7. Diseño del prototipo

Para realizar el diseño del prototipo se optó por seguir la metodología de design thinking y sus 5 pasos, los cuales son: investigación, definición, ideación, prototipado y validación. El primer paso (investigación) se puede encontrar tanto en los antecedentes como en los marcos teórico y conceptual, por otro lado el paso 2 (definición) puede encontrarse en el marco metodológico, el paso 3 (ideación) empieza desde este apartado y llegará hasta el final del mismo, mientras que, el paso 4 (prototipado) tendrá lugar desde el apartado de programación hasta la construcción del prototipo, y por último, el paso 5 (validación), se llevará a cabo con las pruebas de funcionamiento que se realizarán y las recomendaciones para la implementación en un entorno real del prototipo.

7.1 Cálculos matemáticos.

Para dar inicio a la construcción del prototipo, primero se debe contar con una guía visual que ayude a determinar la forma que deben tener los dos robots los cuales conforman este proyecto, por lo que es necesario generar planos tanto mecánicos como electrónicos del brazo robótico y del robot AGV, no obstante, incluso antes de empezar a hacer los planos de ambos robots, es necesario conocer factores como:

- Número de grados de libertad que tendrá el brazo robótico
- Longitud de cada eslabón del brazo robótico

Por otro lado otros factores que se deben conocer para comprender de mejor manera los robots que conforman el prototipo son:

- Peso de cada robot
- Torque de los motores del robot AGV
- Peso máximo que soportaran ambos robots
- Cadena cinemática del brazo robótico
- Cálculos piñón cremallera.

Por lo que al obtener los resultados para dichos factores, no solo se podrán construir planos que puedan funcionar adecuadamente como guía para la construcción del prototipo, sino que se tendrá un mejor entendimiento sobre los límites del mismo.

A continuación se presenta la resolución de los factores ya mencionados.

7.1.1 Grados de libertad del brazo robótico.

El brazo robótico utilizado para el prototipo fue construido con base a un robot puma, no obstante, si bien los robots puma cuentan generalmente con 6 grados de libertad, el cuerpo del robot

presentado con el prototipo cuenta con solo 4 grados de libertad desde su base hasta la ventosa utilizada para el proceso de pick and place, ya que no se tuvo la necesidad de agregar los dos grados faltantes en la ventosa que le permitirían girar sobre sí misma y en uno de sus ejes, por otro parte se decidió agregar un grado de libertad prismático que le permitiera desplazarse en línea recta y así facilitar la tarea de agarrar los pedidos y dejarlos sobre el robot AGV por lo que el brazo robótico quedará finalmente con 5 grados de libertad.

7.1.2 Planos mecánicos brazo robótico y robot AGV.

El brazo robótico planteado cuenta con 5 eslabones, donde 4 de ellos se encuentran unidos por articulaciones de revoluta y uno por una articulación prismática, las longitudes de cada eslabón se eligieron en base a replicar de la mejor manera la estructura de un robot puma, dando como resultado una base de 25.5 cm, un brazo de 38.5 cm (al contar con un contrapeso que evita la inclinación del brazo hacia delante), un antebrazo de 17.3 cm y una mano de 3.8 cm aproximadamente, dichas distancias se pueden apreciar en los planos del brazo robótico junto con otras medidas importantes. Ver figuras 3-10.

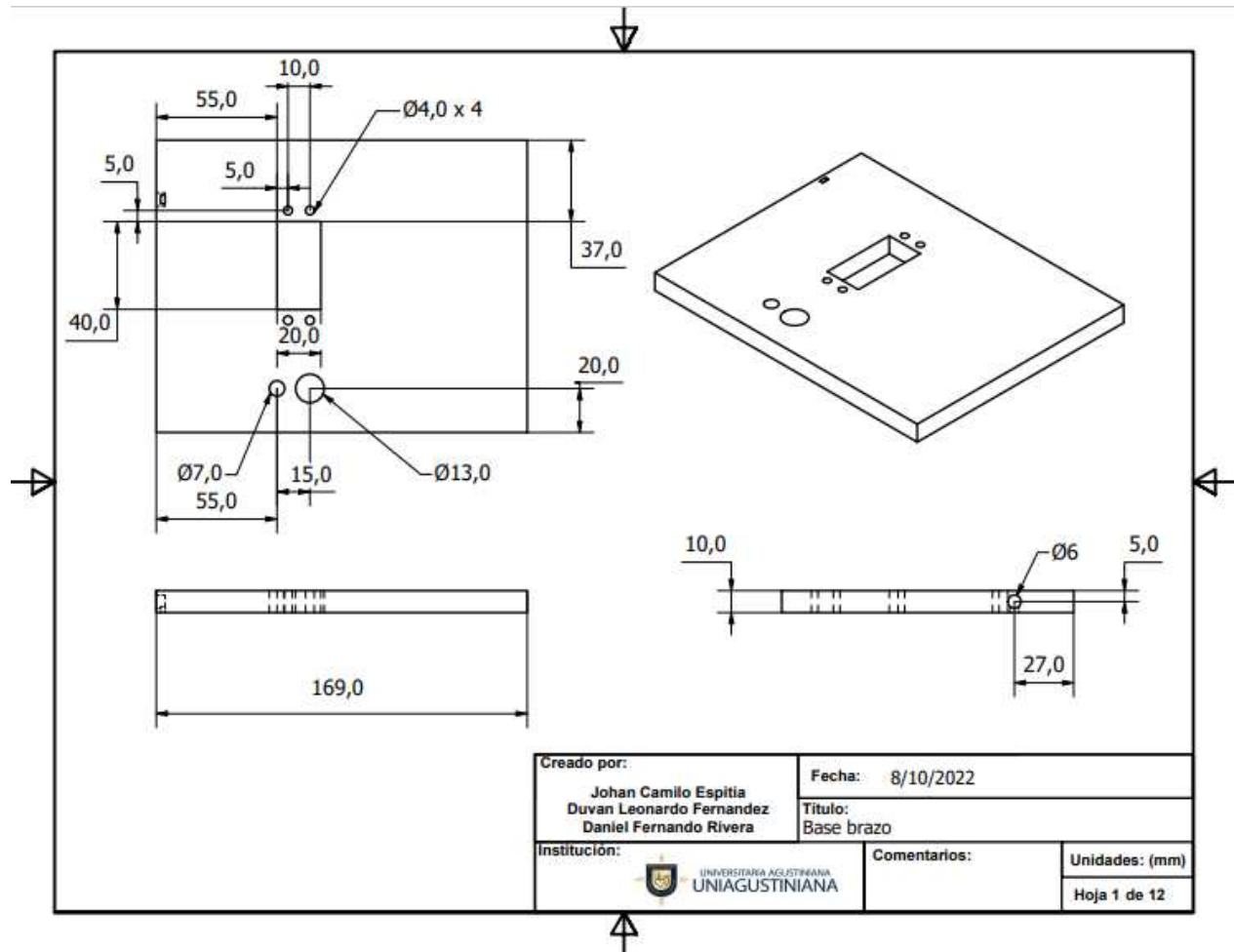


Figura 3. Base del brazo robótico. Elaboración propia (2022).

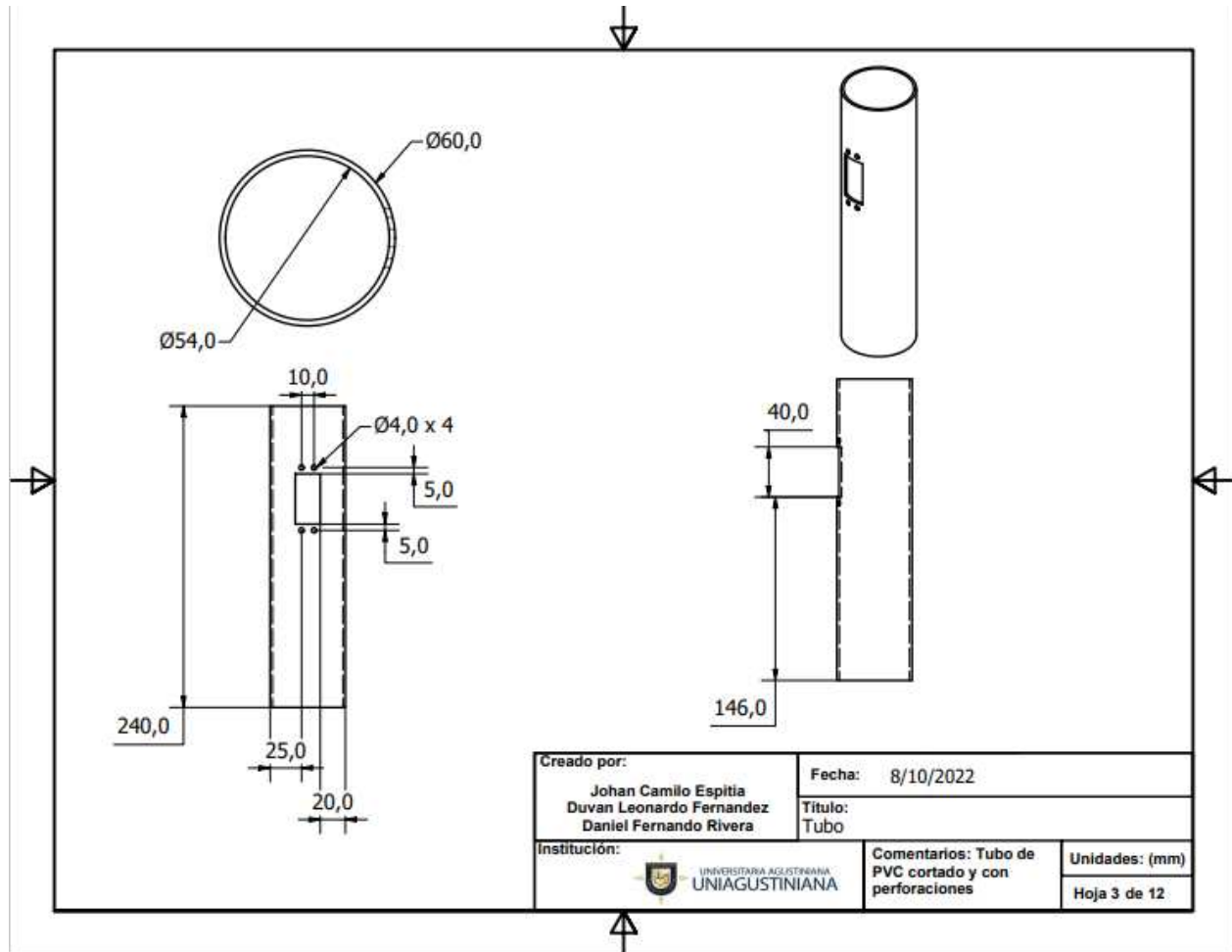


Figura 4. Torso del brazo robótico. Elaboración propia (2022).

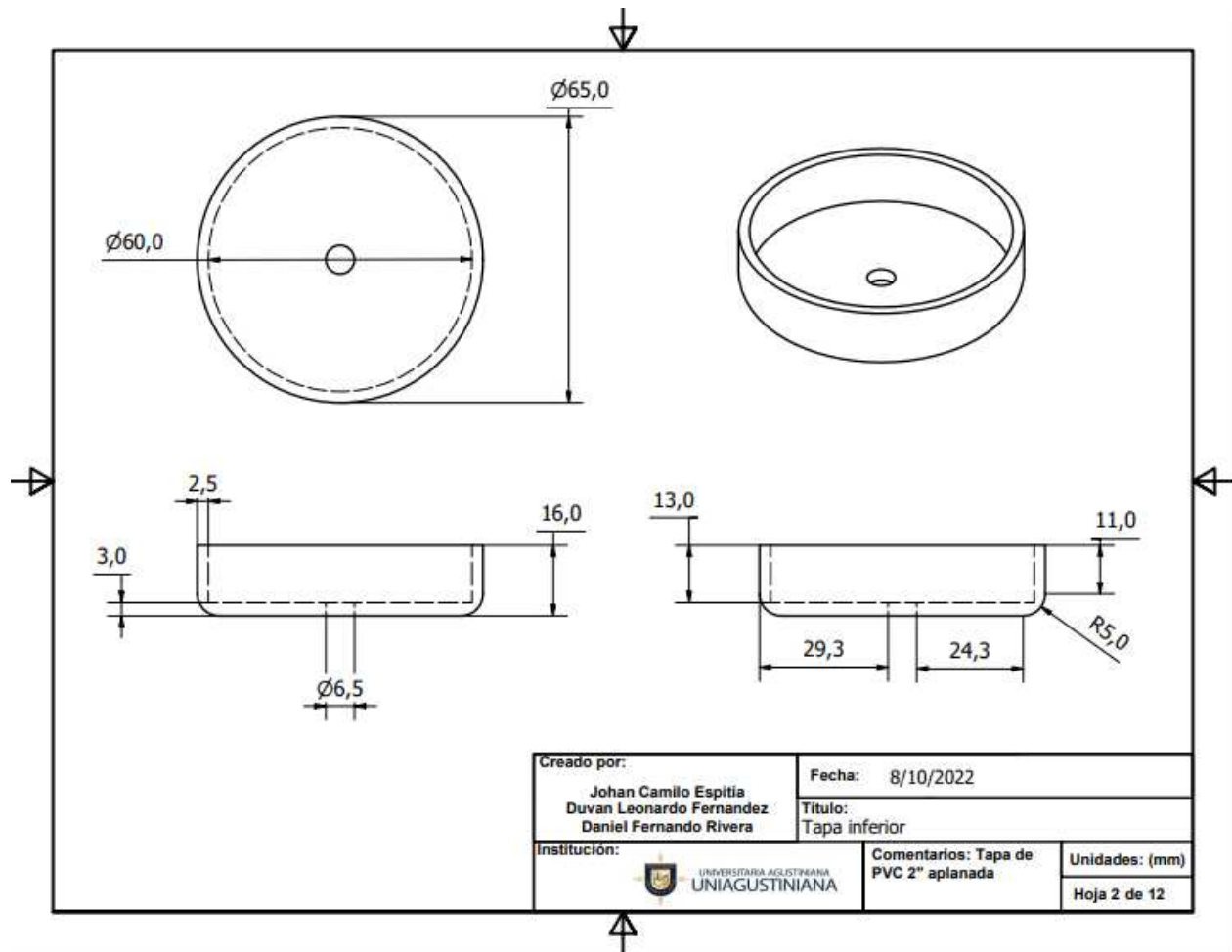


Figura 5. Tapa inferior del torso. Elaboración propia (2022).

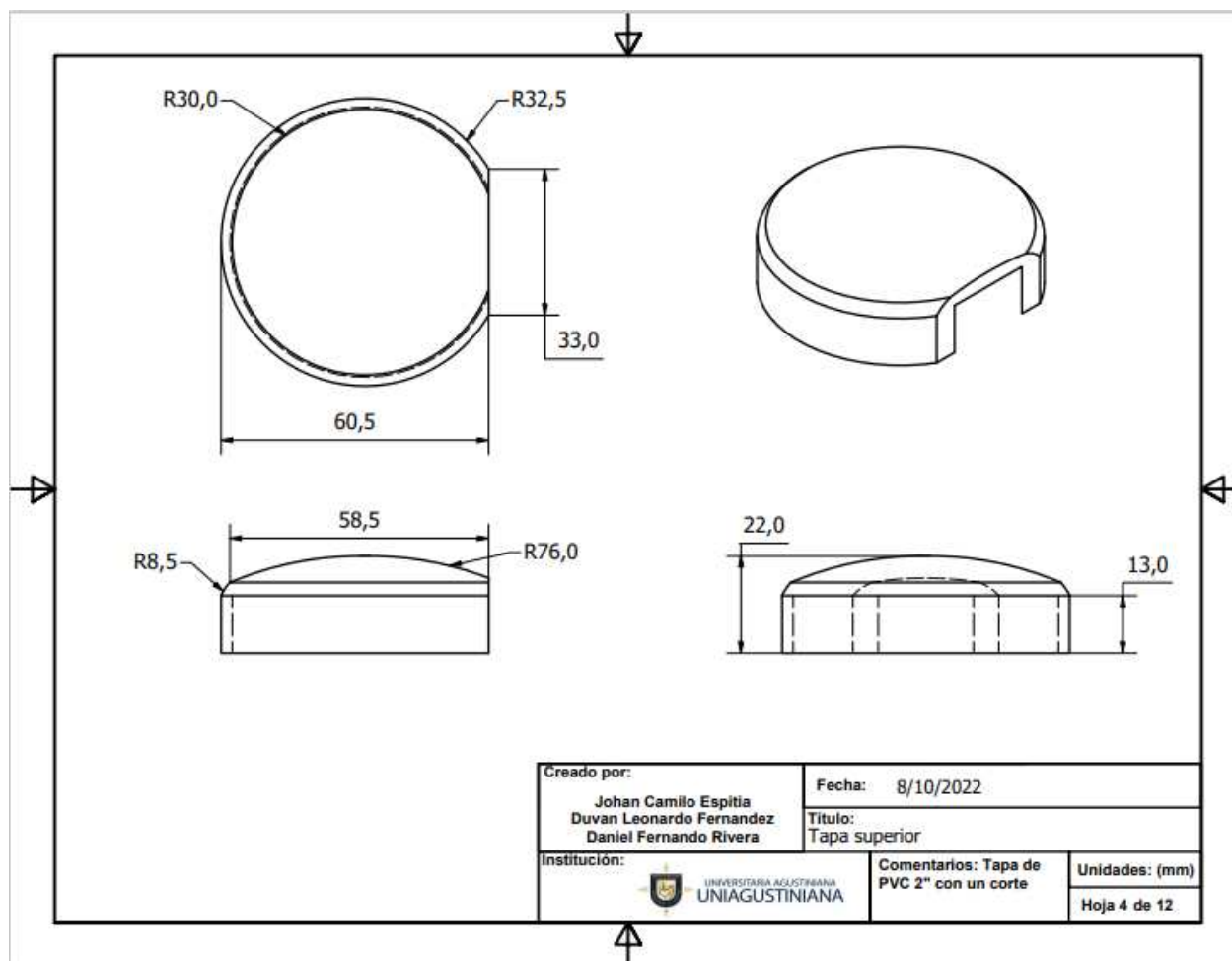


Figura 6. Tapa superior del torso. Elaboración propia (2022).

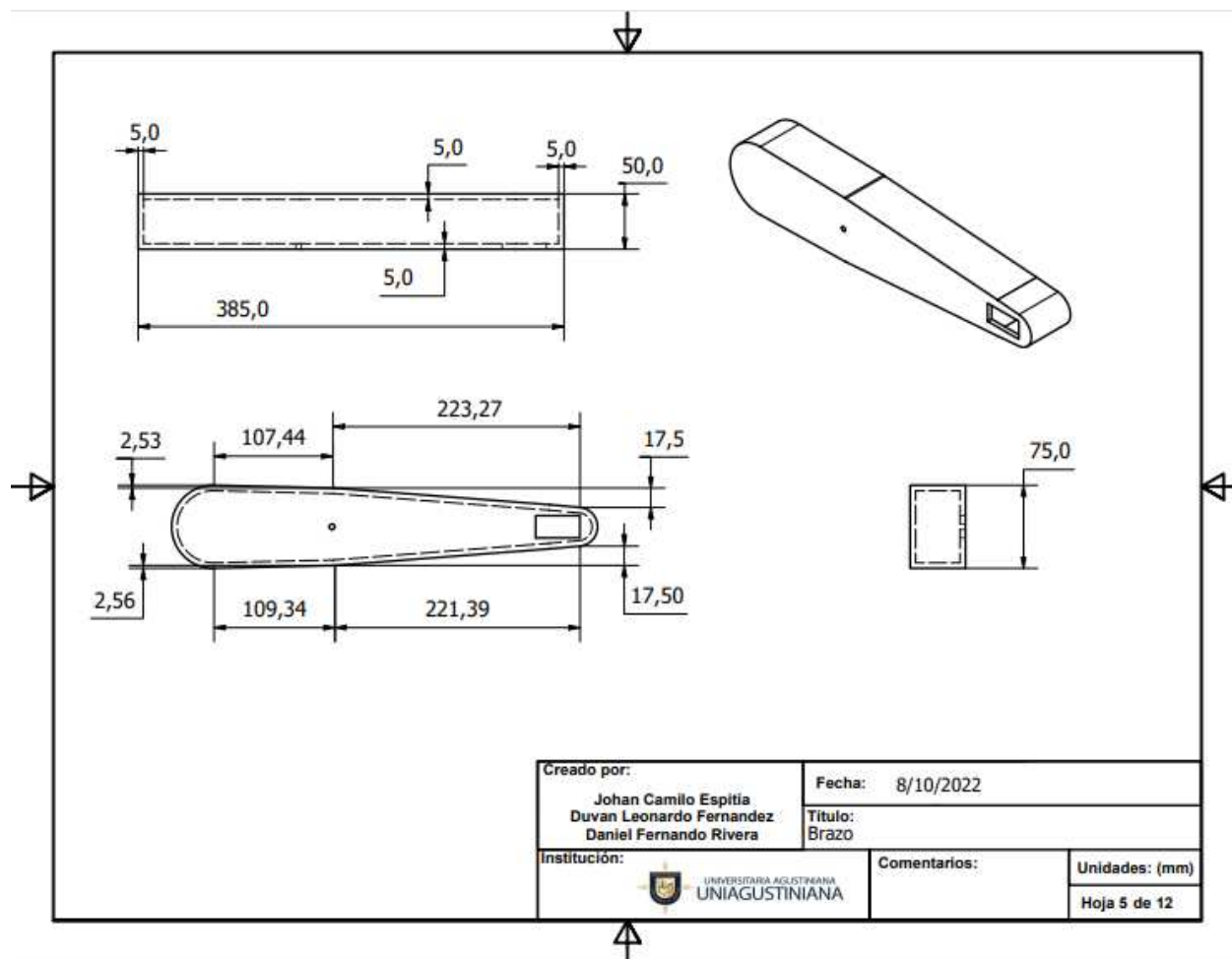


Figura 7. Brazo del brazo robótico parte 1. Elaboración propia (2022).

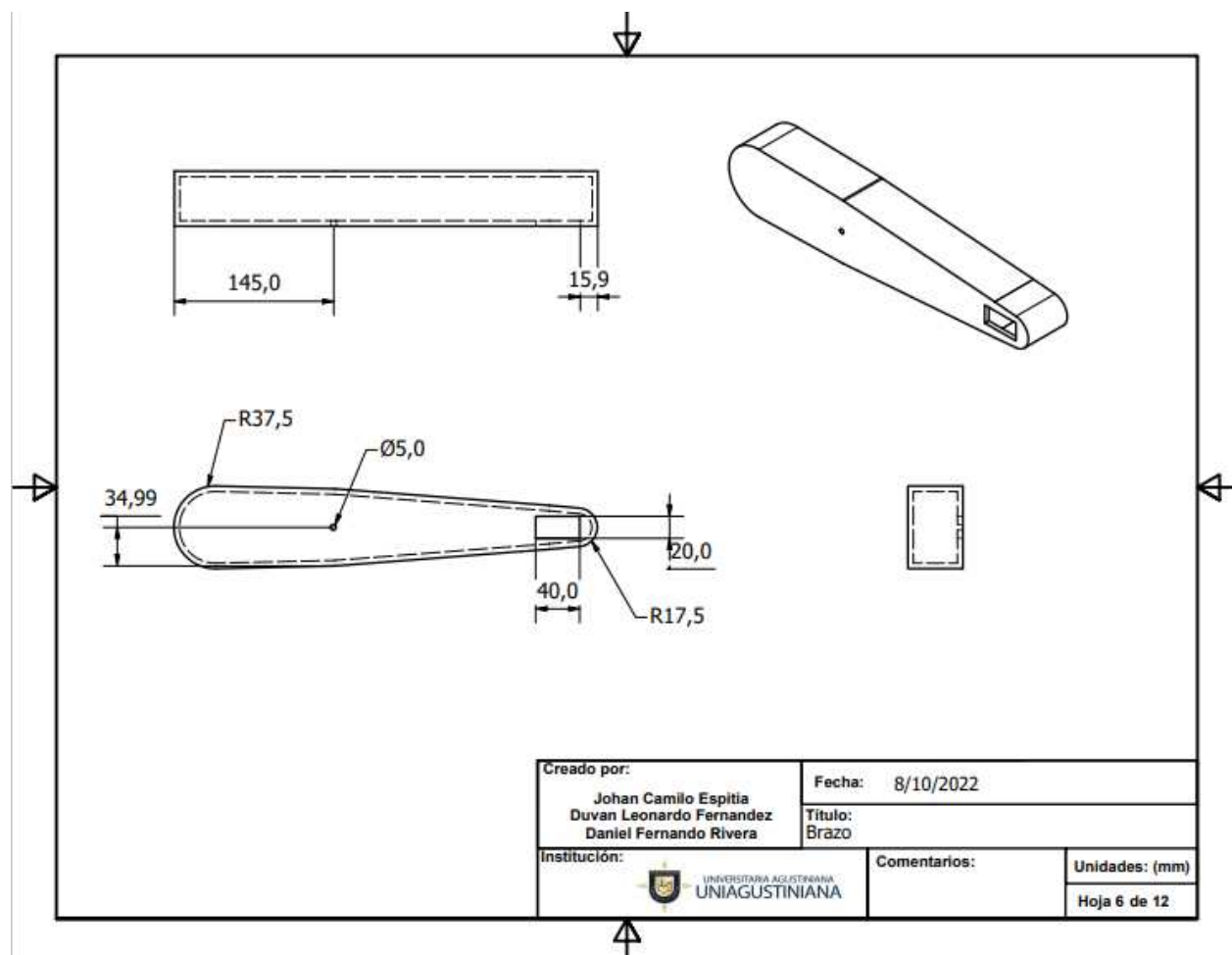


Figura 8. Brazo del brazo robótico parte 2. Elaboración propia (2022).

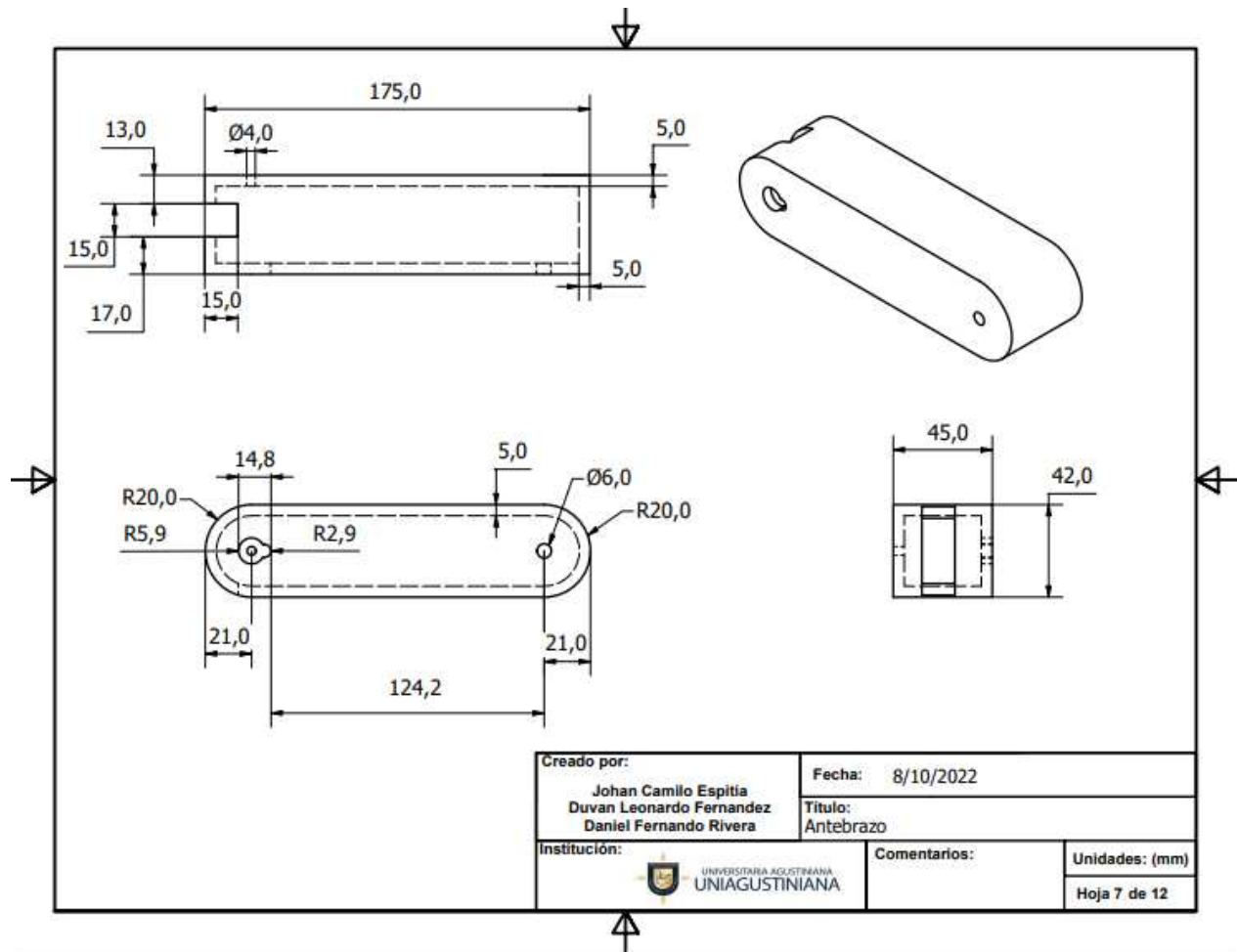


Figura 9. Antebrazo del brazo robótico. Elaboración propia (2022).

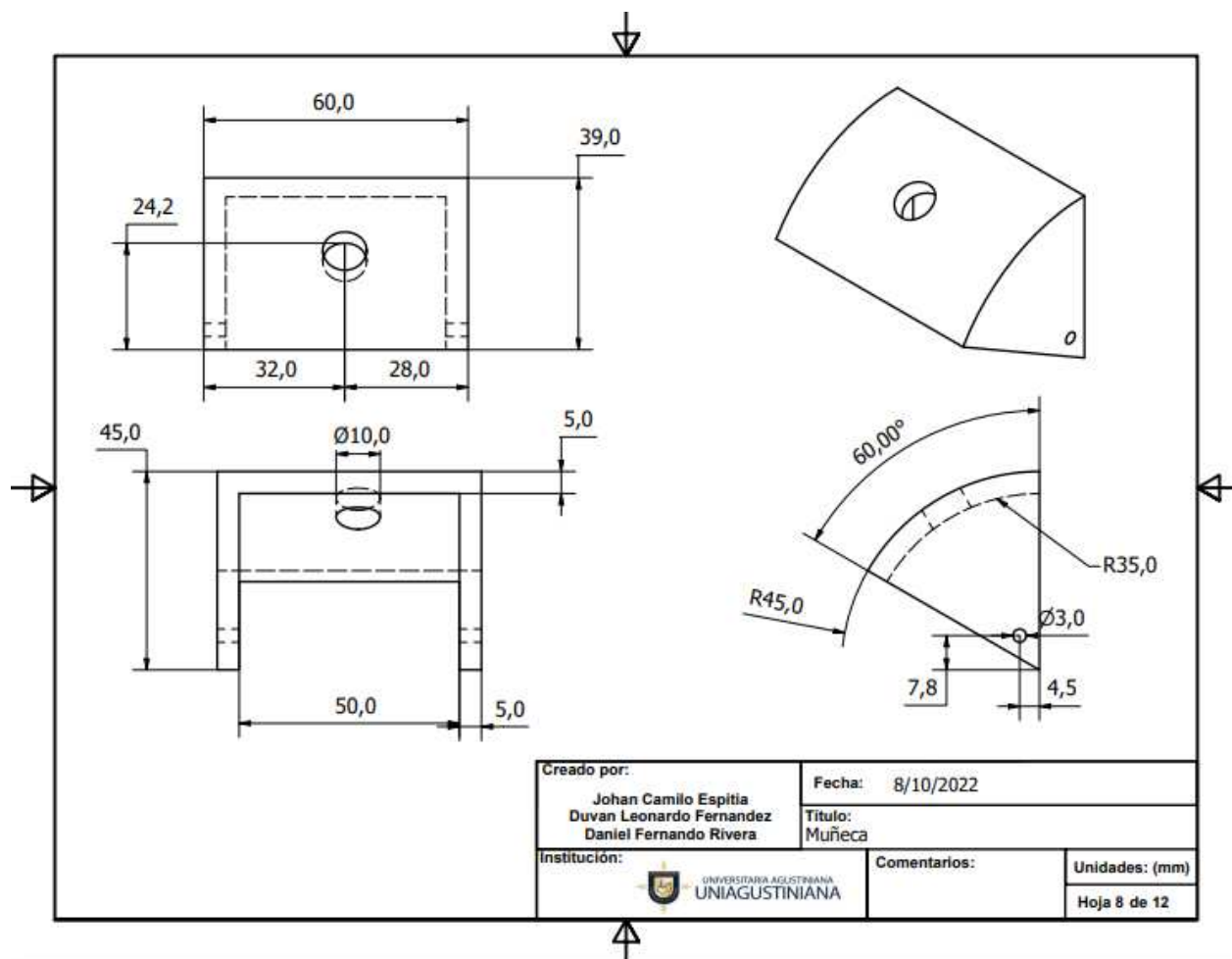


Figura 10. Muñeca de brazo robótico. Elaboración propia (2022).

Por otra parte el robot AGV se diseñó en base a los robots AGV de 4 ruedas, ya que el hecho de tener 4 ruedas facilitaría el giro de robot tanto en otras direcciones como en sí mismo, suprime la necesidad de instalar ruedas de apoyo para mantener la estabilidad del robot, y permite una mejor dispersión de peso sobre su superficie de carga. Además se optó por construir el cuerpo del robot con forma cuadrada a fin de transportar el mayor número de pedidos en un solo viaje, teniendo en cuenta que los pedidos utilizados en las pruebas se encuentran en forma de cubo. Ver figuras 11-19 para evidenciar las medidas estructurales del robot AGV.

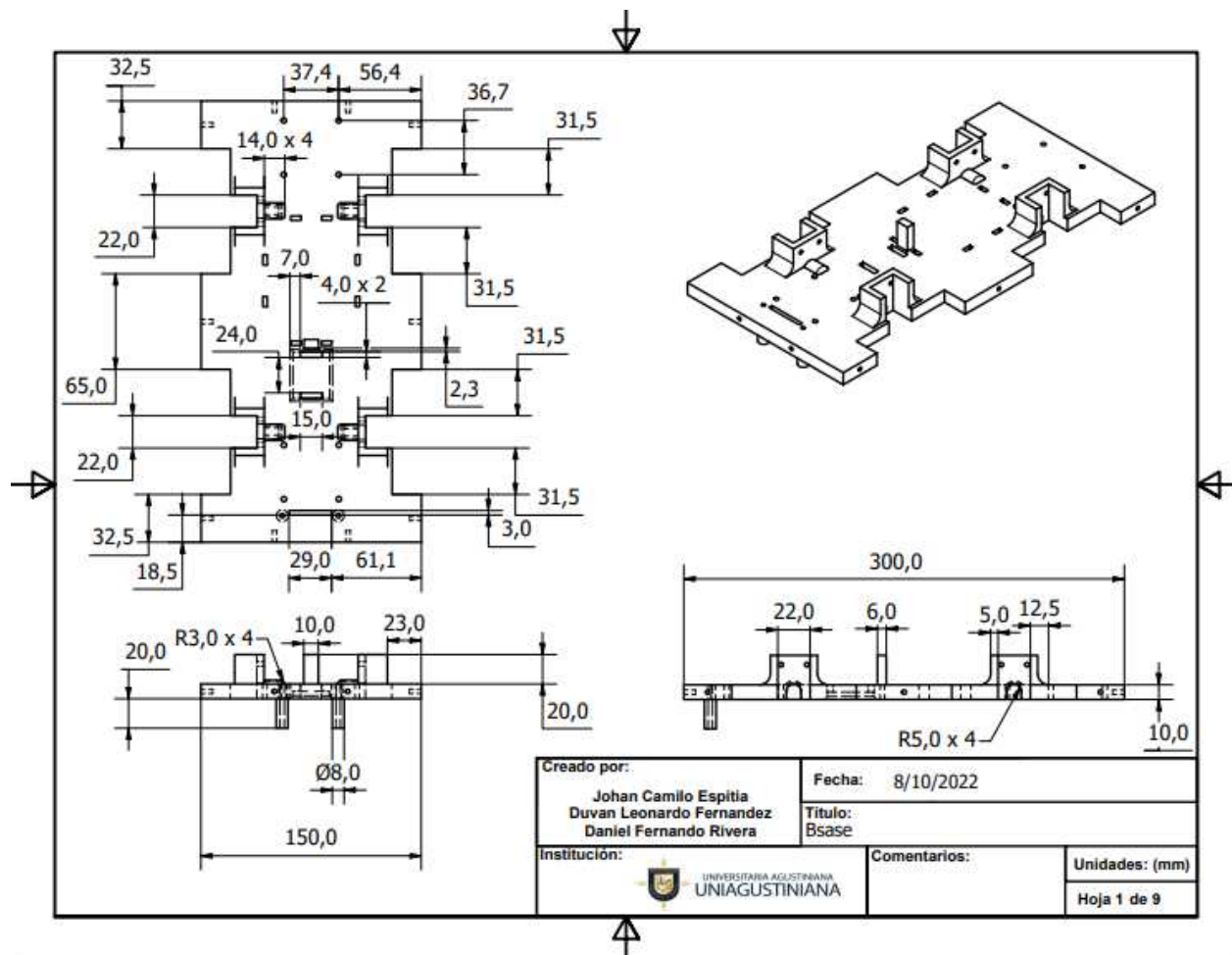


Figura 11. Base del robot AGV parte 1. Elaboración propia (2022).

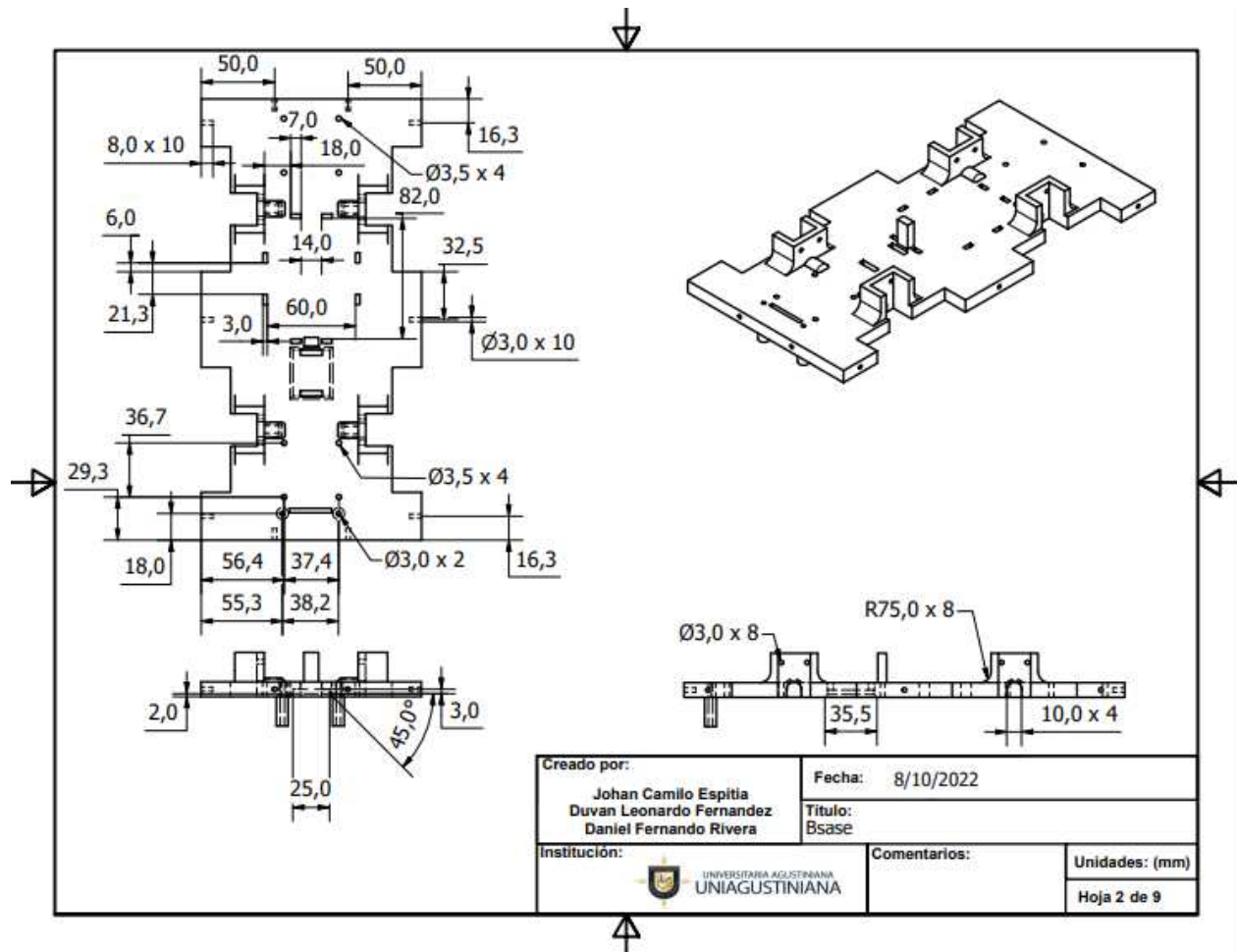


Figura 12. Base del robot AGV parte 2. Elaboración propia (2022).

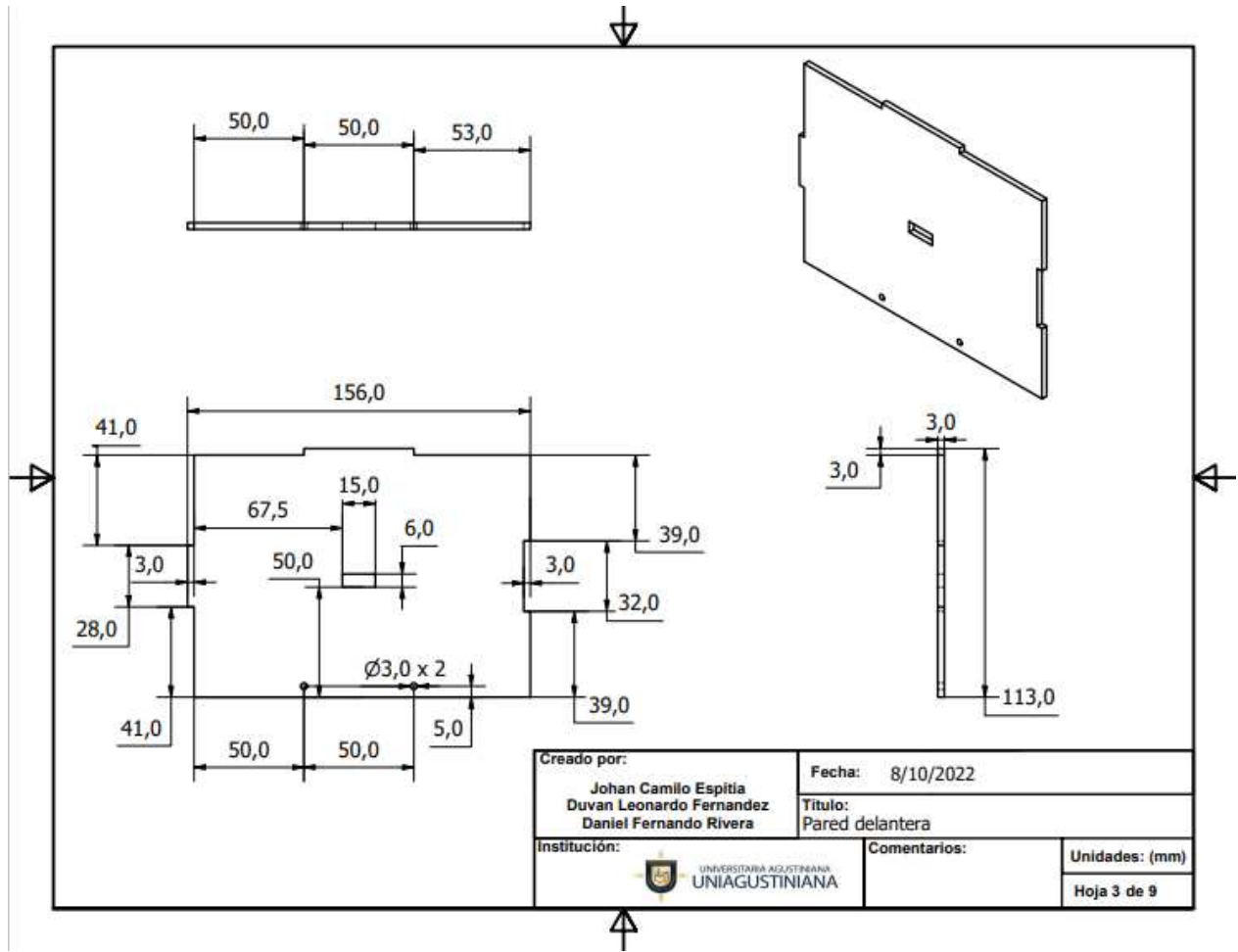


Figura 13. Pared frontal del robot AGV. Elaboración propia (2022).

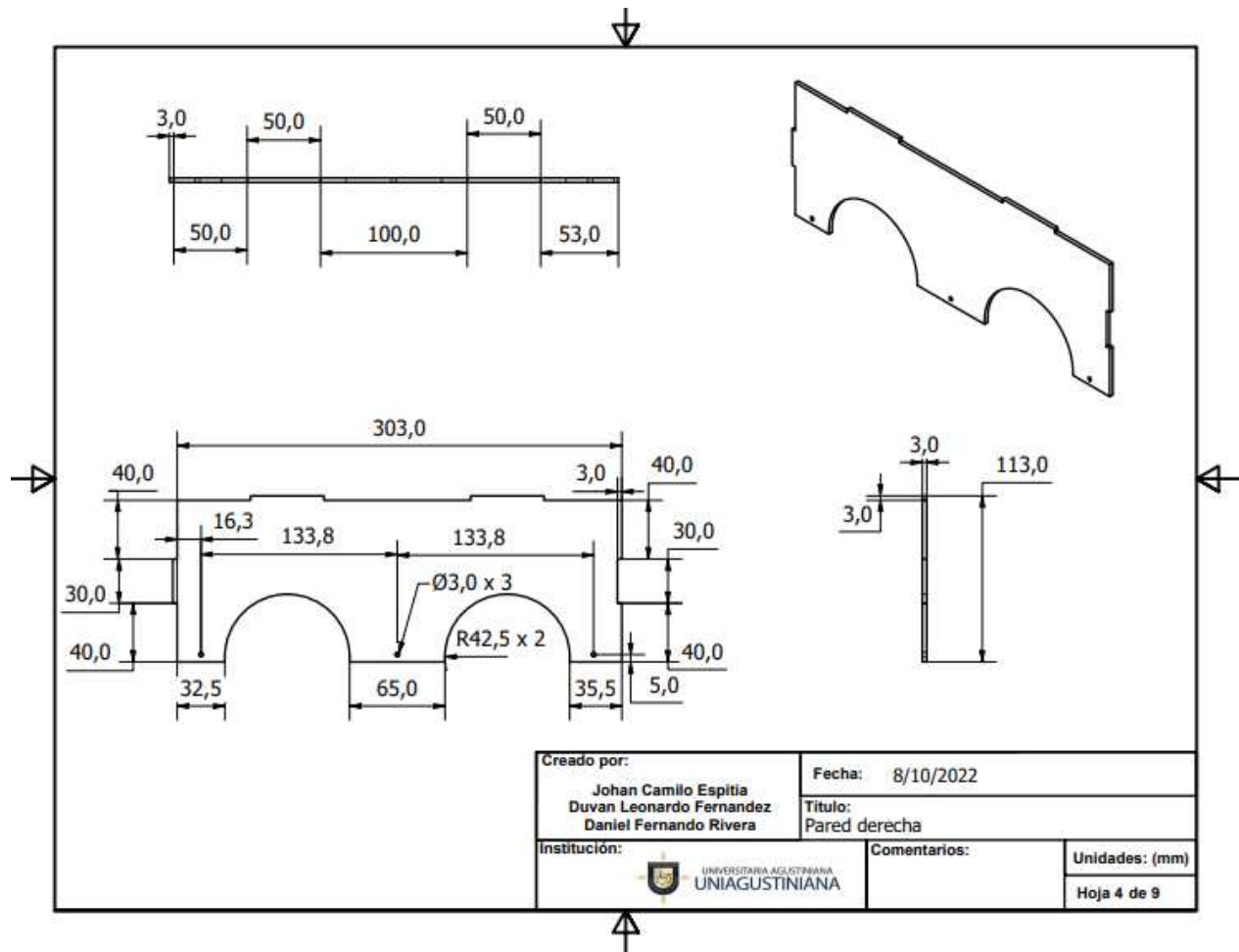


Figura 14. Pared lado derecho del robot AGV. Elaboración propia (2022).

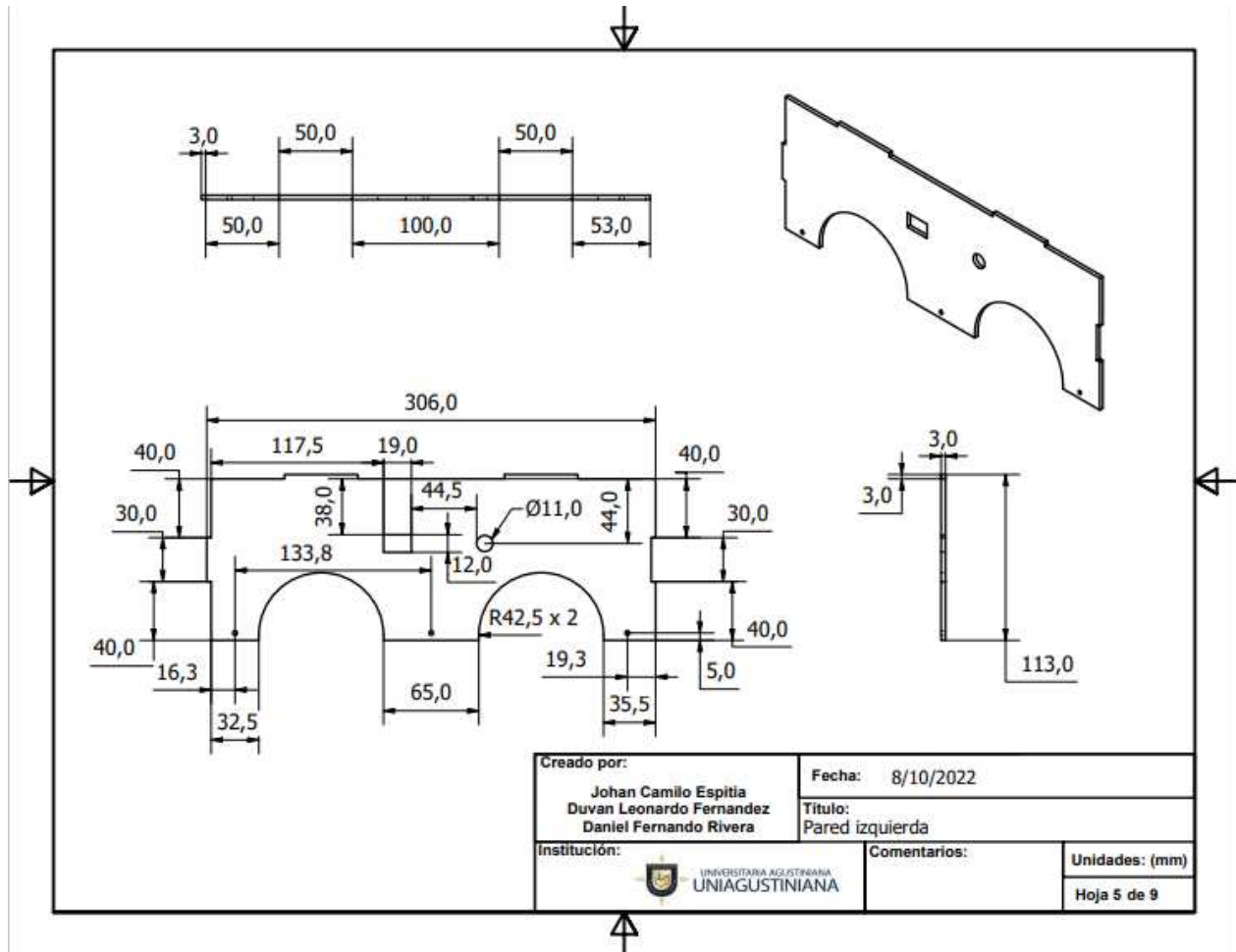


Figura 15. Pared lado izquierdo del robot AGV. Elaboración propia (2022).

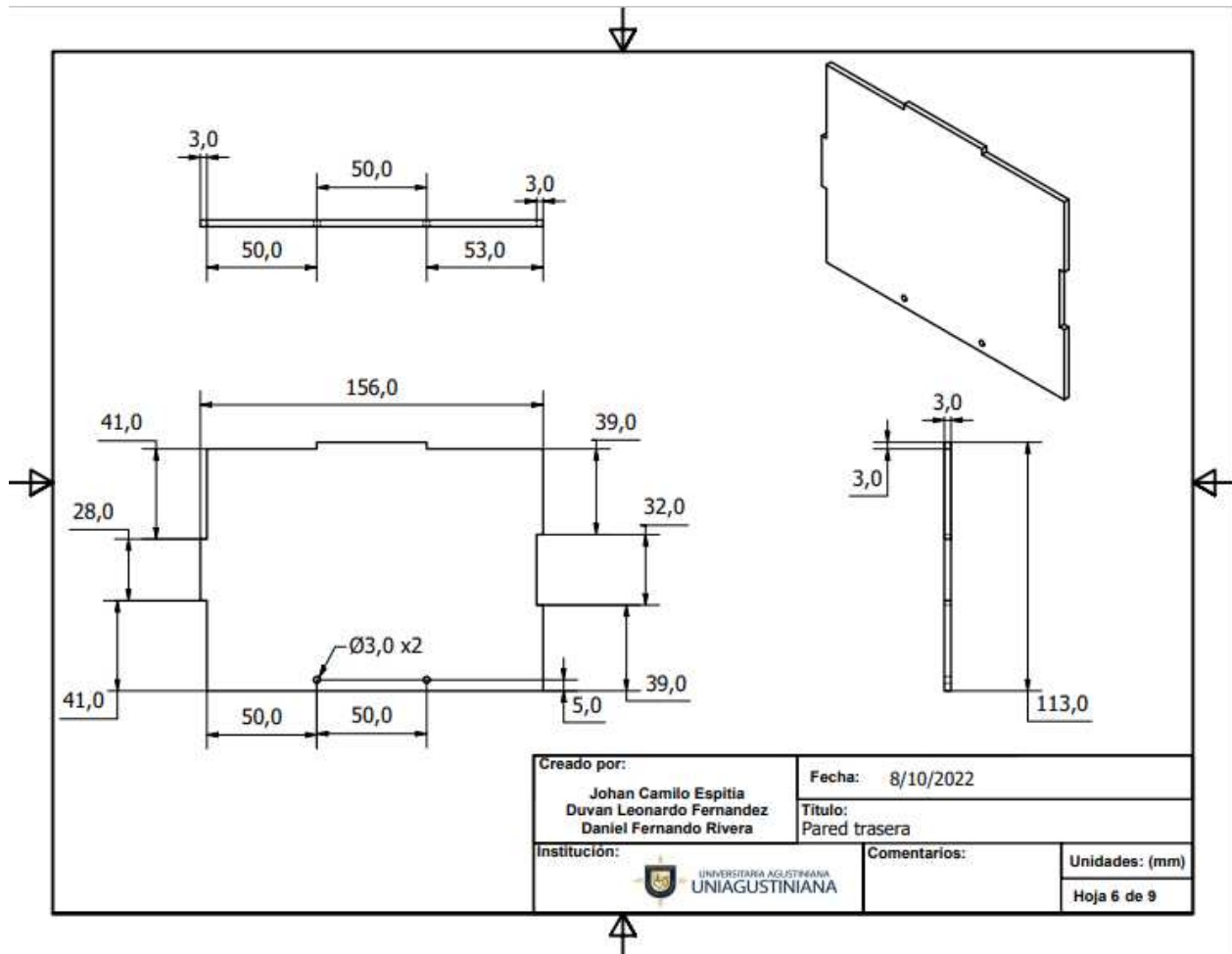


Figura 16. Pared trasera del robot AGV. Elaboración propia (2022).

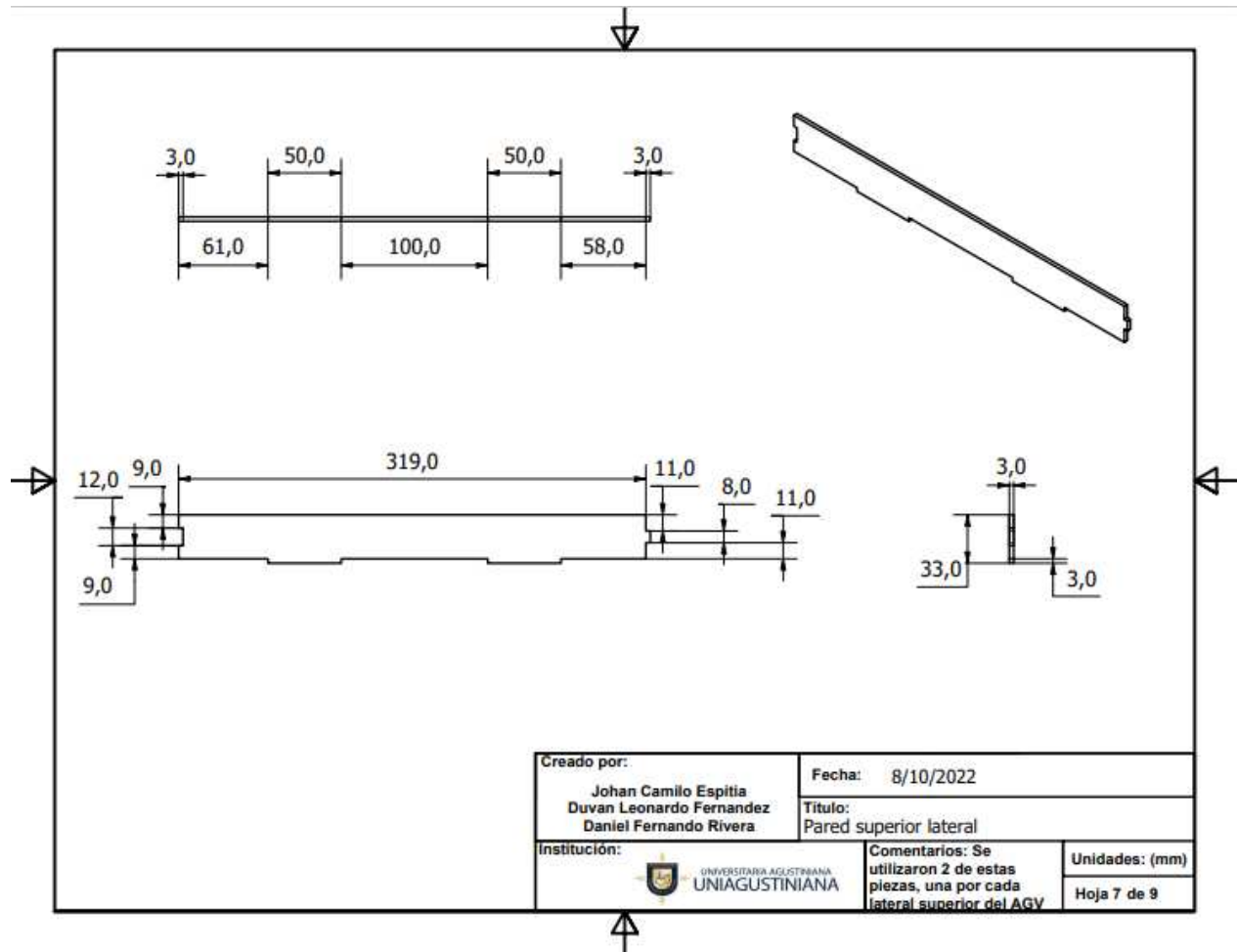


Figura 17. Paredes laterales bandeja de carga. Elaboración propia (2022).

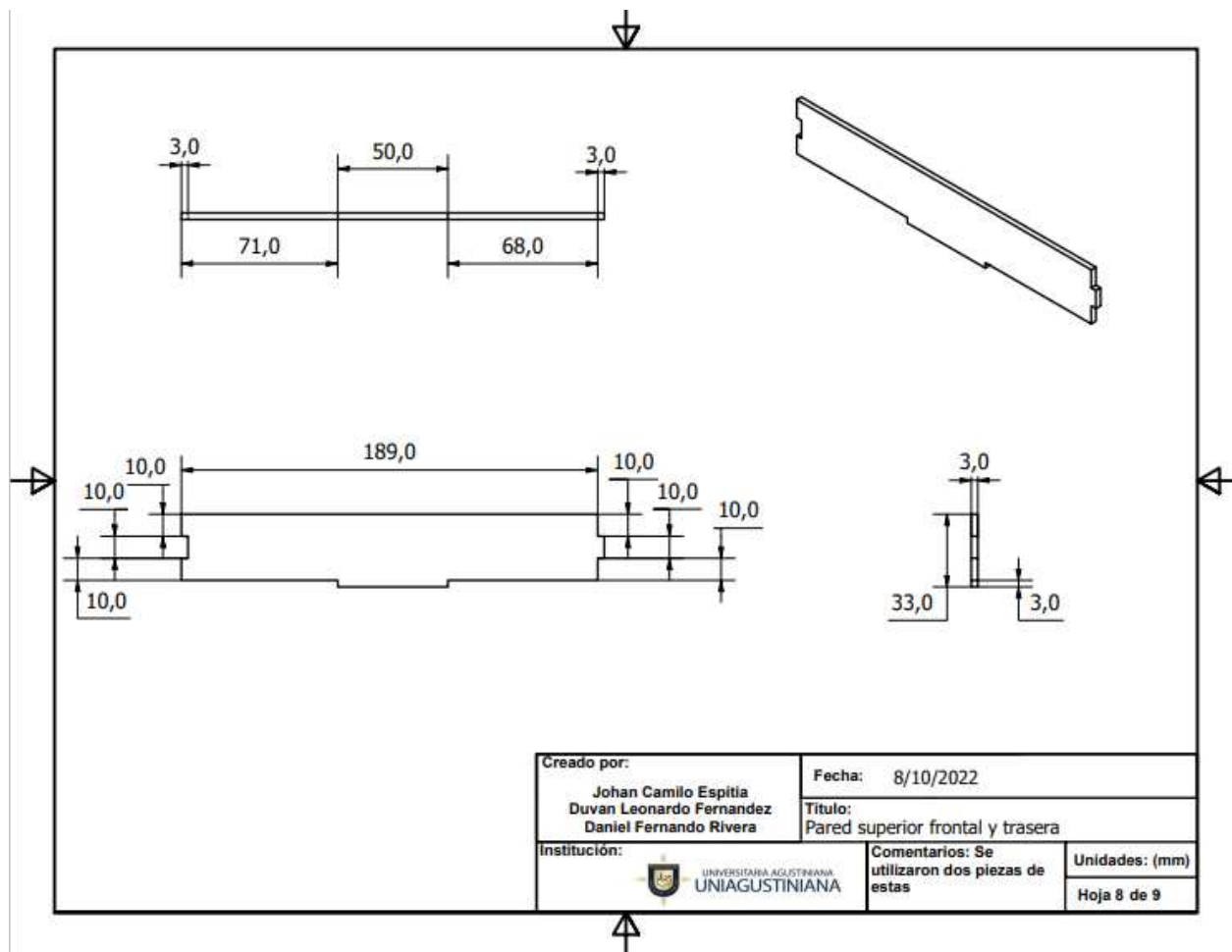


Figura 18. Pared delantera y trasera bandeja de carga. Elaboración propia (2022).

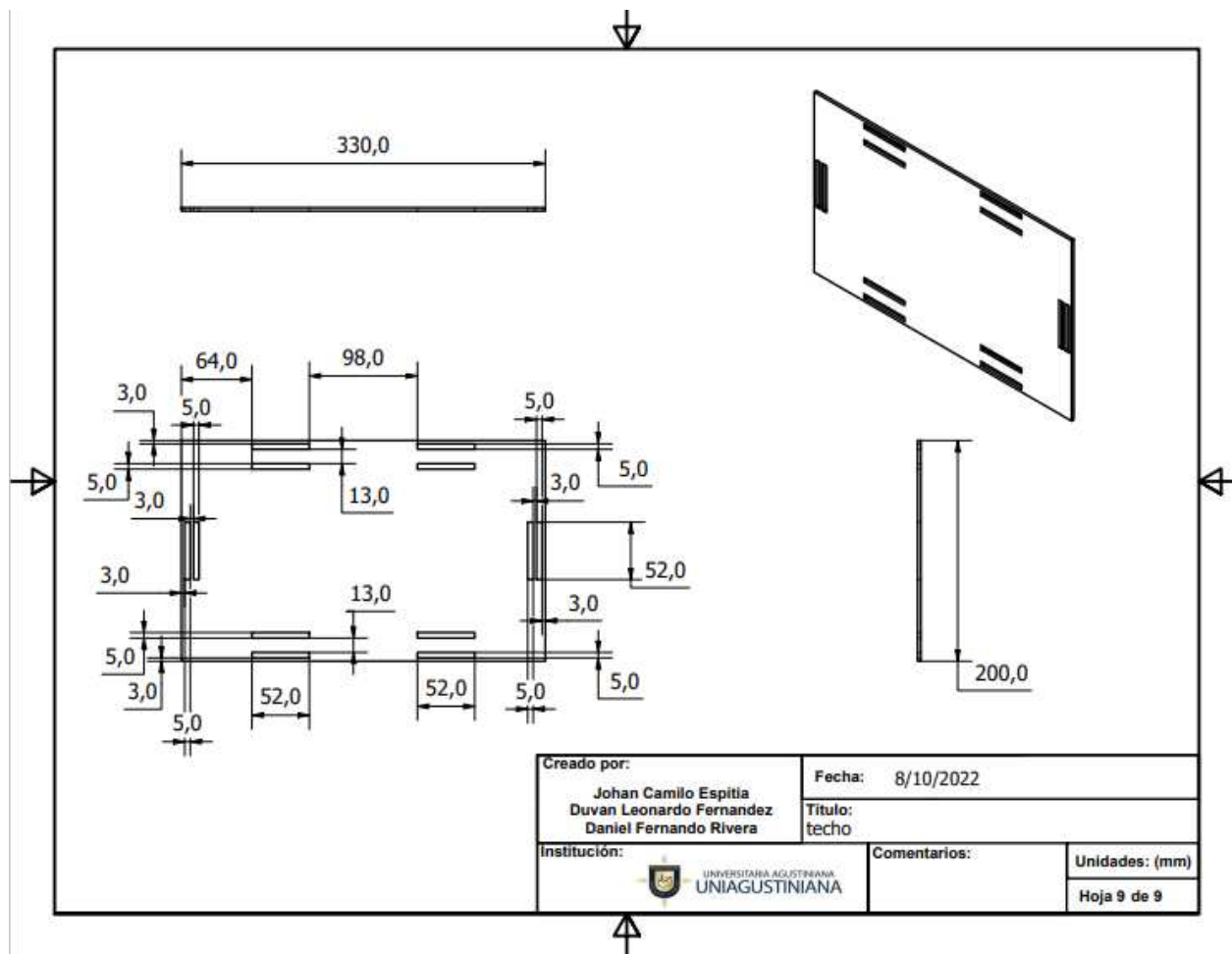


Figura 19. Superficie bandeja de carga. Elaboración propia (2022).

7.1.3 Caracterización y peso del brazo robótico y del robot AGV.

Para el proceso de pick and place que realizará el brazo robótico, es necesario que los motores que se vayan a emplear, no solo puedan mover la estructura del brazo, sino que sean capaces de transportar una carga externa, para esto se requiere que los motores cuenten con cierto nivel de torque, el cual se puede calcular mediante la fórmula general de torque.

$$M = F * d$$

$$M = \text{torque}$$

$$F = \text{fuerza o peso}$$

$$d = \text{distancia}$$

(4)

Como se ve en la formula anterior es necesario conocer la fuerza que tendrá que ejercer el motor y la distancia a la que operará, en el caso de la distancia, esta se puede conocer gracias a los planos mecánicos del brazo robótico, anteriormente mostrados en las figuras 3-10. Por otro lado para

conocer la fuerza, fue necesario pesar cada eslabón del brazo robótico, obteniendo así los valores más aproximados a la fuerza real que tendrá que levantar cada motor. Ver figuras 20-22.



Figura 20. Peso del hombro y brazo junto con un contrapeso de 220g. Elaboración propia (2022).



Figura 21. Peso del codo y antebrazo. Elaboración propia (2022).



Figura 22. Peso de la muñeca y gripper. Elaboración propia (2022).

Una vez pesados los eslabones, cabe resaltar que no fue necesario pesar la pieza de la base, ya que al no contar con movimientos horizontales, su peso no llega a afectar posteriormente a los cálculos para averiguar el peso máximo soportado por el brazo robótico, una vez seleccionados los

motores, por lo que se decidió calcular el torque necesario para la base, tomando el mismo valor que el del eslabón del hombro, al suponer que este eslabón sea el que más fuerza tenga que ejercer.

$$M_{hombro} = 0.027kg * 39cm = 1.053kg * cm \quad (5)$$

$$M_{codo} = 0.082kg * 19.5cm = 1.599kg * cm \quad (6)$$

$$M_{muñeca} = 0.045kg * 6.3cm = 0.2835kg * cm \quad (7)$$

Al conocer los torques mínimos que deberán tener los motores, para levantar solo la estructura del brazo robótico, cabe aclarar que es necesario elevar el valor del torque obtenido para lograr el objetivo de levantar un peso externo. Por lo que se deberá realizar una selección de motores con candidatos que superen los valores obtenidos y verificar por medio de la misma fórmula si el peso a levantar es suficiente para simular el proceso de logística adecuadamente.

Por otra parte se deben escoger sensores que puedan funcionar con un voltaje de 5V ya que este valor entregará el ESP32, que a su vez se utilizará por su capacidad de conectarse y comunicarse vía wifi, además al seleccionar sensores que funcionen con 5V se evitará hacer uso de reguladores de voltaje y en consecuencia no se aumentará el costo de construcción.

Además, conocer el peso de los dos robots ayudará a posteriormente calcular de manera más exacta el peso máximo que los motores de los mismos soportan sin sobre esforzarse, ya que los motores no solo tendrán que soportar una carga ajena a la estructura de los robots, sino que tendrá que ser capaz de levantar su propio peso.

Por lo que para averiguar este peso se utilizó una báscula en la que se colocaron, por una parte cada eslabón del brazo robótico con su respectivo motor por separado, y un peso de su estructura general, por otra parte para el peso del robot AGV no fue necesario separarlo en partes por lo que solo se pesó su estructura general. Ver figura 23 y 26.



Figura 23. Peso en gramos del brazo robótico. Elaboración propia (2022).

7.1.4 Selección de materiales y componentes.

El prototipo planteado en este documento está conformado por un brazo robótico y por un robot AGV, debido al alto costo que tendría la construcción de ambos robots a un tamaño similar a los robots existentes en el mercado, la construcción de ambos robots se realizó como una versión más pequeña a la que usualmente se usaría para la gestión de un almacén, no obstante, el tamaño no es el único factor que incrementa el costo de construcción de los robots, por lo tanto se decidió hacer uso de materiales reciclables como el cartón y PVC para la construcción de los eslabones que conforman el brazo robótico, por otra parte para el cuerpo del robot AGV se optó por usar acrílico transparente para las paredes del cuerpo y material de impresión 3D para su base, reduciendo en gran medida su costo final.

Cabe resaltar la falta de análisis en la selección de materiales de construcción, que al no presentar una relevancia a la hora de construir el prototipo, solo se consideró la facilidad de obtención de los materiales, pero no se vio la necesidad de un análisis completo, por lo que finalmente se optó por omitirlo.

7.1.4.1 selección motores. Para realizar la selección de los motores que le darán movimiento a ambos robots, primero se realizó una comparación entre las características de 4 posibles candidatos, cuyos datos se sacaron de páginas como: sigma electrónica, mercado libre, electronilab y Osaka

electronics, para determinar cuál sería la mejor opción para cada eslabón del brazo robótico y luego se seleccionó un motor para dar movimiento al robot AGV, por lo que en el caso del brazo robótico se consideraron como las características más relevantes lo siguiente: el torque de motor, el voltaje de funcionamiento, el tipo de piñonera, el amperaje máximo, las dimensiones del motor y el costo, mientras que para el robot AGV se consideraron características como: torque, relación de reducción, costo y velocidad. Ver las tablas 4 y 11. Es importante mencionar que para tomar en cuenta a los candidatos en las selecciones se tomó en cuenta su disponibilidad en el mercado, dejando a un lado otras características para analizar en la selección final.

Tabla 4.

Características candidatos servomotores.

servomotores	mg996r	mg995	sg90	mg90d
Torque	11Kg-cm	10Kg-cm	1,5Kg-cm	2,4Kg-cm
V funcionamiento	6V	6V	6V	6V
Tipo piñonera	Metálica	Metálica	Plástica	Metálica
A máxima	2,5A	2,5A	700mA	750mA
Dimensiones (mm)	54x20x45	24x20x45	32,6x12,5x27,3	28,5x12,2x22,8
Costo	\$ 27.700	\$ 32.725	\$ 12.500	\$ 15.800

Nota. En la tabla se evidencian las características a tomar en cuenta para la selección de los servomotores. Elaboración propia (2022).

La primera selección se hizo para determinar el motor de la base del brazo robótico, en este caso, se estableció una mayor importancia sobre el torque, ya que era necesario que el motor tuviera la capacidad de girar toda la estructura y por consiguiente al necesitar un motor con un torque acorde a las necesidades del proceso, su tamaño también tomó relevancia, ya que entre más torque tenga un motor, mayor es su tamaño, además un motor de buen tamaño le daría estabilidad a la construcción, por lo que estas características aunque se les puntuó en el mismo rango de uno (menor puntaje) hasta diez (mayor puntaje) que a las demás características, el torque y el tamaño tuvieron puntuaciones más críticas. Ver la tabla 5.

Tabla 5.

Selección de servomotor para la base.

Selección servomotor para base								
	Dato	Puntaje	Dato	Puntaje	Dato	Puntaje	Dato	Puntaje
servomotores	mg996r	1-10	mg995	1-10	sg90	1-10	mg90d	1-10
Torque	11Kg-cm	10	10Kg-cm	6	1,5Kg-cm	2	2,4Kg-cm	3
V funcionamiento	6V	5	6V	5	6V	5	6V	5
Tipo piñonera	Metalica	5	Metalica	5	Plastica	2	Metalica	5
A maxima	2,5A	8	2,5A	8	700mA	4	750mA	5
Dimensiones (mm)	54x20x45	7	54x20x45	7	32,6x12,5x27,3	4	28,5x12,2x22,8	2
Costo	\$ 27.700	5	\$ 32.725	8	\$ 12.500	8	\$ 15.800	7
Totales		40		39		25		27

Nota. En la tabla se evidencia la puntuación para cada característica, se debe tener en cuenta que el servomotor con mayor puntaje será el elegido. Elaboración propia (2022).

La siguiente selección se hizo para determinar el motor del hombro del brazo robótico, de igual manera que para la base, el torque del motor fue un factor determinante para su posterior elección, ya que como se mencionó anteriormente, fue necesario un motor capaz de girar la estructura del brazo robótico, exceptuando la base en este caso, por lo que manejando el mismo rango de puntuación que se utilizó en la base y el cual se usó para cada selección, el torque y el tamaño del motor tuvieron una relevancia más alta que el resto de características. Ver tabla 6.

Tabla 6.

Selección servomotor para el hombro.

Selección servomotor para hombro								
	Dato	Puntaje	Dato	Puntaje	Dato	Puntaje	Dato	Puntaje
servomotores	mg996r	1-10	mg995	1-10	sg90	1-10	mg90d	1-10
Torque	11Kg-cm	10	10Kg-cm	6	1,5Kg-cm	2	2,4Kg-cm	3
V funcionamiento	6V	5	6V	5	6V	5	6V	5
Tipo piñonera	Metalica	5	Metalica	5	Plastica	2	Metalica	5
A maxima	2,5A	8	2,5A	8	700mA	4	750mA	5
Dimensiones (mm)	54x20x45	7	54x20x45	7	32,6x12,5x27,3	4	28,5x12,2x22,8	2
Costo	\$ 27.700	5	\$ 21.000	7	\$ 12.500	10	\$ 15.800	9
Totales		40		38		27		29

Nota. En la tabla se evidencia la puntuación para cada característica, se debe tener en cuenta que el servomotor con mayor puntaje será el elegido. Elaboración propia (2022).

Por otro lado, la selección para el motor del codo se hizo tomando la característica de costo como la más importante, ya que para este caso se pudo permitir un torque de motor algo menor ya que el motor elegido solo necesitaba ser capaz de girar la estructura desde el codo, donde el peso para este punto es significativamente menor, no obstante, si bien se tomó el factor del costo como el más relevante, esto no significó no tomar en cuenta las demás características. Ver tabla 7.

Tabla 7.

Selección servomotor para codo.

Selección servomotor para codo								
	Dato	Puntaje	Dato	Puntaje	Dato	Puntaje	Dato	Puntaje
servomotores	mg996r	1-10	mg995	1-10	sg90	1-10	mg90d	1-10
Torque	11Kg-cm	10	10Kg-cm	9	1,5Kg-cm	2	2,4Kg-cm	3
V funcionamiento	6V	5	6V	5	6V	5	6V	5
Tipo piñonera	Metalica	5	Metalica	5	Plastica	2	Metalica	5
A maxima	2,5A	8	2,5A	8	700mA	4	750mA	5
Dimensiones (mm)	54x20x45	7	54x20x45	7	32,6x12,5x27,3	4	28,5x12,2x22,8	2
Costo	\$ 27.700	3	\$ 21.000	6	\$ 12.500	10	\$ 15.800	9
Totales		38		40		27		29

Nota. En la tabla se evidencia la puntuación para cada característica, se debe tener en cuenta que el servomotor con mayor puntaje será el elegido. Elaboración propia (2022).

Para la selección del motor para la muñeca del brazo robótico, el tamaño del motor tomó una gran importancia mientras que el torque no fue tan relevante, esto se debe a que para la muñeca era necesario un motor de dimensiones pequeñas ya que el eslabón de la muñeca fue el más pequeño de todos, además al agregar un motor pequeño se pudo disminuir el peso total de la estructura completa, evitando sobre esforzar el motor elegido para la base. Ver tabla 8.

Tabla 8.

Selección servomotor para muñeca.

Selección servomotor para muñeca								
	Dato	Puntaje	Dato	Puntaje	Dato	Puntaje	Dato	Puntaje
servomotores	mg996r	1-10	mg995	1-10	sg90	1-10	mg90d	1-10
Torque	11Kg-cm	10	10Kg-cm	9	1,5Kg-cm	4	2,4Kg-cm	6
V funcionamiento	6V	5	6V	5	6V	5	6V	5
Tipo piñonera	Metalica	7	Metalica	7	Plastica	2	Metalica	7
A maxima	2,5A	8	2,5A	8	700mA	4	750mA	4
Dimensiones (mm)	54x20x45	2	54x20x45	2	32,6x12,5x27,3	4	28,5x12,2x22,8	8
Costo	\$ 27.700	2	\$ 21.000	4	\$ 12.500	9	\$ 15.800	7
Totales		34		35		28		37

Nota. En la tabla se evidencia la puntuación para cada característica, se debe tener en cuenta que el servomotor con mayor puntaje será el elegido. Elaboración propia (2022).

Se seleccionó además un motor paso a paso para darle la capacidad al brazo robótico de moverse a lo largo de uno de sus ejes en línea recta, en esta selección, el torque, el voltaje de funcionamiento y el costo fueron las características más relevantes, la primera para garantizar el movimiento del brazo robótico sin sobre esfuerzo, la segunda ya que voltajes distintos de cinco o doce voltios requieren la implementación de un regulador de voltaje extra, y el costo para mantener al mínimo el gasto de construcción del prototipo. Ver tablas 9 y 10.

Tabla 9.

Características candidatos motor paso a paso.

Motores paso a paso	Nema-17	28byj-48	Nema-23
torque	4 kg-cm	0,299 Kg-cm	12,85 kg-cm
V funcionamiento	12V	5V	2,5V
Longitud eje	18mm	5mm	21mm
Dimensiones (mm)	60x42x42	28x24x35	57x57x56
Costo	\$ 50.000	\$ 14.000	\$ 150.000

Nota. En la tabla se evidencian las características a tomar en cuenta para la selección del motor paso a paso. Elaboración propia (2022).

Tabla 10.

Selección motor paso a paso.

Selección motor paso a paso						
	Dato	Puntaje	Dato	Puntaje	Dato	Puntaje
Motores paso a paso	Nema-17	1-10	28byj-48	1-10	Nema-23	1-10
torque	4 kg-cm	6	0,299 Kg-cm	1	12,85 kg-cm	10
V funcionamiento	12V	10	5V	10	2,5V	1
Longitud eje	18mm	6	5mm	2	21mm	7
Dimensiones (mm)	60x42x42	5	28x24x35	3	57x57x56	5
Costo	\$ 50.000	5	\$ 14.000	7	\$ 150.000	1
totales		32		23		24

Nota. En la tabla se evidencia la puntuación para cada característica, se debe tener en cuenta que el motor paso a paso con mayor puntaje será el elegido. Elaboración propia (2022).

Por último se seleccionaron 4 motorreductores para el robot AGV, ya que este contará con 4 ruedas que le permitirán transportar más peso que si solo tuviera dos, para esta selección se tuvo en cuenta principalmente el torque de los moto reductores, el costo y el voltaje de funcionamiento, esto ya que un motor con mayor torque permite el transporte de cargas más elevadas, no obstante un bajo costo por motor tomó inclusive más relevancia ya que era necesario adquirir 4 de estos, además como se mencionó anteriormente un motor que funcionara ya sea a 5 o 12 voltios no tendría necesidad de adicionar reguladores de voltaje para evitar estropearlos. Ver tablas 11 y 12.

Tabla 11.

Características candidatos motorreductor.

motorreductor	Amarillo de plastico	metalico pololu	Pololu-2215
Torque	1kg-cm	1,1kg-cm	1,6kg-cm
Velocidad	200 rpm	625rpm	400rpm
Relación de reducción	48:1	50:1	75:1
V funcionamiento	5-12V	6V	6V
Costo	\$ 10.000	\$ 58.000	\$ 72.000

Nota. En la tabla se evidencian las características a tomar en cuenta para la selección de los motorreductores para el robot AGV. Elaboración propia (2022).

Tabla 12.

Selección motorreductores.

Selección motorreductor						
	Dato	Puntaje	Dato	Puntaje	Dato	Puntaje
motorreductor	Amarillo de plastico	1-10	metalico pololu	1-10	Pololu-2215	1-10
Torque	1kg-cm	7	1,1kg-cm	8	1,6kg-cm	10
Velocidad	200 rpm	4	625rpm	8	400rpm	6
Relación de reducción	48:1	4	50:1	5	75:1	7
V funcionamiento	5-12V	10	6V	4	6V	4
Costo	\$ 10.000	10	\$ 58.000	6	\$ 72.000	3
Totales		35		31		30

Nota. En la tabla se evidencia la puntuación para cada característica, se debe tener en cuenta que el motorreductor con mayor puntaje será el elegido y que se adquirirán 4 unidades del mismo. Elaboración propia (2022).

7.1.4.2. Selección de sensores. Durante el proceso de logística que realizarán el brazo robótico y el robot AGV se decidió controlar las variables de distancia y color mediante sensores, en el caso del brazo robótico se añadió un sensor infrarrojo a un costado de la ventosa, a fin de darle la capacidad de reconocer si el pedido seleccionado se encuentra en el lugar indicado, mientras que en el caso del robot AGV, este cuenta con un sensor de línea, un sensor de color y un sensor infrarrojo, donde el sensor de línea facilitará el recorrido del robot AGV hasta un punto del almacén (donde el brazo robótico le entregara el pedido) y devuelta al punto de entrega (donde el robot AGV entregará el pedido), por otro lado el sensor de color le ayudará a identificar la estación en la que se encuentra mediante parches de color en el suelo, y por último el sensor infrarrojo ayudará a evitar que el robot AGV llegue a chocar con una pared, los datos se obtuvieron de páginas como: yo robotics, electronilab, mactronica, mercado libre, dynamo electronics y electronic spot.

Empezando por la selección de los sensores infrarrojos, se optó por usar el mismo sensor en ambos robots para minimizar posibles viajes a distintos puntos de venta, se tomaron en cuenta los factores de costo, voltaje de funcionamiento y distancia de detección para realizar la selección, en este caso el costo fue el factor más relevante. Aunque la distancia de detección también tuvo relevancia, no se vio la necesidad de una gran distancia de detección en el caso del brazo robótico ya que la distancia entre este y el pedido no sería tan pronunciada. Ver tablas 13 y 14.

Tabla 13.

Características candidatos sensor infrarrojo.

Sensor infrarrojo	FC-51	Ky-032
V funcionamiento	5V	5V
costo	\$ 4.500	\$ 9.000
Distancia de detección	2-40cm	2-8cm

Nota. En la tabla se evidencian las características a tomar en cuenta para la selección del sensor infrarrojo para el robot AGV y el brazo robótico. Elaboración propia (2022).

Tabla 14.

Selección sensor infrarrojo.

Selección sensor infrarrojo				
	Dato	Puntaje	Dato	Puntaje
Sensor infrarrojo	FC-51	1-10	Ky-032	1-10
V funcionamiento	5V	5	5V	5
costo	\$ 4.500	10	\$ 16.000	3
Distancia de detección	2-8cm	4	2-40cm	10
Totales		19		18

Nota. En la tabla se evidencia la puntuación para cada característica, se debe tener en cuenta que el sensor infrarrojo con mayor puntaje será el elegido y que se adquirirán 2 unidades del mismo. Elaboración propia (2022).

Para la selección del sensor de color solo se tomaron en cuenta el voltaje de funcionamiento y el costo, donde el costo tomó más relevancia ya que los voltajes de los candidatos permitían la conexión sin necesidad de adicionar un regulador de voltaje. Ver tablas 15 y 16.

Tabla 15.

Características candidatos sensor de color.

Sensor de color	TCS3200	TCS34725
V funcionamiento	5,5V	3,8V
costo	19000	24000

Nota. En la tabla se evidencian las características a tomar en cuenta para la selección del sensor de color para el robot AGV. Elaboración propia (2022).

Tabla 16.

Selección sensor de color.

Selección sensor de color				
	Dato	Puntaje	Dato	Puntaje
Sensor de color	TCS3200	1-10	TCS34725	1-10
V funcionamiento	5,5V	7	3,8V	4
costo	\$ 19.000	8	\$ 24.000	6
Totales		15		10

Nota. En la tabla se evidencia la puntuación para cada característica, se debe tener en cuenta que el sensor de color con mayor puntaje será el elegido. Elaboración propia (2022).

Por último se realizó la selección del sensor de línea, para el cual se necesitaron seis canales con el fin de programar correctamente el comportamiento del robot con respecto a los canales activos, por esta razón el número de canales con los que contaba el sensor fue el factor más relevante seguido del costo, ya que en caso de que el sensor contará con menos de seis canales, se generaría la necesidad de comprar sensores adicionales. Ver tablas 17 y 18.

Tabla 17.

Características sensor de línea.

Selección sensor línea				
	Dato	Puntaje	Dato	Puntaje
Sensor línea	Sensor línea digital	1-10	Sensor línea analógico	1-10
V funcionamiento	5V	5	5V	5
costo	\$ 7.700	9	\$ 20.000	5
Canales	1	1	8	10
Distancia detección óptima	3cm	7	3mm	3
Totales		22		23

Nota. En la tabla se evidencian las características a tomar en cuenta para la selección del sensor de línea para el robot AGV. Elaboración propia (2022).

Tabla 18.

Selección sensor de línea.

Selección sensor línea				
	Dato	Puntaje	Dato	Puntaje
Sensor línea	Sensor línea digital	1-10	Sensor línea analógico	1-10
V funcionamiento	5V	5	5V	5
costo	\$ 7.700	9	\$ 20.000	5
Canales	1	1	8	10
Distancia detección óptima	3cm	7	3mm	3
Totales		22		23

Nota. En la tabla se evidencia la puntuación para cada característica, se debe tener en cuenta que el sensor de color con mayor puntaje será el elegido. Elaboración propia (2022).

7.1.5 Pesos máximos.

Los pesos máximos determinan las cargas que soporta cada robot sin sobre esfuerzo, para averiguar estos pesos, en el caso del brazo robótico fue necesario calcular la fuerza de cada uno de los motores seleccionados anteriormente usando la fórmula para el torque.

$$M = F * d$$

$$M = \text{momento o torque.}$$

$$F = \text{fuerza aplicada.} \quad (8)$$

$$d = \text{distancia al eje de giro.}$$

Se despeja la ecuación, obteniendo.

$$F = \frac{M}{d}$$

Para empezar se calculó la fuerza del motor ubicado en el hombro del brazo, usando el valor del torque del motor y la distancia desde el eje de giro hasta el extremo de la ventosa.

$$F = \frac{11Kg * cm}{39 cm} = 0.282kg \rightarrow 282g \quad (9)$$

El siguiente calculo fue el de la fuerza del motor del codo, de igual manera se utilizó el valor del torque del motor seleccionado y la distancia desde el eje de giro del codo hasta el extremo de la ventosa.

$$F = \frac{10Kg * cm}{19.5 cm} = 0.012Kg \rightarrow 512g \quad (10)$$

Por último, para el cálculo de la fuerza del motor de la muñeca se utilizó el torque del motor seleccionado y la distancia desde el eje de giro de la muñeca hasta el extremo de la ventosa.

$$F = \frac{2.4Kg * cm}{6.3cm} = 0.380Kg \rightarrow 380g \quad (11)$$

Una vez obtenidos los resultados se pudo determinar que el brazo robótico tiene la capacidad de levantar como máximo 282g, es importante aclarar que este valor corresponde a la carga soportada con el brazo totalmente extendido, por lo que en este estado el brazo no puede soportar una carga más alta, contrariamente del caso en el que el brazo no se encuentra totalmente extendido, donde al no tener que realizar tanto esfuerzo que estando extendido, el brazo sería capaz de levantar más peso. La posición del brazo es a su vez la razón por la que no se realizó un cálculo de fuerza con el motor de la base, pues al encontrarse totalmente extendido, el motor de la base no se ve obligado a realizar demasiada fuerza.

Por otra parte para identificar el peso máximo soportado por el robot AGV se optó por definir este valor de manera experimental, para esto se puso al robot en marcha mientras se le añadía una carga cada vez más alta, una vez se visualizó un sobre esfuerzo por parte de los motorreductores seleccionados, la carga puesta hasta ese punto se tomó como el peso máximo que soportaría el AGV siempre y cuando se desprece su peso, ya que de contar el peso de la estructura el peso máximo aumenta a la suma de la carga total que puede llevar el AGV y el peso de su estructura . Ver figuras 24-26.



Figura 24. Carga máxima sin sobre esfuerzo. Elaboración propia (2022).



Figura 25. Carga máxima experimental en gramos. Elaboración propia (2022).



Figura 26. Peso de la estructura del robot AGV. Elaboración propia (2022).

7.1.6 Cadena cinemática.

Para calcular la cadena cinemática del brazo robótico necesaria para determinar las posiciones que pueden tomar cada uno de los eslabones del robot, se realizó una representación gráfica que servirá de guía para identificar los ejes de giro que tienen cada uno de los eslabones. Ver figura 27.

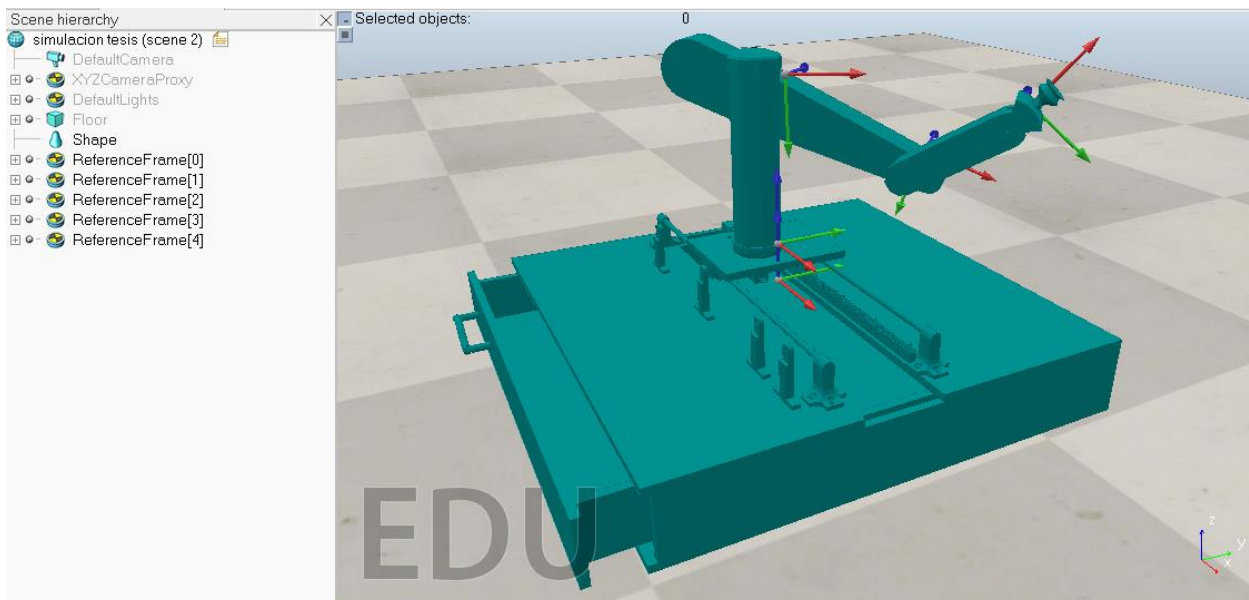


Figura 27. Definición cadena cinemática. Elaboración propia (2022).

Al tener la representación gráfica de la cadena cinemática finalizada, se puede determinar con mayor facilidad los parámetros necesarios para realizar la tabla de Denavit-Hartenberg, al analizar brevemente la representación de la figura 6. Ver tablas 19 y 20.

Tabla 19.

Parámetros generales de Denavit-Hartenberg.

Eslabones	θ	d_z	a_x	α
1	0	0	L1	0
2	θ_1	d1	0	0
3	θ_2	L2	0	90°
4	θ_3	L3	0	0
5	θ_4	L4	0	0

Nota. Los datos de la tabla representan los parámetros para un robot de 5 grados de libertad cualquiera, elaboración propia (2022).

Tabla 20.

Parámetros específicos de denavit-hartenberg.

eslabones	Θ	dz	ax	α
1	0	0	0,47m	0
2	Θ_1	0,24m	0	0
3	Θ_2	0,385m	0	90°
4	Θ_3	0,175m	0	0
5	Θ_4	0,045m	0	0

Nota: los datos de la tabla representan los parámetros para el robot de 5 grados de libertad propuesto en el documento, elaboración propia (2022).

(12)

Una vez se obtienen los parámetros de denavit-hartenberg, se puede hallar la transformación homogénea A_i que representará el movimiento de cada eslabón del brazo robótico, para esto se deben multiplicar 4 transformaciones básicas.

$$A_i = R_{z,\Theta_i} * T_{z,d_i} * T_{x,a_i} * R_{x,\alpha_i}$$

$$A_i = \begin{vmatrix} \cos \Theta_i & -\sin \Theta_i & 0 & 0 \\ \sin \Theta_i & \cos \Theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Siendo R_z el ángulo de giro de la articulación, T_z el desplazamiento del eslabón, T_x la longitud del eslabón y R_x el giro del eslabón, al multiplicar estas 4 transformaciones básicas se obtiene.

$$A_i = \begin{vmatrix} \cos \Theta_i & -\sin \Theta_i \cos \alpha_i & \sin \Theta_i \sin \alpha_i & a_i \cos \Theta_i \\ \sin \Theta_i & \cos \Theta_i \cos \alpha_i & -\cos \Theta_i \sin \alpha_i & a_i \sin \Theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Para continuar se deben obtener las matrices homogéneas de paso A_1 hasta A_5 , para lo cual se usará la matriz A_i hallada anteriormente, donde se reemplazaran los valores de los ángulos de giro a fin de simplificar al máximo las matrices de paso.

$$A_1 = \begin{vmatrix} 1 & 0 & 0 & a_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (13)$$

$$A_2 = \begin{vmatrix} \cos \Theta_2 & -\sin \Theta_2 & \sin \Theta_2 & 0 \\ \sin \Theta_2 & \cos \Theta_2 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$\begin{aligned}
A_3 &= \begin{vmatrix} \cos \theta_3 & 0 & \sin \theta_3 & 0 \\ \sin \theta_3 & 0 & -\cos \theta_3 & 0 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{vmatrix} \\
A_4 &= \begin{vmatrix} \cos \theta_4 & -\sin \theta_4 & \sin \theta_4 & 0 \\ \sin \theta_4 & \cos \theta_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{vmatrix} \\
A_5 &= \begin{vmatrix} \cos \theta_5 & -\sin \theta_5 & \sin \theta_5 & 0 \\ \sin \theta_5 & \cos \theta_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{vmatrix}
\end{aligned} \tag{14}$$

Por último se define la matriz de transformación del robot que representará matemáticamente la cinemática del brazo robótico, para hallar la matriz de transformación de debe multiplicar las 5 matrices de paso definidas en el paso anterior.

$$T_0^5 = \begin{vmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{28}$$

$$\begin{aligned}
T_0^5 &= \begin{vmatrix} 1 & 0 & 0 & a_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} \cos \theta_2 & -\sin \theta_2 & \sin \theta_2 & 0 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} \cos \theta_3 & 0 & \sin \theta_3 & 0 \\ \sin \theta_3 & 0 & -\cos \theta_3 & 0 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{vmatrix} \\
&\quad * \begin{vmatrix} \cos \theta_4 & -\sin \theta_4 & \sin \theta_4 & 0 \\ \sin \theta_4 & \cos \theta_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} \cos \theta_5 & -\sin \theta_5 & \sin \theta_5 & 0 \\ \sin \theta_5 & \cos \theta_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{vmatrix}
\end{aligned}$$

Se necesario mencionar que el vector a de la matriz T, hace alusión a la relación de acercamiento entre en efector final y un objeto ajeno a la cadena cinemática, el vector s indica el eje de desplazamiento de la herramienta, en el caso de este documento, una ventosa, el vector n corresponde a la normal de los planos y por último, el termino el vector p representa el desplazamiento de la herramienta en los ejes sub 0.

$$\begin{aligned}
n_x &= \cos(\theta_5)(\sin(\theta_2) * \sin(\theta_4) + \cos(\theta_4)(\cos(\theta_2)\cos(\theta_3) - \sin(\theta_2) * \sin(\theta_3))) \\
&\quad + \sin(\theta_5)(\cos(\theta_4)\sin(\theta_2) - \sin(\theta_4)(\cos(\theta_2)\cos(\theta_3) - \sin(\theta_2) * \sin(\theta_3))) \\
n_y &= \cos(\theta_4)\cos(\theta_5)(\cos(\theta_2)\sin(\theta_3) + \cos(\theta_3) * \sin(\theta_2)) - \sin(\theta_4) \\
&\quad * \sin(\theta_5)(\cos(\theta_2)\sin(\theta_3) + \cos(\theta_3) * \sin(\theta_2)) \\
n_z &= \cos(\theta_4) * \sin(\theta_5) + \cos(\theta_5) * \sin(\theta_4)
\end{aligned}$$

$$s_x = \cos(th5)(\cos(th4)\sin(th2) - \sin(th4)(\cos(th2)\cos(th3) - \sin(th2) * \sin(th3))) \\ - \sin(th5)(\sin(th2)\sin(th4) + \cos(th4)(\cos(th2)\cos(th3) - \sin(th2) \\ * \sin(th3)))$$

$$s_y = \cos(th4) * \sin(th5)(\cos(th2)\sin(th3) + \cos(th3) * \sin(th2)) - \cos(th5) \\ * \sin(th4)(\cos(th2)\sin(th3) + \cos(th3) * \sin(th2))$$

$$s_z = \cos(th4) * \cos(th5) - \sin(th4) * \sin(th5)$$

$$a_x = \cos(th2) * \sin(th3) + \cos(th3) * \sin(th2) + \sin(th5)(\sin(th2)\sin(th4) \\ + \cos(th4)(\cos(th2)\cos(th3) - \sin(th2) * \sin(th3))) \\ + \sin(th4)(\cos(th2)\cos(th3) - \sin(th2) * \sin(th3))$$

$$a_y = \sin(th2) * \sin(th3) - \cos(th2) * \cos(th3) + \sin(th4)(\cos(th2)\sin(th3) \\ + \cos(th3) * \sin(th2)) + \cos(th4) * \sin(th5)(\cos(th2)\sin(th3) \\ + \cos(th3) * \sin(th2))$$

$$a_z = \sin(th4) * \sin(th5)$$

$$p_x = (77 * \sin(th2))/200 + (11 * \cos(th2) * \sin(th3))/50 + (11 * \cos(th3) \\ * \sin(th2))/50 + (9 * \sin(th4)(\cos(th2) * \cos(th3) - \sin(th2) \\ * \sin(th3)))/200 + 47/100$$

$$p_y = (11 * \sin(th2) * \sin(th3))/50 - (11 * \cos(th2) * \cos(th3))/50 + (9 \\ * \sin(th4)(\cos(th2) * \sin(th3) + \cos(th3) * \sin(th2)))/200$$

$$p_z = 5/8$$

Se debe resaltar que al tratarse de un prototipo no fue necesario realizar una simulación ya que los datos se obtendrían mediante pruebas realizadas más adelante.

7.1.7 piñón-cremallera.

Como se mencionó anteriormente, se adicionó un grado de libertad prismático al brazo robótico, para lo cual se diseñó una cremallera, la cual fue dividida en 3 secciones para facilitar su posterior impresión en 3D, para el correcto diseño de la cremallera fue necesario hallar los siguientes ítems. Ver tabla 21 y ver figura 28.

Tabla 21

Ítems para diseño de cremallera.

Ítem	Símbolo
Modulo	m
Ángulo de presión	α
Número de dientes	z
Altura de la cremallera	H
Paso primitivo	P
Diámetro primitivo	d
Diámetro exterior	d_a
Diámetro de fondo	d_f
Diámetro base	d_b
Distancia entre centros	a_s
Altura del diente	h
Altura de la cabeza del diente (Addendum)	h_a
Altura del pie del diente (Dedendum)	h_r
Longitud de cremallera rectangular	L

Nota. Blanco, Cebrian y García (s.f).

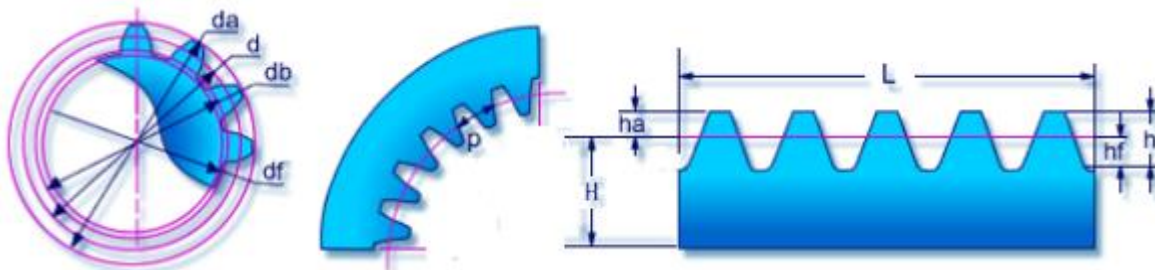


Figura 28. Medidas de piñón-cremallera, Blanco, Cebrian y García (s.f).

Antes de iniciar con los cálculos para resolver los ítems de la figura anterior, es necesario resaltar que se utilizó un piñón cuyo diámetro interior se ajustara al diámetro del eje del motor paso a paso ya seleccionado. Ver figura 29 para evidenciar las medidas del piñón utilizado.

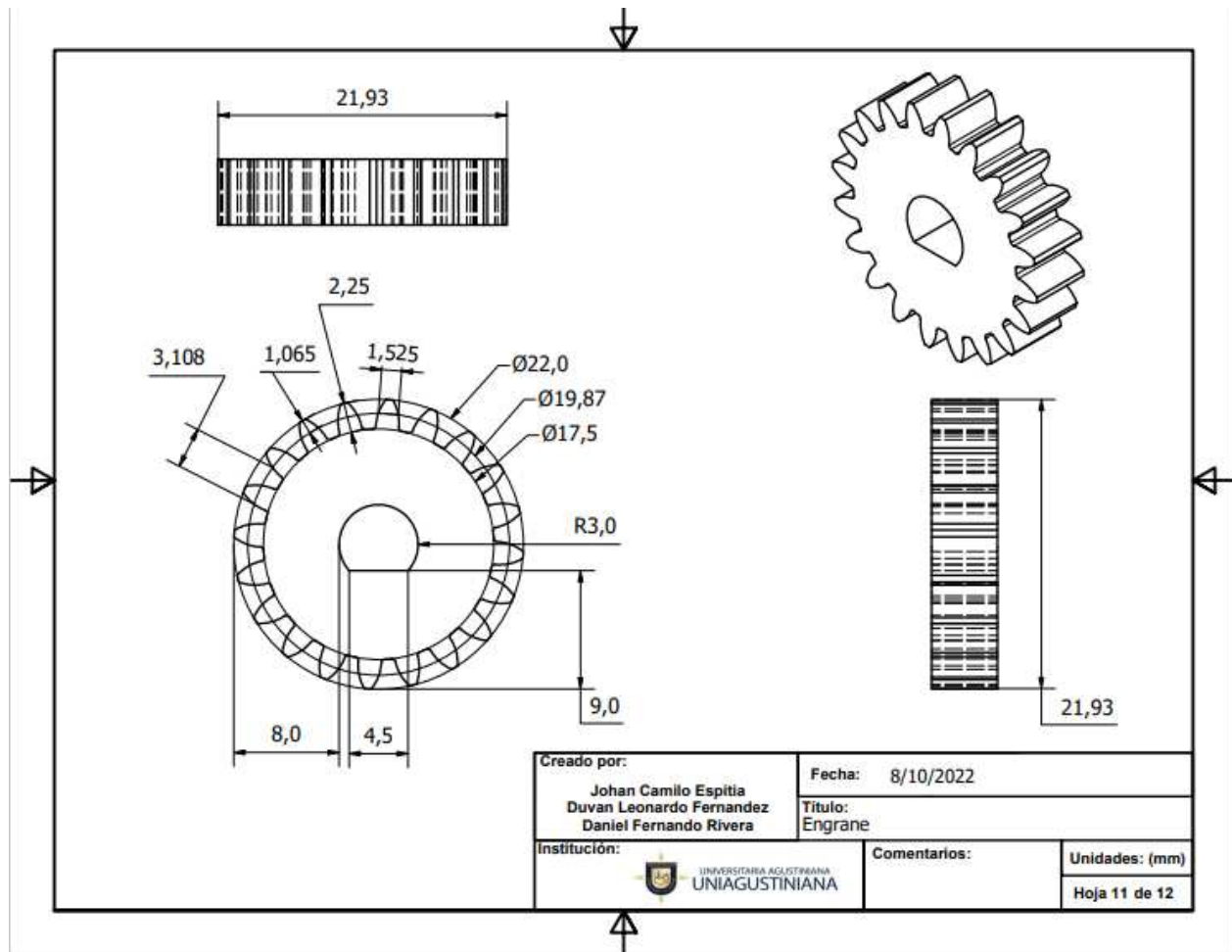


Figura 29. Piñón o engrane utilizado. Elaboración propia (2022).

Para comenzar los cálculos de la cremallera, se debe empezar por hallar los ítems que no necesiten cálculos al poder saber sus valores analizando la figura30, por lo que se puede decir que.

$$d = 19.87mm \quad (15)$$

$$d_a = 22mm \quad (16)$$

$$d_f = 17.5mm \quad (17)$$

$$z = 20 \quad (18)$$

Con los valores obtenidos mediante el análisis de la figura 30, se procede a hallar el valor del módulo, el cual indica el tamaño del piñón, para entender mejor las medidas ver figura 29.

$$m = \frac{d}{z} \quad (19)$$

$$m = \frac{19.87}{20} = 0.9935mm$$

El siguiente paso es hallar el valor del paso primitivo y las alturas de los dientes de la cremallera.

$$P = m * \pi = 3.1211mm \quad (20)$$

$$h = 2.25 * m = 2.2353mm \quad (21)$$

$$h_a = m = 0.9935mm \quad (22)$$

$$h_f = 1.25 * m = 1.2418mm \quad (23)$$

Una vez obtenidos estos valores se puede diseñar más fácilmente la cremallera, para la cual su longitud fue de 45.3 cm, no obstante como ya se ha mencionado para facilitar su impresión en 3D se optó por dividirla en 3 secciones con longitudes iguales. Ver figura 30.

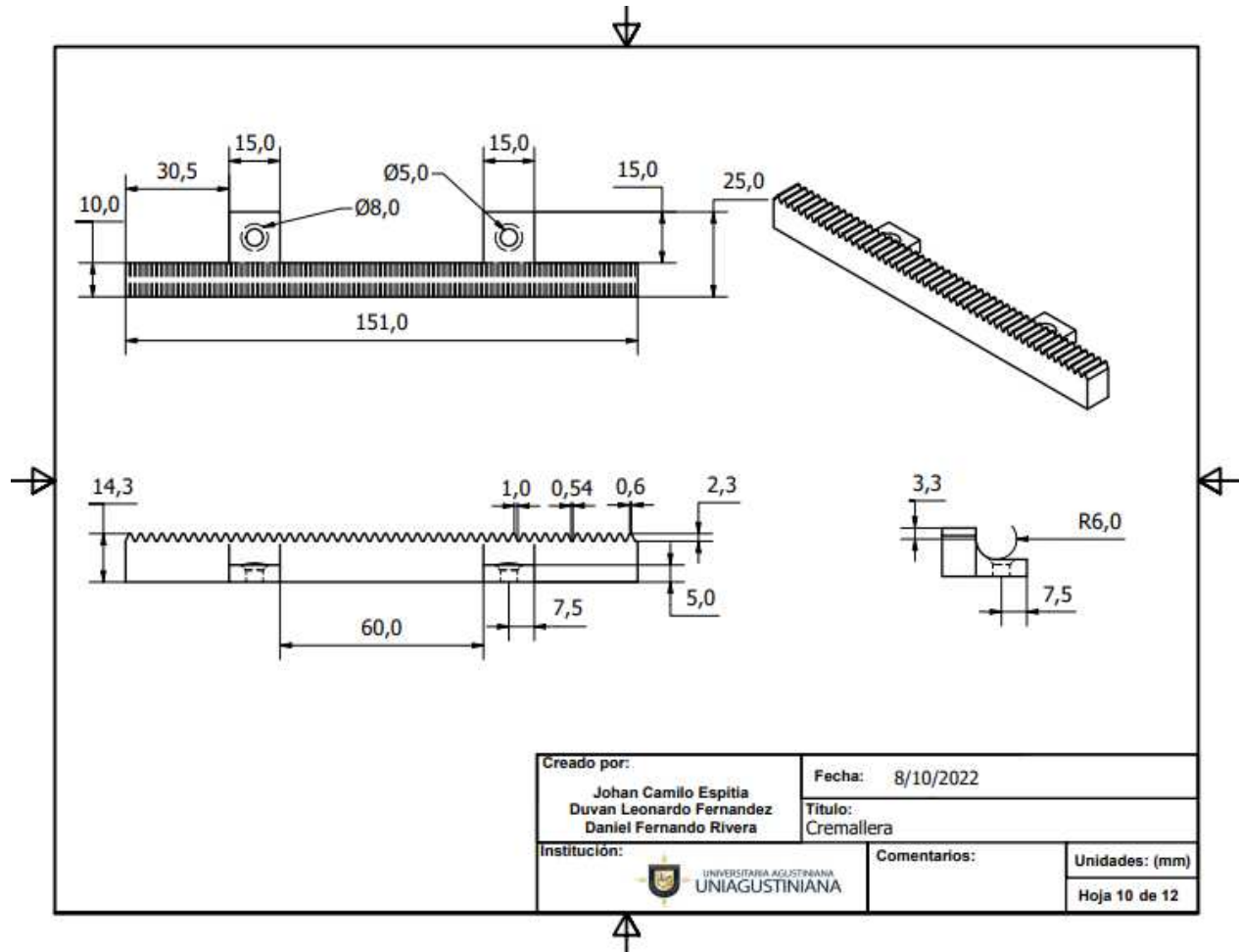


Figura 30. Planos de una sección de la cremallera. Elaboración propia (2022).

Con el diseño de la cremallera terminado se puede además determinar el número de pasos del piños y el número de vueltas que este deberá dar por la cremallera.

$$L = 453\text{mm} \quad (24)$$

$$N^{\circ} \text{ pasos} = \frac{L}{p} = 145.14 \quad (25)$$

$$N^{\circ} \text{ vueltas} = \frac{N^{\circ} \text{ pasos}}{z} = 7.257 \quad (26)$$

7.2 Planos electrónicos.

A continuación se presentan los planos electrónicos de los dos robots que conforman el prototipo planteado en este documento, junto con sus respectivos cálculos.

7.2.1 Circuito brazo robótico.

A la hora de diseñar el circuito del brazo robótico, lo primero a tener en cuenta fue el consumo de todos sus componentes, pues la fuente que se eligiera debía ser capaz de mantener todos los

elementos (actuadores, sensores y placa controladora), encendidos y sin caídas de corriente o voltaje. Debido a esto, se realizó una lista con todos los elementos y el consumo de cada uno. Ver tabla 22, tras esto, se identificó que el elemento con mayor voltaje de operación es el motor paso a paso (12V) y la corriente máxima que se puede consumir durante el uso del sistema es un total de 8.13 A, lo que significa que la fuente tenía que cumplir con dicho voltaje y mínimo esa cantidad de corriente, por tanto, se eligió una fuente de 12V y 10A.

Tabla 22.

Consumo de componentes

Elemento	Voltaje (V)	Corriente máxima (A)	Cantidad	Total (corriente)
ESP32	5	0,225	1	0,225
Leds	5	0,023	5	0,115
Sensor proximidad	5	0,06	1	0,06
Relé	5	0,09	1	0,09
Motor paso a paso	12	1	1	1
Servomotor mg995 y mg996r	6,5	2	3	6
Servomotor mg90s	6,5	0,7	1	0,7
Total			12	8,19

Nota. En la tabla se presentan los consumos de voltaje y corriente de los elementos presentes en el circuito, además del consumo total de corriente. Elaboración propia, (2022).

Sin embargo, en la tabla anterior se puede identificar que hay 3 voltajes distintos que varían según cada elemento (5, 6.5 y 12), por lo que la alimentación se tuvo que dividir en dos etapas, por un lado, la de control con consumo de 5V, que incluye la placa controladora y los sensores, y, por otro lado, la etapa de potencia, que a su vez se dividió también en dos, una parte para los servomotores con voltaje de 6.5V y otra para el motor paso a paso con voltaje de 12V.

De modo similar, para controlar la corriente que consumen los elementos del circuito, se optó por emplear resistencias con valores que correspondan a lo que cada elemento debía consumir como máximo. Por ejemplo, para los LED'S, se emplearon resistencias de 220Ω , las cuales, con 5V dan una corriente de aproximadamente 0.027 A. A continuación, se dan a conocer los cálculos realizados para las resistencias correspondientes a los servomotores y al motor paso a paso.

Siguiendo la ley de Ohm, se empleó la siguiente fórmula para calcular el valor correspondiente a las resistencias de los servomotores

$$R = \frac{V}{A} \quad (27)$$

Es entonces que, para los servomotores mg995 y mg996r, cuya corriente máxima es de 2.5A, se optó por elegir una resistencia que asegurara un paso de corriente inferior a 2A, por lo que, teniendo en cuenta que el voltaje de este servomotor es de 6.5V, el resultado fue:

$$R = \frac{6.5}{2} = 3.25\Omega \quad (28)$$

Debido a que este valor de resistencia exacto no se haya en el mercado, se optó por elegir una resistencia de 3.3Ω , valor que sólo difiere del calculado por 0.05Ω , adicionalmente, teniendo como prioridad la seguridad del servomotor, tal como se puede ver en el plano del circuito, también se emplearon fusibles de 2A que, en caso de alguna subida de corriente inesperada, proteja el elemento de quemaduras. Ver figura 31.

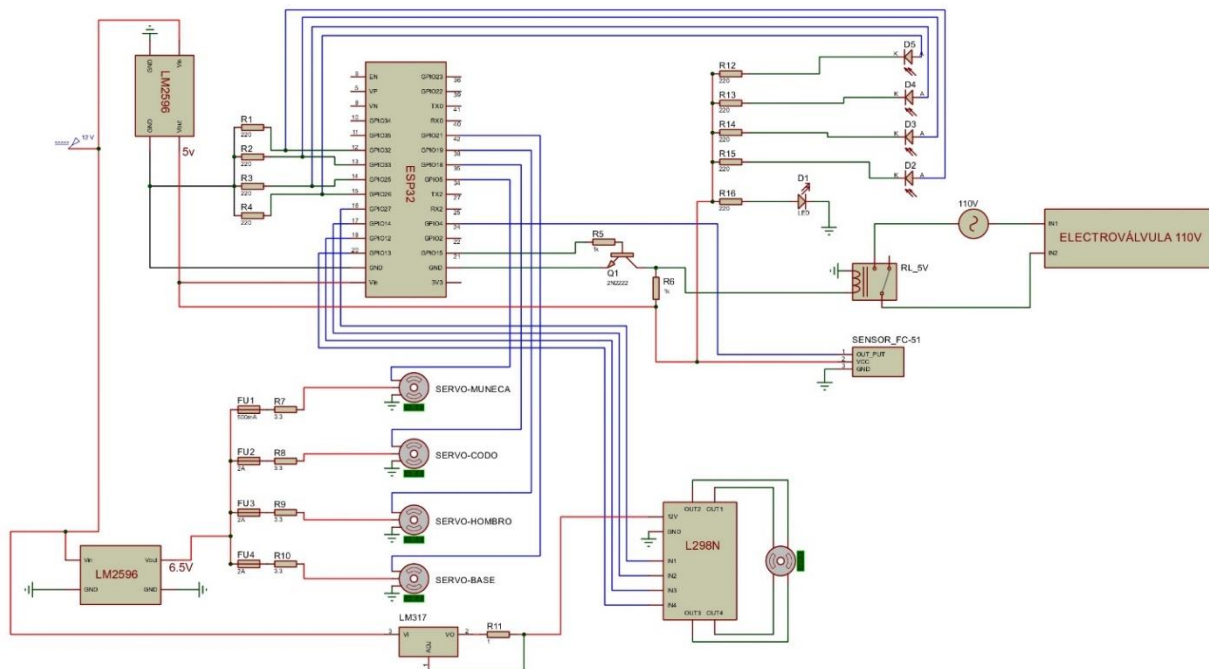


Figura 31. Plano electrónico del brazo robótico. Elaboración propia (2022).

Siguiendo la misma fórmula que en el cálculo anterior, para el servomotor mg90s, cuya corriente máxima es de 700mA, se empleó el cálculo de su respectiva resistencia, dando como resultado.

$$R = \frac{6.5}{0.7} \approx 9.29\Omega \quad (29)$$

Si bien, se realizaron pruebas de funcionamiento al servomotor del brazo, utilizando una resistencia de $10\ \Omega$ (valor comercial más próximo al calculado) el resultado no fue satisfactorio, el dispositivo, inicialmente, no tenía potencia, cuando llevaba carga (25-180g), esta simplemente iba balanceándose, sin la estabilidad que este servomotor debía darle, adicionalmente, en la mayoría de ocasiones, se le enviaba una orden desde la placa controladora y el mismo no la obedecía, emitía una vibración leve y hasta no recibir algún estímulo externo, no reaccionaba.

Fue entonces que se decidió emplear una resistencia de menor intensidad, en este caso una con valor de 6.8Ω , no obstante, el resultado fue nuevamente desfavorable, además, el motor no tenía fuerza, hubo ocasiones, en las que no obedecía a las órdenes enviadas. Esta misma prueba se realizó con una resistencia de 5Ω , dando el mismo resultado, cabe aclarar que, conforme la resistencia era menor, era más frecuente que el servomotor obedeciera y se percibía una mayor fuerza.

Finalmente, se puso a prueba una resistencia de $3.3\ \Omega$, mismo valor que las empleadas en los servomotores mg995 y mg996r, en este caso, sí se percibió que el servomotor tuviera fuerza al levantar objetos, así como en todos los ensayos hechos, el servomotor obedeció y se movió adecuadamente, por lo que se optó por dejar esta resistencia en el circuito final. Sin embargo, al igual que con los anteriores servomotores, se empleó un fusible que no permitiera la circulación de una corriente potencialmente peligrosa para el servomotor, en este caso la corriente máxima que puede soportar es de 700mA, por lo que el valor elegido para dicho fusible fue de 500mA.

Por último, y a diferencia de los servomotores, para el motor paso a paso se optó por emplear un limitador de corriente, pues este motor no consume corriente de manera mesurada y según la fuerza que se le estuviera aplicando, sino que este, estando quieto (estado en el que se encuentra la mayoría del tiempo), consume 3A o más, lo que genera que no sólo se caliente innecesariamente el sistema, sino que, tras un corto periodo de tiempo, el circuito del L298N, deja de funcionar.

Es entonces que se emplea el ya mencionado limitador. Un circuito integrado LM317 y una resistencia, este consiste en conectar la resistencia entre las terminales “Adjust” y “Vout” del integrado, se crea un circuito cerrado con voltaje de 1.25V y es allí donde se calcula la resistencia a aplicar empleando la ley de Ohm y teniendo en cuenta que la corriente máxima deseada es de 1A.

$$R = \frac{1.25}{1} = 1.25\Omega \quad (30)$$

Una vez obtenido el resultado de la resistencia, se requiere calcular el valor de la potencia que disipará el circuito, pues, si se coloca una resistencia que no soporte tal potencia, esta se quemará. Teniendo esto en cuenta y que el valor de resistencia a emplear será de 1Ω (valor más próximo hallado en el mercado), el cálculo queda así:

$$A = \frac{1.25}{1} = 1.25A \therefore P = 1.25 * 1.25 = 1.5625W \quad (31)$$

Con este valor presente, se decide que la resistencia que se empleará, será de 1Ω y $5W$, valor suficiente para disipar gran parte del calor generado y, por ende, no generar altas temperaturas en el circuito. Así mismo, se coloca al LM317 también un disipador de calor, puesto que este circuito integrado también genera calor con la disipación de corriente.

Por otro lado, debido a que el voltaje que el ESP32 entrega cuando está en “HIGH” es de $3.3V$ y no es suficiente para activar el relé de $5V$, se utilizó un transistor 2N2222 y dos resistencias de $1K\Omega$, tal como se ve en el circuito, con el fin de que al enviar la señal de “HIGH” del ESP32, el transistor envíe $5V$ al relé y lo active.

Finalmente, cabe aclarar que fue necesario emplear un sistema “pull down” con los LED’S receptores, pues sin ningún sistema, estos permanecían permanentemente enviando una señal de “HIGH” al ESP32, esto debido a que el LED no envía ninguna señal cuando no está siendo estimulado, es entonces que cualquier alteración en el ambiente o cualquier señal basura, causan que el ESP32 perciba que está recibiendo un “HIGH”. Sin embargo, al realizar el ya mencionado “pull down”, colocando una resistencia entre el pin del ESP32 conectado al LED y al GND del sistema, ya hay una señal de “LOW” llegando permanentemente al pin hasta que se active el LED y dicha señal cambie por un “HIGH”.

7.2.2 Circuito robot AGV.

Al igual que el circuito empleado en el brazo, antes de construir el circuito del AGV, fue necesario buscar una fuente de poder que proporcionara voltaje constante todos los dispositivos empleados, diferenciándose, en este caso, en que al ser un robot móvil, dicha fuente debía ser una batería, para la que fue necesario calcular su voltaje y autonomía con base en los voltajes y consumos de cada una de las piezas del circuito. Ver tabla 23.

Tabla 23.

Consumo de componentes robot AGV.

Elemento	Voltaje (V)	Corriente (A)	Cantidad	Total (corriente)
ESP32	5	0,225	1	0,225
Sensor QTR-8A	5	0,16	1	0,16
Sensor FC-51	5	0,06	1	0,06
Sensor TCS3200	5	0,002	1	0,002
Módulo L298N	12	0,036	2	0,072
Motorreductor	12	0,25	4	1
Total				1,519

Nota. En la tabla se presentan los consumos de voltaje y corriente de los elementos presentes en el circuito, además del consumo total de corriente. Elaboración propia (2022).

Una vez conocido el consumo de los elementos del circuito, se optó por una batería de 12V, puesto que los motores y el módulo trabajan con este voltaje, y de 10Ah, con los que, suponiendo que el circuito estuviera trabajando en su máxima potencia todo el tiempo (valores utilizados para realizar la tabla 23), la autonomía sería de más de 6 horas según los cálculos:

$$Wb = 12 * 10 = 120W \quad (32)$$

$$Wc = 12 * 1.519 = 18.228W \quad (33)$$

$$Autonomía = \frac{Wb}{Wc} = \frac{120}{18.228} \approx 6.58H \quad (34)$$

*Wb = potencia de la batería = voltaje de la batería * corriente de la batería*

*Wc = potencia del circuito = voltaje de la batería * corriente del circuito*

Al obtener los resultados de los cálculos, sólo fue necesario acoplar la batería al circuito, dividir los voltajes de la parte de control (ESP32 y sensores) y de potencia (módulos L298N y sus respectivos motores reductores), que en este caso son 5V y 12V respectivamente y, por último, conectar todo el sistema para poner el mismo en marcha y operación. Ver figura 32.

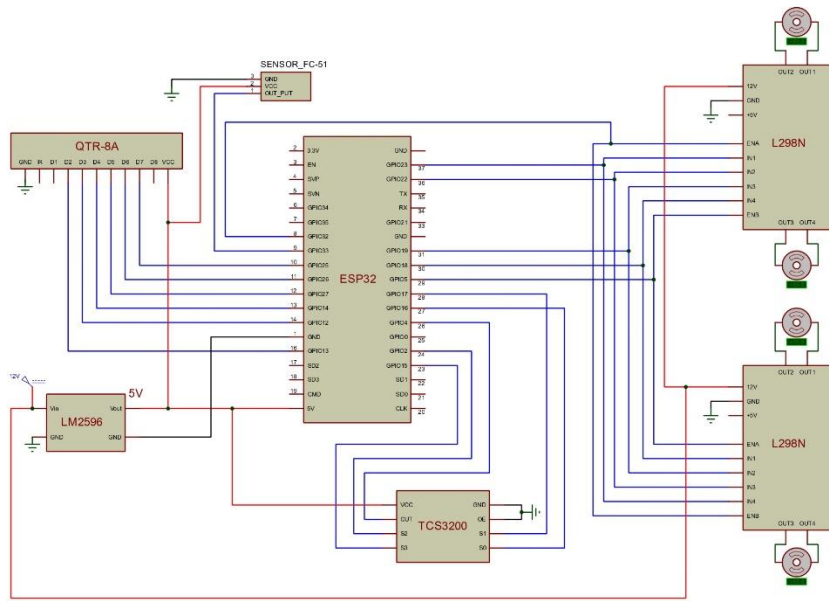


Figura 32. Plano electrónico del robot AGV. Elaboración propia (2022).

7.3 Otros planos.

Además de los planos del brazo robótico, el robot AGV y el piñón-cremallera. Se realizaron también los planos de los soportes del riel por el cual se desplazará el brazo robótico y los soportes para leds emisor-receptor que ayudarán al brazo robótico a identificar la ubicación de los pedidos en las estanterías. Ver figuras 33 y 34.

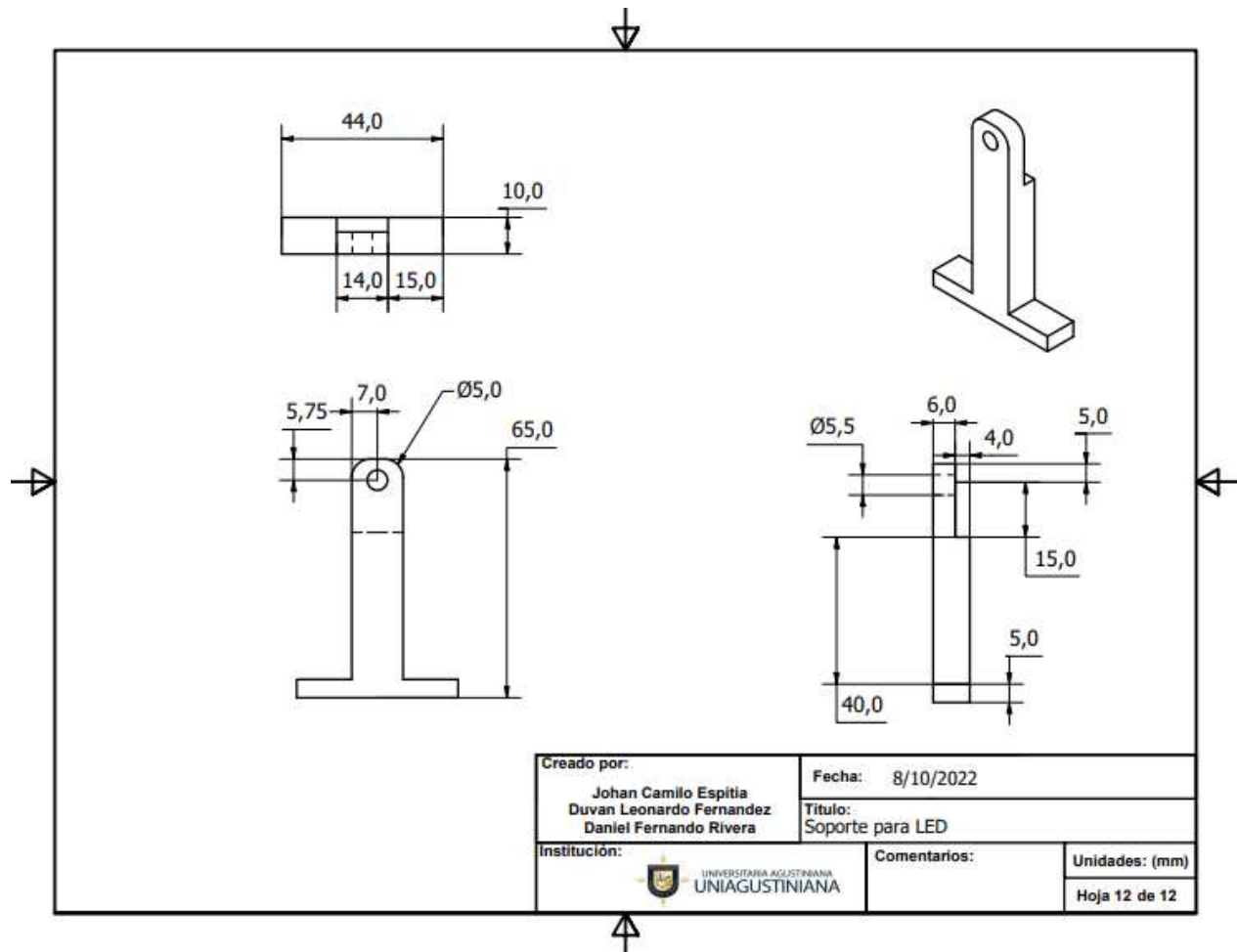


Figura 33. Soporte para leds emisor-receptor. Elaboración propia (2022).

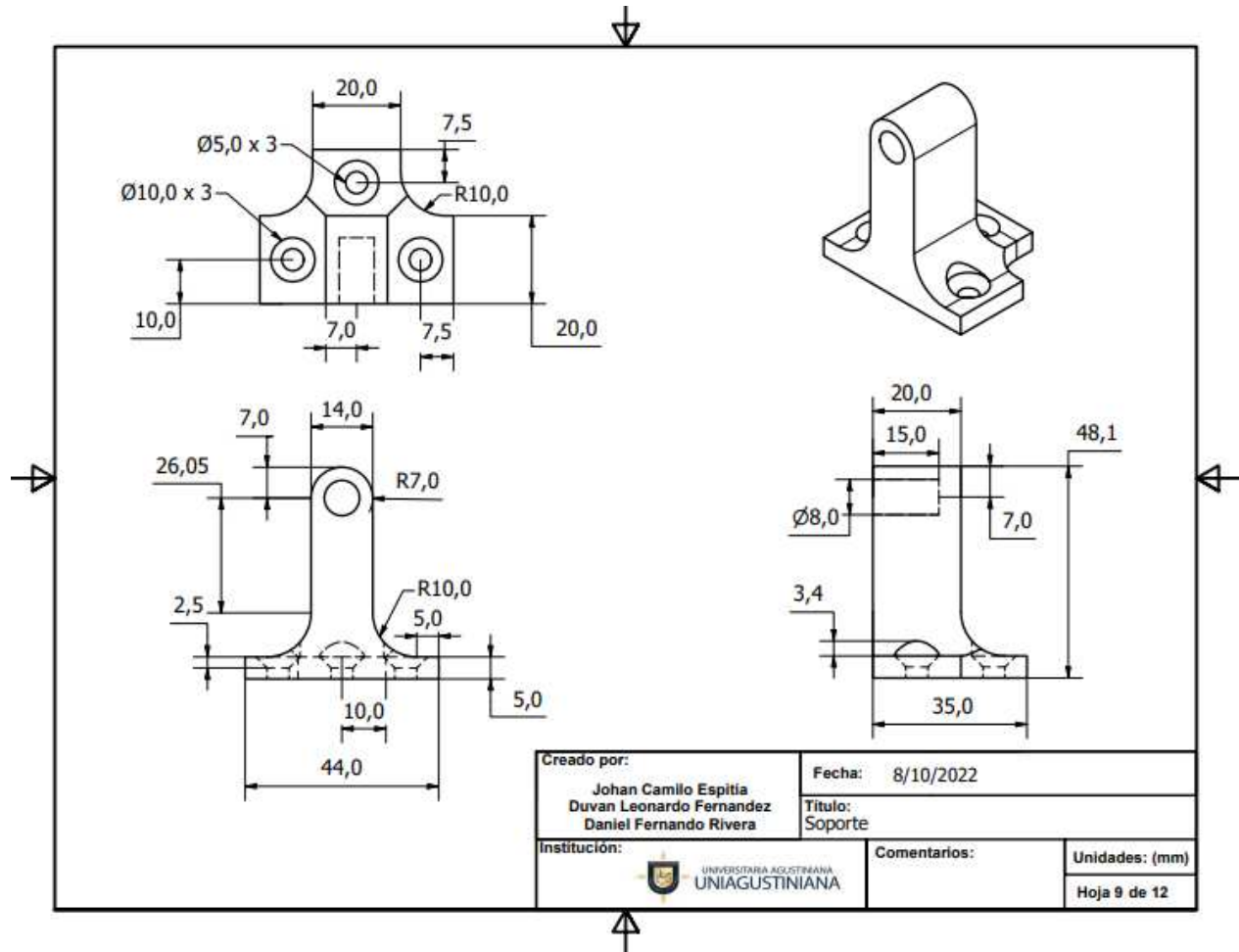


Figura 34. Soporte para riel. Elaboración propia (2022).

Además de los soportes, se implementó un sistema neumático, utilizando un eyector de vacío, una electroválvula, una ventosa y mangueras de conexión, por lo que fue necesario realizar los cálculos correspondientes para seleccionar los elementos correctos, para lograr esto se siguió la guía de CAMOZZI, 2018 para selección de elementos neumáticos.

Primero se tuvo que calcular la teoría de fuerza de agarre, la cual dicta la relación de fuerzas en la ventosa.

$$F_{th} = m * (g + a) * S$$

$$F_{th} = \text{teoría de fuerza de agarre}$$

$$m = \text{masa en kg}$$

$$g = \text{aceleración de la gravedad}$$

$$a = \text{sistema de aceleración.}$$

$$S = \text{factor de seguridad, valor mínimo de 1.5}$$

(35)

Teniendo en cuenta que para un peso de 282g se necesita un fuerza de 11kg-cm, el valor del sistema de aceleración será.

$$a = \frac{F}{m} = \frac{11kg * cm}{0.282kg} = \frac{1.078N}{0.282kg} = \frac{3.822m}{s^2} \quad (36)$$

$$F_{th} = 0.282kg \left(9.81 \frac{m}{s^2} + 3.822 \frac{m}{s^2} \right) * 1.5 = 5.76N \quad (37)$$

Una vez obtenido el valor teórico de la fuerza de agarre, se puede hallar la fuerza de succión, que ayudará a seleccionar el generador de vacío.

$$F_s = \frac{F_{th}}{n}$$

$$F_s = \text{fuerza de succión} \quad (38)$$

$$n = \text{numero de ventosas}$$

$$F_s = \frac{5.76N}{1} = 5.76N$$

Por último se selecciona el generador de vacío con ayuda de los rangos de succión que provee la guía.

$$\text{ventosa } 30mm \text{ } \varnothing = V_s \rightarrow 4.15L/min$$

$$V = n * V_s$$

$$V_s = \text{rango teórico de succión} \quad (39)$$

$$V = \text{rango requerido de succión}$$

$$V = 1 * \frac{4.15L}{min} = 4.15 \frac{L}{min}$$

Con los valores obtenidos se seleccionaron los siguientes elementos neumáticos. Ver figuras 35 y 36.



Ref: 86805

PNEUMATIC PNEU TEC 

ZK-06

Certificaciones:

Especificaciones	Descripción	Descargas
Tipo de operación	Generador de vacío	
Diámetro de boquilla	1mm ² - 1,5mm ² - 2mm ² (según referencia)	
Flujo máximo (l/min)	36 - 95 - 110 (según referencia)	
Presión de vacío	-3 a -8 Psi (-0,21 a -0,56kg/cm ²)	
Presión de operación	14 a 232 Psi	
Puerto conexión	1/8" - 1/4" - 3/8" NPT (según referencia)	
Material	Cuerpo en aleación de aluminio y sellos en NBR	

Figura 35. Eyectador de vacío seleccionado. Eléctricas Bogotá Ltda. (s.f).



Figura 36. Electroválvula seleccionada. Eléctricas Bogotá Ltda. (s.f).

Para finalizar, al adquirir los elementos neumáticos, se realizó la debida conexión de los mismos, misma que se puede visualizar en la figura 37.

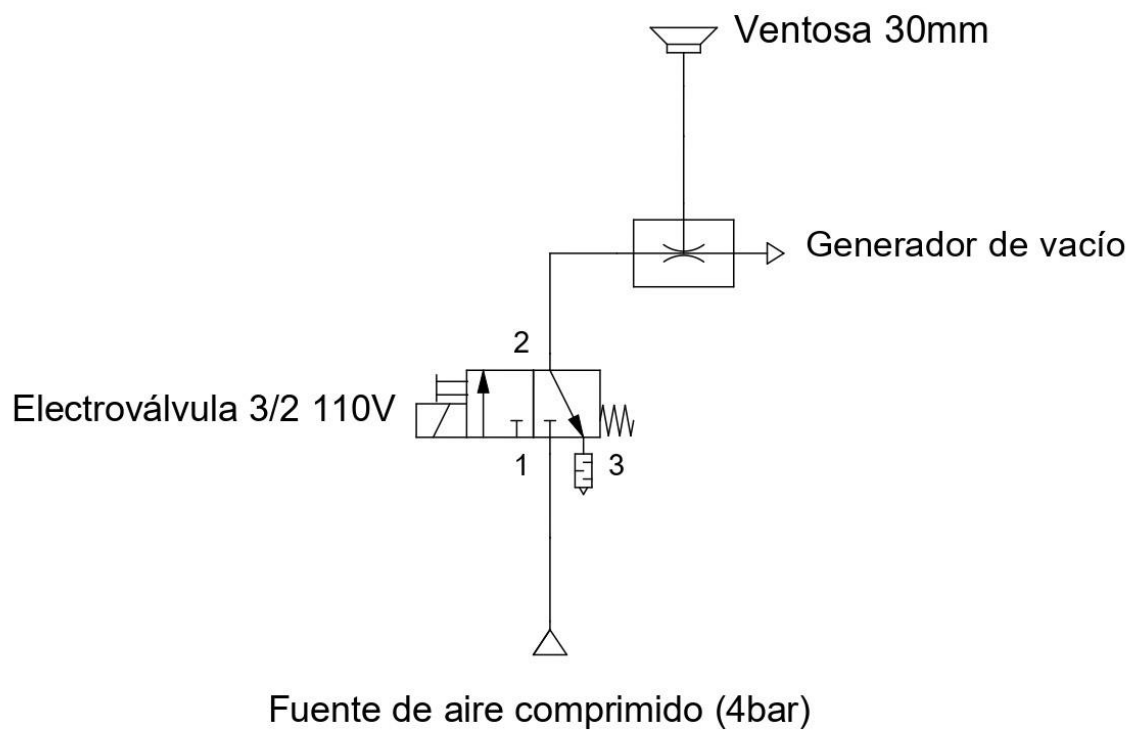


Figura 37. Plano de conexión neumática. Elaboración propia (2022).

Se debe mencionar que la electroválvula se activa gracias al relé del brazo robótico. Ver figura 31.

8. Programación y comunicación.

Para realizar la programación de ambos robots se optó por hacer uso del lenguaje C++, al compararlo con otro compatible con el micro controlador ESP32, esto permitió visualizar algunas ventajas que este lenguaje presentaba frente al otro. La comparación de características se realizó entre los lenguajes C++ y Python. Ver tabla 24.

Tabla 24.

Comparación C++ y Python.

Comparación lenguajes de programación		
Característica	C++	Python
Simplicidad de lenguaje	Cuenta con mas reglas de sintaxis y convenciones de programa	Su sintaxis es mas simple
Velocidad	Mas rapido	Mas lento
Gestión de memoria	Es manual al no tener recolector de basura	Su recolector de basura proporciona la gestion de memori de manera
Declaración de variables	Se deben declarar las variables junto a su tipo	Se pueden declarar variables sin necesidad de declara su tipo
Compilado	No se encuentra tipado estaticamente	Se encuentra tipado estaticamente
Bibliotecas	se deben descargar o fabricar	Bibliotecas integradas
Usos	Codificación de hardware y creación de sistemas	Aprendizaje automatico e inteligencia artificial

Nota. Los datos de la tabla fueron tomados de Ravikiran AS. Elaboración propia (2022).

Una vez completada la comparación se visualizó el hecho de que la velocidad del lenguaje C++ es superior a la del lenguaje de Python, esta característica fue la determinante para elegir el lenguaje C++, aun cuando el lenguaje Python resulta ser más sencillo de utilizar.

8.1 Diagramas de flujo y conexión.

Al programar los dos robots que conforman el prototipo y la interfaz que utilizará el usuario para realizar el pedido, se pensó sobre la forma en la que funcionaría y se comunicaría el sistema. Por lo que, se realizaron 3 diagramas de flujo (uno para el AGV, uno para el brazo robótico y uno

para la interfaz. Ver figuras 38-40, en donde se especifican las señales que se enviarán cada una de las pares que conforman el prototipo, así como también la información que se deberá ingresar en la interfaz para utilizarla. Adicionalmente, se realizó un diagrama de comunicación en el que se muestran el medio y orden de comunicación entre los elementos del sistema. Ver figura 41.

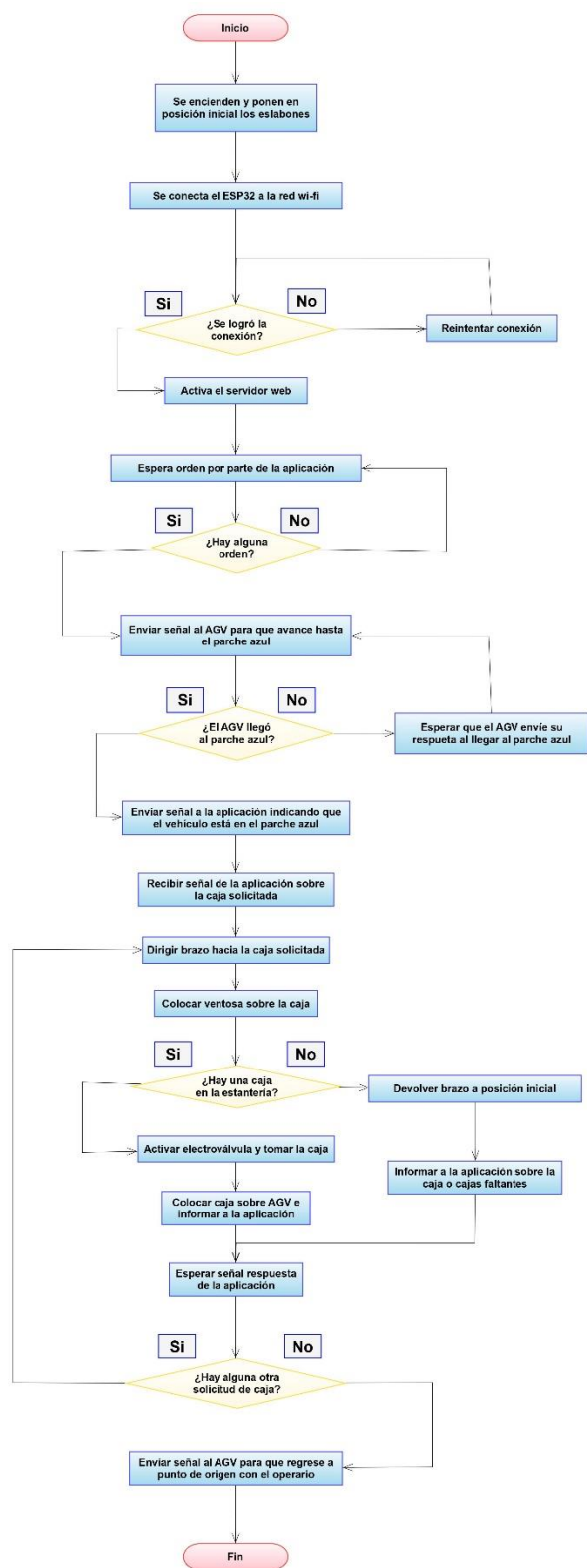


Figura 38. Diagrama de flujo del brazo robótico. Elaboración propia (2022).

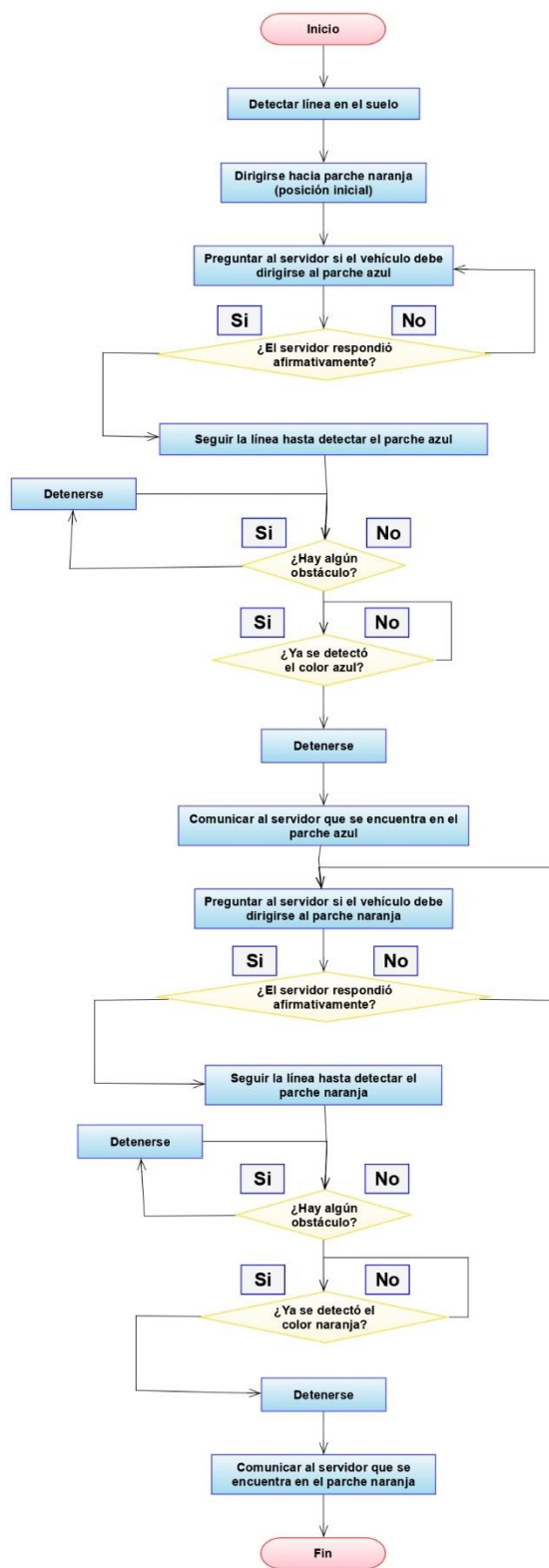


Figura 39. Diagrama de flujo robot AGV. Elaboración propia (2022).

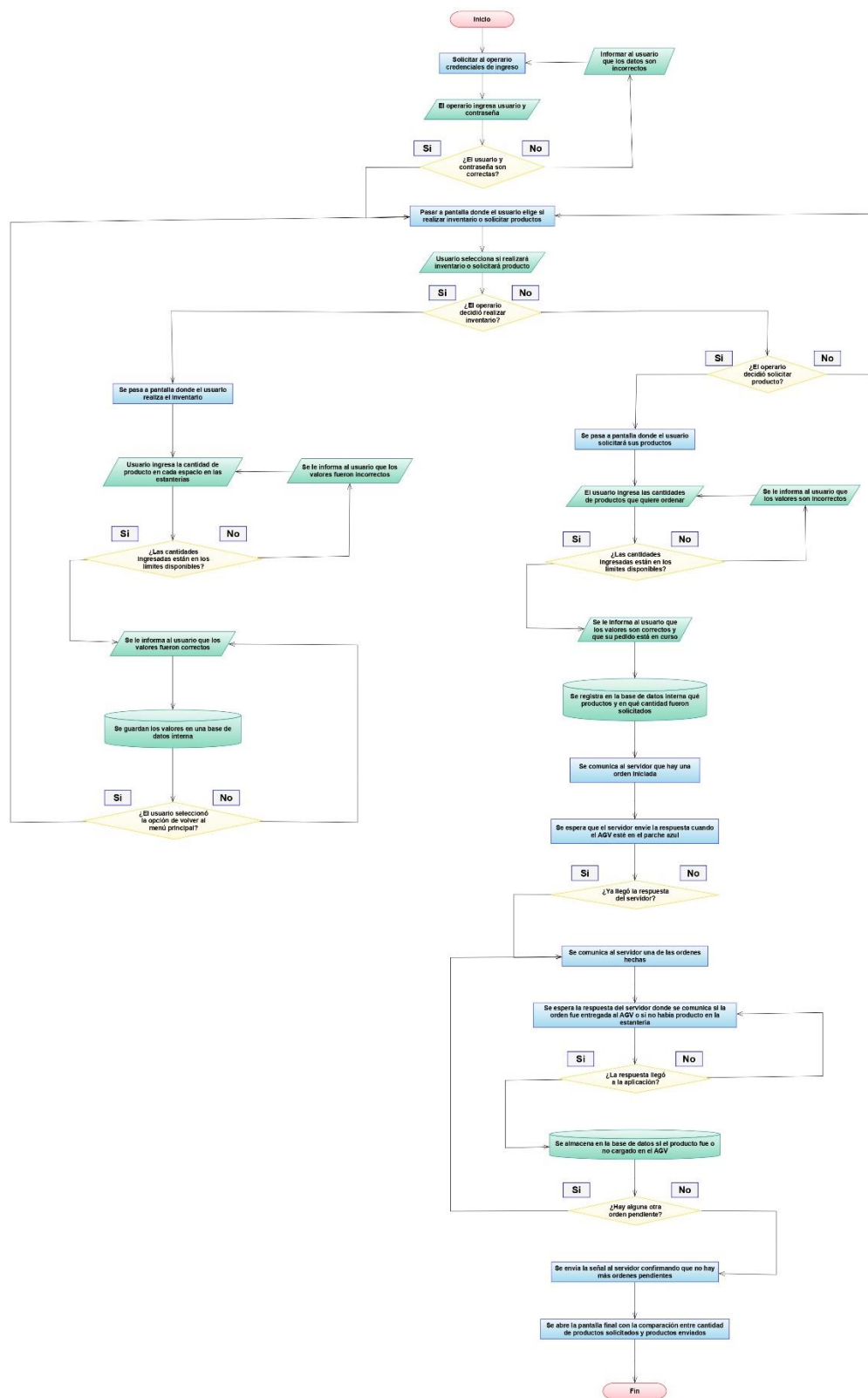


Figura 40. Diagrama de flujo interfaz de usuario. Elaboración propia (2022).

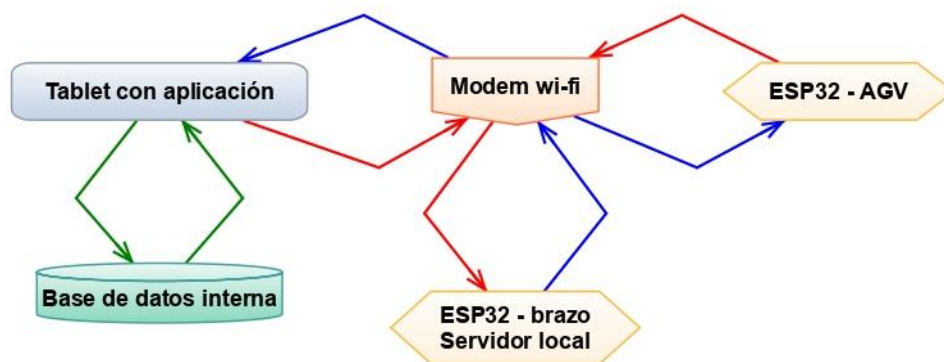


Figura 41. Diagrama de comunicación del sistema. Elaboración propia (2022).

8.2 Programación brazo robótico.

En este apartado se presenta la programación del brazo robótico, el cual no solo realiza el proceso de pick and place, entregándole el pedido al robot AGV para que este posteriormente lo entregue al usuario, sino que también se conecta a la red y desempeña el papel de servidor central, tal y como se puede ver en la figura 41, recibiendo los datos enviados tanto por la interfaz como por el robot AGV y enviando respuestas a los mismos.

Es entonces que, en su programa, lo primero que se realiza es la inclusión de librerías, en este caso se usaron: “WiFi.h”, encargada de que la tarjeta ESP32 active sus funciones de wifi, de forma que más adelante se pueda programar la conexión a red y el servidor, “Stepper.h” y “Servo.h”, librerías que permiten a la tarjeta controladora controlar motores paso a paso y servomotores respectivamente. Ver figura 42.

```
#include <WiFi.h>
#include <Stepper.h>
#include <Servo.h>
```

Figura 42. Inclusión de librerías. Elaboración propia (2022).

Posteriormente, A continuación, se declara lo relacionado con la conexión wifi y el servidor, por un lado, se inicia un servidor en el puerto 80 y se declaran cuáles serán las credenciales (ssid y contraseña) de la red wifi a la que se conectará la placa. Ver figura 43.

```
WiFiServer server(80);

const char* ssid = "Almacen_brazo_AGV";
const char* password = "JgT48F*hw53GyJ";
```

Figura 43. Declaración de credenciales. Elaboración propia (2022).

Seguido a esto, se inician las variables para controlar el motor paso a paso, estas incluyen los pines que se conectan de la placa ESP32 a el módulo L298N (Módulo conectado al motor paso a

paso) y los pasos por revolución que se manejan en el programa, en este caso, el motor hará 200 pasos para realizar una revolución completa. Por otra parte, también se inician las variables de los 5 sensores empleados (4 leds receptores y 1 sensor de proximidad fc-51), cada una declarada con su respectivo pin de conexión. Ver figura 44.

```
//Variables motor paso a paso
const int stepsPerRevolution = 200;

Stepper myStepper(stepsPerRevolution, 12, 13, 27, 14);

//Variables sensores

int S1 = 26;
int S2 = 25;
int S3 = 33;
int S4 = 32;

int SB = 4;
```

Figura 44. Declaración de variables motor paso a paso y sensores. Elaboración propia (2022).

Y finalmente, se inicializan, tanto el relé, con su respectivo pin de conexión, como el nombre de cada uno de los servomotores a controlar. Ver figura 45.

```
//Relay

int relay = 15;

//Declaración de servomotores

Servo muneca;
Servo codo;
Servo hombro;
Servo base;
```

Figura 45. Declaración de variables de relé y servomotores. Elaboración propia (2022).

Tras declarar la mayoría de variables y pines a emplear a lo largo de la programación, se realiza el setup del programa, el cual inicia declarando a qué pin está conectado cada servomotor y enviando cada uno de estos a su posición inicial, dejando un segundo de espera entre cada posicionamiento, esto con el fin de evitar que se muevan todos los servomotores al mismo tiempo y causen un accidente o golpeen algún objeto cercano. Así mismo, se define a cada uno de los sensores como señales que entrarán en el sistema y al relé como una señal de salida. Ver figura 46.

```

base.attach(21);
base.write(85);

delay(1000);

hombro.attach(19);
hombro.write(70);

delay(1000);

codo.attach(18);
codo.write(170);

delay(1000);

muneca.attach(5);
muneca.write(60);

delay(1000);

pinMode(S1, INPUT);
pinMode(S2, INPUT);
pinMode(S3, INPUT);
pinMode(S4, INPUT);
pinMode(SB, INPUT);

pinMode(relay, OUTPUT);

```

Figura 46. Inicialización de variables en “void setup”. Elaboración propia (2022).

Luego de esto, se establece la velocidad del motor paso a paso como 60 RPM, y mediante un ciclo while, se activa a este mismo para que se dirija hacia el led receptor 1, cuando el motor traslada el brazo hasta este punto, el sistema se encuentra completamente en posición inicial. Una vez este proceso termina, el ESP32 se conecta a la red wifi y hace uso de una IP fija, pues es a esta dirección a la que se conectarán el AGV y la aplicación, por lo que si esta llegara a variar, sería necesario re configurar todos los dispositivos. Ver figura 47.

```

myStepper.setSpeed(60);
Serial.println("Velocidad en 60 RPM");

val = 1;
while(LS1 != HIGH){
  myStepper.step(val);
  LS1 = digitalRead(S1);
  delay(10);
}

// Conexión WIFI
WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}

IPAddress ip(192, 168, 43, 2);
IPAddress gateway(192, 168, 43, 1);
IPAddress subnet(255, 255, 255, 0);
WiFi.config(ip, gateway, subnet);

Serial.println("");
Serial.println("WiFi conectado");
Serial.println(WiFi.localIP());
server.begin();

```

Figura 47. Inicio del paso a paso y conexión a red. Elaboración propia (2022).

No obstante, antes de programar el bucle “void loop”, fue necesario programar otros bucles con tareas repetitivas en el programa, estos se hicieron con el fin de no añadir líneas de programación de manera innecesaria, pues en caso de que se necesite realizar una de estas acciones, sólo basta con llamar al bucle en el que se encuentre, facilitando los cambios en estas tareas que se repetirán varias veces, adicionalmente, en caso de que fuera necesario realizar un cambio en estas acciones, solo basta con cambiar lo que hay en el bucle.

En este caso, los primeros bucles que se programaron fueron los de las 12 diferentes posiciones a las que se puede trasladar el robot para alcanzar las cajas en cada una de las estanterías, adicionalmente, se programó en cada uno de estos bucles la condición de que con ayuda del sensor de proximidad, el ESP32 pueda diferenciar si hay o no una caja en la posición seleccionada, esto para evitar que se active la electroválvula de manera innecesaria y poder informar a la aplicación en caso de que se solicite una caja que realmente no esté en el inventario.

En la primera posición del brazo robótico, el paso a paso traslada el brazo hasta el led receptor 2 o sensor 2, luego, los servomotores giran a sus respectivos grados mediante ciclos for, con el objetivo de no realizar movimientos bruscos y una vez el gripper está sobre el pedido o en la posición donde este debería estar, se emplea el sensor de proximidad indicando si se encuentra o no. Si el pedido está, la electroválvula se activa, y de la misma forma en que se trasladaron todos los eslabones a este punto, los mismos vuelven a su posición inicial con el pedido sujeto. En caso de que el pedido no se encuentre, no se activa la electroválvula, pero el brazo regresa igualmente a su posición inicial. Esta tarea, como ya se mencionó, se repitió con cada una de las 12 posiciones, modificando por separado el sensor al que debe trasladarse el brazo con ayuda del motor paso a paso y los grados a los que rotan los servomotores en cada una de estas posiciones. En consecuencia, se crearon 12 bucles de posición. Ver figuras 48 y 49.

<pre> void pos1() { LS1 = digitalRead(S1); LS2 = digitalRead(S2); if (LS1 == HIGH) { while (LS2 != HIGH) { myStepper.step(-val); LS2 = digitalRead(S2); } myStepper.step(-40); } for (bg = 85; bg < 177; bg = bg + 1) { base.write(bg); delay(10); } delay(200); for (cg = 170; cg > 115; cg = cg - 1) { codo.write(cg); delay(10); } delay(500); delay(700); for (hg = 75; hg > 50; hg = hg - 1) { hombro.write(hg); delay(10); } delay(2500); LSB = digitalRead(SB); </pre>	<pre> delay(2500); LSB = digitalRead(SB); if (LSB == LOW) { digitalWrite(relay, HIGH); obj = 1; } else { obj = 0; } delay(1000); delay(1000); for (cg = 115; cg < 160; cg = cg + 1) { if (hg < 120) { hg = hg + 1; } codo.write(cg); hombro.write(hg); } delay(200); for (mg = 60; mg > 50; mg = mg - 1) { muneca.write(mg); delay(10); } delay(200); for (bg = 177; bg > 85; bg = bg - 1) { base.write(bg); delay(10); } </pre>	<pre> for (cg = 160; cg < 170; cg = cg + 1) { codo.write(cg); delay(10); } delay(200); for (hg = 120; hg > 70; hg = hg - 1) { hombro.write(hg); delay(10); } for (mg = 50; mg < 60; mg = mg + 1) { muneca.write(mg); delay(10); } } delay(200); LS1 = digitalRead(S1); if (LS1 == LOW) { while (LS1 != HIGH) { myStepper.step(val); LS1 = digitalRead(S1); } } delay(200); </pre>
--	--	--

Figura 48. Bucle de posición 1. Elaboración propia (2022).

```

837 void pos1() {
931
932 void pos2() {
1019
1020 void pos3() {
1109
1110 void pos4() {
1195
1196 void pos5() {
1285
1286 void pos6() {
1373
1374 void pos7() {
1463
1464 void pos8() {
1549
1550 void pos9() {
1639
1640 void pos10() {
1725
1726 void pos11() {
1815
1816 void pos12() {
...

```

Figura 49. Los 12 bucles de posición agrupados. Elaboración propia (2022).

De manera similar a estos 12 bucles, se programaron 6 bucles de entrega (ver figuras 50 y 51), cada uno con una de las posiciones en las que el brazo robótico puede colocar las cajas sobre el AGV. Adicionalmente, se creó un bucle que decide de forma ordenada en qué orden realizar los 6 bucles de entrega, con el fin de que el proceso de colocar los productos sea siempre ordenado y las cajas no choquen entre sí al ser colocadas. Ver figura 52.

```

void entregal() {

for (cg = 170; cg > 110; cg = cg - 1) {
  codo.write(cg);
  delay(10);
}
for (mg = 60; mg > 30; mg = mg - 1) {
  muñeca.write(mg);
  delay(10);
}
for (hg = 70; hg > 40; hg = hg - 1) {
  hombro.write(hg);
  delay(10);
}
delay(500);
digitalWrite(relay, LOW);
delay(1000);
for (cg = 110; cg < 170; cg = cg + 1) {
  if (hg < 70 && mg < 60) {
    hg = hg + 1;
    mg = mg + 1;
  }
  codo.write(cg);
  hombro.write(hg);
  muñeca.write(mg);
  delay(10);
}
}
}

```

Figura 50. Bucle de entrega 1. Elaboración propia (2022).

```

1938 void entregal() {
1966 void entrega2() {
1977 void entrega3() {
2008 void entrega4() {
2044 void entrega5() {
2075 void entrega6() {

```

Figura 51. Los 6 bucles de entrega agrupados. Elaboración propia (2022).

```

void entrega() {
  delay(500);
  int X = 1;
  if (etg == 1 && X == 1) {
    entrega3();
    etg = etg + 1;
    X = 0;
  }
  if (etg == 2 && X == 1) {
    entrega4();
    etg = etg + 1;
    X = 0;
  }
  if (etg == 3 && X == 1) {
    entrega5();
    etg = etg + 1;
    X = 0;
  }
  if (etg == 4 && X == 1) {
    entrega6();
    etg = etg + 1;
    X = 0;
  }
  if (etg == 5 && X == 1) {
    entrega1();
    etg = etg + 1;
    X = 0;
  }
  if (etg == 6 && X == 1) {
    entrega2();
    etg = 1;
    cst = 1;
    azul = 0;
    X = 0;
  }
}
}

```

Figura 52. Bucle que define siguiente bucle de entrega. Elaboración propia (2022).

Una vez explicados los bucles externos al “void loop”, se realiza este, el cual como se observa en la figura 53 inicia, recibiendo datos de cualquier cliente que se conecte al servidor y una vez haya alguno conectado, hay 2 variables que guardan la información que el cliente envía, por un lado la variable tipo “char” llamada “c”, que lee y toma individualmente todos los caracteres recibidos y por otro lado la variable tipo “string”, “header”, que toma los caracteres de “c” y los une, formando cadenas con la información recibida.

```

WiFiClient client = server.available();

if (client) {
  Serial.println("Nuevo cliente");
  String currentLine = "";
  while (client.connected()) {
    if (client.available()) {
      char c = client.read();
      Serial.write(c);
      header += c;
    }
  }
}

```

Figura 53. Conexión con cliente y obtención de datos. Elaboración propia (2022).

Por otra parte, Con respecto a la comunicación con los clientes, cabe aclarar que para que alguno se conecte al servidor, aparte de estar conectado a la misma red wifi, este tiene que ingresar al servidor mediante un navegador e ingresando como enlace la IP fija del servidor más una barra inclinada “/” seguida del complemento de la acción, petición o mensaje que envía, como por ejemplo podría ser: “http://192.168.43.2/orden111”.

Es entonces que, cuando un cliente se conecta e ingresa con alguno de los enlaces ya establecidos, la variable “header” eventualmente tendrá entre su cadena de caracteres un texto como el siguiente: “GET /orden111”, y estos textos posteriormente, mediante el comando “header.indexOf” se buscarán, y al cliente que se conecte con uno de estos, se le enviará una respuesta. Por ejemplo, el AGV únicamente se conecta mediante tres enlaces, “http://192.168.43.2/device1”, “http://192.168.43.2/naranja” y “http://192.168.43.2/azul”. Cuando se comunica con el primero, está esperando la orden para dirigirse ya sea hacia el sitio de entrega (donde se encuentra el operario) o hacia el sitio de carga (donde está el brazo robótico), cuando se comunica con el segundo y tercer enlace, avisa al servidor que ya se encuentra en alguno de los dos sitios ya mencionados.

Por otro lado, Por su lado, el servidor, para enviar una respuesta a los clientes, emplea la función `client.print()`; y en este caso, imprime la variable de tipo “String” “rta”, que, según las variables internas del mismo programa, es diferente en cada ocasión dependiendo de lo que la interfaz haya comunicado y de lo que el mismo brazo ya haya realizado. Siguiendo el ejemplo de la comunicación con el AGV. Ver figura 54, cuando este se conecta mediante el enlace terminado en “/device1”, el servidor puede enviar “rta” de tres maneras distintas, como un “0” (si no se necesita que el vehículo se desplace), como un “20” (si se necesita que el carro llegue al sitio de carga) y, por último, como un “30” (si ya el pedido está listo y el AGV debe volver con el operario al sitio de entrega).

```

if (header.indexOf("GET /device1") >= 0) {
  if (cst == 1) {
    rta = "20";
  }
  else if (cst == 2) {
    rta = "30";
  }
  else {
    rta = "0";
  }
  client.print(rta);
}
if (header.indexOf("GET /naranja") >= 0) {
  Serial.println("AGV en Naranja");
  azul = 0;
  cst = 0;
}
else if (header.indexOf("GET /azul") >= 0) {
  Serial.println("AGV en azul");
  azul = 1;
  cst = 0;
}
}

```

Figura 54. Respuesta del servidor a cliente en “/device1”. Elaboración propia (2022).

De igual manera, la aplicación del operario se puede conectar al servidor con 4 tipos de enlaces diferentes, el primero terminado en “/ordenini”, este avisa al servidor que se ha solicitado una orden y es ahí cuando se le ordena al AGV dirigirse hacia el brazo robótico. Seguido a este comando, la aplicación envía el segundo, terminado en “/espera”, al cual se le debe responder con un “10” cuando el AGV haya llegado al sitio de carga, en caso contrario, se le envía un “0”. Tras informar que el vehículo está en posición, la interfaz envía uno de los 24 posibles comandos de órdenes, dos por cada posición de estantería, puesto que de cada posición se puede solicitar un máximo de dos existencias según disponibilidad de inventario.

Para las órdenes de una sola existencia por estante, cuyo enlace termina en “1”, como “/orden111” o “/orden231”, se pueden enviar 2 tipos de respuesta a la aplicación, una terminada en “0”, que significa que la caja fue dejada con éxito en el vehículo, y otra terminada en “1”, que representa lo opuesto, es decir que no había realmente una caja en la estantería y, por tanto, el brazo no llevó nada al AGV. Sin embargo, para los pedidos de dos existencias por estante, que terminan en “2”, como “/orden112”, hay tres respuestas posibles, las terminadas en “0” y “1” pero también una terminada en “2”, que significa que, de las 2 existencias solicitadas, sólo se entregó una. Ver figura 55.

```

else if (header.indexOf("GET /orden111") >= 0) {
    pos1();
    if(obj == 1){
        entrega();
        rta = "110";
    }
    else{
        rta = "111";
    }
    client.print(rta);
}
else if (header.indexOf("GET /orden112") >= 0) {
    pos1();
    if(obj == 1){
        entrega();
    }
    else{
        rta = "111";
    }
    if (rta != "111"){
        pos1();
        if(obj == 1){
            entrega();
            rta = "110";
        }
        else{
            rta = "112";
        }
    }
    client.print(rta);
}

```

Figura 55. Respuesta del servidor ante solicitud de caja. Elaboración propia (2022).

Posterior a esta orden, y tras recibir la respuesta correspondiente, desde la interfaz se puede, por un lado, enviar otra orden y repetir este proceso hasta que en el AGV haya un máximo de 6 existencias, o puede enviar el enlace “/ordent”, que le comunica al servidor que ya la orden está completa en el AGV y que este ya debe desplazarse de vuelta al sitio de entrega junto al operario, ara que este tome las existencias solicitadas y si lo necesita, realice nuevamente otra solicitud.

8.3 programación del Robot AGV

En la programación realizada para el control del AGV, la cual inicia de igual forma a la programación del brazo robótico, declarando las librerías a utilizar, en este caso también se usó “WiFi.h”, para que la tarjeta controladora habilite sus funciones de conexión a redes, sin embargo, para este programa, la segunda librería que se emplea es la de “HTTPClient.h”, que habilita el programa para conectarse a un servidor web y actuar como un cliente en el mismo. Es entonces que, se declaran en primer lugar, las credenciales de la red wifi a la que se conectará el dispositivo (ssid y contraseña), siendo esta la misma red a la que se conectó el brazo y, en segundo lugar, se declaran los diferentes enlaces a los que se podrá conectar el AGV dentro del servidor, ver figura 56.

```

#include <WiFi.h>
#include <HTTPClient.h>

const char* ssid = "Almacen_brazo_AGV";
const char* password = "JqR48F*hw53GyJ";

const char* esp32Server = "http://192.168.43.2/device1";
const char* ServerAzul = "http://192.168.43.2/azul";
const char* ServerNaranja = "http://192.168.43.2/naranja";

```

Figura 56. Conexión a red y declaración de enlaces a utilizar. Elaboración propia (2022).

Posteriormente a este procedimiento, se declaran todas las variables relacionadas a los sensores y actuadores del vehículo, estando en primer lugar, los módulos L298N, encargados del control de los motorreductores que dan movimiento y dirección al vehículo, en segundo lugar, se encuentra el sensor de color tcs 3200, que, como su nombre lo indica, escanea los colores sobre los que está el vehículo y devuelve un valor para el color rojo, uno para el color verde y otro para el color azul, en tercer lugar, se encuentra el módulo QTR-8a que cuenta con 8 sensores que diferencian el color blanco del color negro y que será utilizado para seguir una línea negra en un suelo blanco. Finalmente, está el sensor de proximidad, que indicará cuando haya un obstáculo en frente del vehículo. Ver figura 57.

```

//Pines L298N
int veld = 5;
int derp = 19;
int dern = 18;
int izqp = 23;
int izqn = 22;
int veli = 32;

//Pines sensor color tcs3200
int s0 = 16;
int s1 = 17;
int s2 = 2;
int s3 = 15;
int out = 4;

//Pines sensor de blanco y negro
int D2 = 13;
int D3 = 12;
int D4 = 14;
int D5 = 27;
int D6 = 26;
int D7 = 25;

//Pin sensor de proximidad
int SP = 33;

```

Figura 57. Declaración de sensores y actuadores del AGV. Elaboración propia (2022).

Tras declarar estas variables, se realiza el “void setup”, en el que, primero se toman todas estas variables y se definen a las variables asociadas a los sensores como “inputs” y a las variables que representan a los actuadores como “outputs”, seguido a esto, se realiza la conexión a la red wifi y, por último, se posiciona el AGV sobre el parche de color naranja. Para llevar a cabo esta tarea, se utiliza un ciclo “while” y se llama a los bucles “andar”, “colorfun” y “quieto”, que, aunque se

explicarán a detalle más adelante, son respectivamente los que realizan las acciones de desplazamiento por la línea del suelo sin salirse ni colisionar, leer en qué color está posicionado el AGV y de detener al mismo. Ver figura 58.

```
void setup() {
  Serial.begin(115200);
  pinMode(veld,OUTPUT);
  pinMode(vell,OUTPUT);
  pinMode(derp,OUTPUT);
  pinMode(dern,OUTPUT);
  pinMode(izqp,OUTPUT);
  pinMode(izqn,OUTPUT);
  pinMode(D2,INPUT);
  pinMode(D3,INPUT);
  pinMode(D4,INPUT);
  pinMode(D5,INPUT);
  pinMode(D6,INPUT);
  pinMode(D7,INPUT);
  pinMode(SF,INPUT);
  pinMode(s0,OUTPUT);
  pinMode(s1,OUTPUT);
  pinMode(s2,OUTPUT);
  pinMode(s3,OUTPUT);
  pinMode(out,INPUT);
  digitalWrite(s0,HIGH);
  digitalWrite(s1,HIGH);

  WiFi.begin(ssid,password);
  Serial.println("conectando");
  while(WiFi.status() != WL_CONNECTED){
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("Conectado a la red con la IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  Serial.println("posicionando vehiculo");

  colorst = 0;
  while(colorst != 2){
    andar();
    colorfun();
  }
  quieto();
}
```

Figura 58. Inicialización de variables y ubicación en posición inicial. Elaboración propia (2022).

Ahora bien, antes de definir el “void loop”, al igual que con el programa del brazo, en el código se colocaron bucles y variables que realizan acciones por fuera del mencionado “loop”, en primer lugar están las variables tipo “String” “getRequest”, “getRequest1” y “getRequest2”, vistas en la figura 59, cada una de estas, al ser llamada, se conecta al servidor local con los enlaces previamente definidos. Concretamente, “getRequest” se conecta mediante el enlace “http://192.168.43.2/device1”, “getRequest1” mediante el enlace “http://192.168.43.2/naranja” y “getRequest2” mediante el enlace “http://192.168.43.2/azul”. Una vez conectadas, estas variables reciben la información que el servidor imprima en ellas como respuesta y dicha respuesta será utilizada en el programa según sea el caso.

```
String getRequest(const char* esp32Server){
  HTTPClient http;
  http.begin(esp32Server);

  int httpResponseCode = http.GET();

  String payload = "...";

  if(httpResponseCode > 0){
    payload = http.getString();
  }
  else{
    payload = "0";
  }

  http.end();
  return payload;
}
```

Figura 59. Variable que se conecta a “/device1”. Elaboración propia (2022).

De igual forma, se realizan los bucles, siendo uno de ellos el llamado “colorfun”, en este se realizan 2 acciones, la primera es emplear el sensor de color para obtener un valor de cada color (rojo, verde y azul) y colocar dicho valor en una variable nombrada con el color que le corresponde. Tras esto, se comparan en un rango dichos valores para identificar si estos 3 corresponden a los colores que diferencian el sitio de entrega con el de carga (siendo en este caso los colores naranja y azul respectivamente), si no coinciden con los de ninguno, según cada caso, a la variable “colorst” se le asigna uno u otro valor. Ver figura 60.

```
void colorfun(){

  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);
  rojo = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
  digitalWrite(s3, HIGH);
  azul = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
  digitalWrite(s2, HIGH);
  verde = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

  //Azul
  if ((rojo >= 25 && rojo <= 28) && (verde >= 12 && verde <= 14) && (azul >= 5 && azul <= 8)){
    colorst = 1;
  }
  //Naranja
  else if ((rojo >= 6 && rojo <= 9) && (verde >= 16 && verde <= 19) && (azul >= 15 && azul <= 18)){
    colorst = 2;
  }
  else{
    colorst = 3;
  }
}
```

Figura 60. Bucle para leer color. Elaboración propia (2022).

Posteriormente, se encuentran los diferentes bucles, siendo los más sencillos, los 4 movimientos, nombrados “adelante”, “atrás”, “izquierda” y “derecha”, además del estado “quieto”, estos al ser llamados en la programación, le envían al módulo L298N la señal para realizar alguno de los