Implementación de un sistema de detección y evasión de obstáculos de bajo costo para drones de ala fija

Presentado por

Cindy Lorena Aviles Correa

Jeison David Garzón Bernal

Fundación Universitaria Los Libertadores

Facultad de Ingeniería y Ciencias Básicas

Programa de Ingeniería Aeronáutica

Bogotá D.C, Colombia

Implementación de un sistema de detección y evasión de obstáculos de bajo costo para drones de ala fija

Presentado por

Cindy Lorena Aviles Correa

Jeison David Garzón Bernal

título de

Ingeniero Aeronáutico

Dirigida por

Didier Aldana Rodríguez

Codirector

Diego Leonardo Ávila Granados

Presentada a

Programa de Ingeniería Aeronáutica

Fundación Universitaria Los Libertadores

Bogotá D.C, Colombia.

Notas de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Bogotá DC, noviembre de 2021.

Las directivas de la Fundación Universitaria Los Libertadores, los jurados calificadores y el cuerpo do- cente no son responsables por los criterios e ideas expuestas en el presente documento. Estos corres- ponden únicamente a los autores y a los resultados de su trabajo.

Dedicatoria

Este trabajo de grado es dedicado a tantas personas que no alcanzaría a nombrarlas a todas, sin embargo, le debo todo el ánimo, las correcciones, a quién en momentos de dificultad y en ocasiones en que quería desfallecer me impulsaba a seguir, me recordaba que el esfuerzo y sacrificio siempre trae su recompensa, en que los cambios nos complementan como personas, en que necesitamos caer para volvernos a levantar, eso y muchas otras cosas más se las debo a ella, ella es mi mamá.

Cindy Lorena Aviles

Quisiera dedicar este trabajo a todas las personas que me han apoyado, acompañado, guiado a realizar este y muchos más logros, por los buenos consejos que nunca faltaron ni por las buenas disposiciones de las personas más allegadas, a mi madre y a mi padre que siempre han luchado por dar lo mejor.

Jeison David Garzón

Agradecimientos

Nuestro profundo y más sincero agradecimiento a la Universidad Fundación Universitaria Los Libertadores, sobre todo al profesor Didier Aldana, director del semillero de investigación y del presente proyecto, por brindarnos su conocimiento, su apoyo, su tiempo y adicional a eso los componentes electrónicos que fueron empleados, logrando que el objetivo económico se cumpliera.

Agradecemos de igual forma al teniente coronel Cáceres y los tenientes Ardila, Medaglia y Casallas, por compartir sus conocimientos, su paciencia, abrir un espacio tanto en sus trabajos como en sus vidas ayudando a enfocar el proyecto, a llevarlo a cabo de la mejor manera, al prestar las instalaciones del CETIA y sus herramientas, el conocimiento adquirido brindado por ustedes es un premio y una gran ganancia.

Jeison y Cindy

No puedo dejar de lado a la persona que me brindó su casa, su familia, para cumplir tanto con el proyecto de grado como con las pasantías, a sus explicaciones y correcciones logrando que la tesis se encaminara al objetivo propuesto, a realizarse en su totalidad, gracias profe Zoila Velandia.

Mis amigas Gina y Yurany, por comprender que hay prioridades, por brindar apoyo incondicional, por apoyarme y dar ánimos en los momentos más cruciales, agradezco infinitamente su amistad.

Cindy Lorena Aviles

Estoy particularmente agradecido por la ayuda brindada por nuestro tutor Didier Aldana que nos apoyó, guio y condujo a la finalización de este proyecto, Sin usted y sus virtudes, su paciencia y constancia este trabajo no lo hubiésemos logrado. Sus consejos fueron siempre útiles cuando no salían de nuestro pensamiento las ideas para escribir lo que hoy hemos logrado. Usted formó parte importante de esta historia con sus aportes profesionales que lo caracterizan. Muchas gracias.

Me gustaría expresar mi más sincero agradecimiento a mi madre Sandra Bernal y padre Martin Bayona motor de apoyo con sus consejos tiempo dedicado, por apoyarme aun cuando mis ánimos decaían. En especial, para darme palabras de apoyo y un abrazo reconfortante para renovar energías.

Estoy particularmente agradecido con Fabián Romero por sus palabras de aliento y paciencia en todos los aspectos, por su fortaleza emocional brindada con paz, la ayuda que me ha brindado ha sido sumamente importante, en los momentos más tormentosos siempre ayudándome, no fue sencillo culminar, pero siempre estuviste ahí motivador, diciendo que lo lograría.

Me gustaría ofrecer mi agradecimiento especial a mi compañera Cindy Aviles que sin ella este reto hubiese sido aún más complicado, por su voluntad y dedicación, paciencia y comprensión, por sus buenos sentimientos y conocimientos.

Jeison David Garzón

Índice general

Índice de figuras

Índice de Tablas

Abreviaturas

AA	Aeronaves Autónomas
AV	Vehículos Autónomos
ESC	Electronic Speed Controller
FAC	Fuerza Aérea Colombiana
FULL	Fundación Universitaria Los Libertadores
GPS	Global Positioning System
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
INEGI	Instituto Nacional de Estadística y Geografía
LED	Light Emitting Diode
LiDAR	Laser Imaging Detection and Ranging
Lipo	Polímero de litio
PCB	PolyChlorinated Biphenyls
PT	Potothreads
RPA	Robotic Process Automation
SIAP	Sistema de Información Agroalimentaria y Pesquera
SGM	Servicio Geológico Mexicano
SSP	Seguridad Pública
UCAV	Unmanned Combat Air Vehicle
UAV	Unmanned aerial vehicle
V/STOL	Vertical/Short Take-Off and Landing

Implementación de un sistema de detección y evasión de obstáculos de bajo costo para drones de ala fija

Resumen

El presente trabajo fue diseñado e implementado en un prototipo de sistema de detección de obstáculos por medio de diferentes sensores como: cámara HD, sensor LiDAR, y un sensor infrarrojo, instalado en un dron de ala fija tipo ala Zagi fabricado para el proyecto con el propósito de realizar pruebas en tierra. La metodología consistió en fusionar los diferentes códigos de control de los dos sensores y la cámara para la detección de obstáculos y enviar señales de control a los servomotores que gobiernan las superficies de control (alerones) y al motor brushless que da empuje al dron y fuerza. Se plantearon varios escenarios que se implementaron mediante una lógica definida en tablas de verdad que mediante la señal de control generada modifica las revoluciones del motor Brushless y la posición de los alerones según la distancia a la que sea detectado un objeto, para generar una respuesta evasiva. El resultado es un prototipo didáctico funcional de bajo costo, que permite entender el funcionamiento y la lógica de los sistemas de detección y evasión para drones.

Palabras claves: Dron, Lidar, Infrarrojo, detección de obstáculos,

Implementation of a low-cost obstacle detection and avoidance system for fixed-wing drones

Abstract

This work was designed and implemented in a prototype obstacle detection system using different sensors such as: HD camera, LiDAR sensor, and an infrared sensor, installed in a fixed wing drone Zagi wing type manufactured for the project in order to perform ground tests. The methodology consisted of fusing the different control codes of the two sensors and the camera for obstacle detection and sending control signals to the servomotors that control the control surfaces (ailerons) and the brushless motor that gives thrust and power to the drone. Several scenarios were proposed and implemented by means of a logic defined in truth tables that by means of the generated control signal modifies the revolutions of the brushless motor and the position of the ailerons according to the distance at which an object is detected, in order to generate an evasive response. The result is a low-cost functional didactic prototype, which allows to understand the operation and logic of detection and avoidance systems for drones.

Keywords: Drone, Lidar, Infrared, obstacle detection.

Capítulo 1

Introducción

La intervención de las tecnologías y la practicidad de las mismas, ha causado una evolución industrial e intelectual a pasos agigantados, el uso de diversas tecnologías ha podido lograr que el hombre surque los cielos, conquistando nuevos retos en la optimización de las máquinas haciendo una eficiencia productiva y eficaz, es allí, donde se estudian variables de temperatura, combustible, fuerzas externas, entre muchas otras, en estos años contemporáneos es muy común hablar y ver máquinas dominando los cielos ya sean tripuladas o no, es decir, controladas remotamente o simplemente programadas, llamando estas tecnologías objetos voladores no tripulados (Dron o UAV) [1], tal definición es adquirida por la Universidad de Valencia, este tipo de artefactos son tan eficientes que hay prototipos de uso militar armados (Unmanned Combat Air Vehicle) UCAV, por lo cual carecen de piloto a bordo, es controlado en tiempo real con un desempeño, agresivo, hostil, de vigilancia y monitoreo, esta tecnología no es asequible ya que es de un presupuesto elevado, por los servicios que ofrece y la calidad de su construcción, este artefacto militarmente lo posee Estados Unidos [2] como lo divulga el portal de noticias BBC News mundo a través de su portal web.

Internacionalmente, México usa sus drones dependiendo de sus sectores como: secretarías de la Defensa Nacional (Sedena), Marina (Semar), y Seguridad Pública (SSP), así como el Centro de Investigación y Seguridad Nacional (Cisen), el Servicio Geológico Mexicano (SGM), el Sistema de Información Agroalimentaria y Pesquera (SIAP) de la Secretaría de Agricultura, y el Instituto Nacional de Estadística y Geografía (INEGI), con fines usados en vuelos de inteligencia o para seguridad, obtener insumos para cartografía, gestión de cultivos y áreas boscosas así como lo evidencia el portal de divulgación de tecnología “expancion.mx” [3], en sectores de seguridad gubernamental muchos países tienen reglas estrictas dando como ejemplo un roce entre Venezuela y Colombia acusando a Colombia por invasión de espacio aéreo con un artefacto no tripulado [4], o la sorprendente coordinación y programación de la exhibición de los juegos olímpicos de Tokio 2020 [5].

En Colombia la tecnología de drones está en fase de investigación y en proceso de desarrollo de prototipos. Como ejemplo se puede dar el “Vant Solventus” desarrollado por jóvenes de La Fundación Universitaria Los Libertadores en colaboración con la Universidad de Sao Paulo en Brasil [6]. Adicionalmente empresas del sector aeronáutico han desarrollado algunos prototipos principalmente en el área de la seguridad y vigilancia. Con el fin de contribuir en el campo académico e investigativo este proyecto tiene como objetivo general implementar un prototipo de sistema de detección y evasión de obstáculos de bajo costo para drones de ala fija para pruebas en tierra. Y será desarrollado a través de los siguientes objetivos específicos: i) Realizar la simulación y el montaje del circuito electrónico para la detección de obstáculos mediante sensor de distancia infrarroja, Sensor

LiDAR y cámara de video, ii) Seleccionar el sistema de control y actuación de las superficies primarias de control de vuelo del dron, a partir de un algoritmo de decisión implementado en un micro controlador, iii) Usar un modelo ya existente de dron de ala fija y modificarlo según la necesidad del proyecto en un programa de diseño y iv) Ensamblar el sistema de detección de obstáculos para la implementación de un prototipo de dron de ala fija, escogido para realizar pruebas en tierra.

El dron de ala fija tipo ala delta (Zagi) fabricado para el desarrollo del proyecto cuenta con tres sensores de reconocimiento de proximidad de obstáculos y entorno, gobernado por un Arduino operable y funcional elaborado con materiales de bajo costo, incursionando en tecnologías de bajo presupuesto, genera respuestas de control eficaces para detectar y evadir objetos, respuestas logradas mediante el desarrollo y articulación sinérgica de códigos de control que toman la información de los sensores y generan señales de control (con base en una lógica que fue definida mediante tablas de verdad), para gobernar los actuadores que gobiernan las superficies primarias de control de vuelo y el sistema de propulsión (el motor), cuya eficacia y eficiencia se validó mediante pruebas en tierra comprobando los escenarios de detección planteados.

Capítulo 2

Planteamiento del Problema

Dentro del funcionamiento actual de las aeronaves remotamente pilotadas de ala fija (RPA) que operan en Colombia, no se cuenta con la capacidad de percibir con efectividad obstáculos en un trayecto delimitado, el cual es definido por el operador (piloto) de manera manual o implementando un grupo de waypoints, que son marcas en una ruta que elimina la necesidad de la línea de vista permanente entre el piloto y la aeronave. Sin embargo, estos trayectos que son seguidos de manera autónoma por la aeronave no son completamente infalibles para impedir posibles colisiones y se compromete la integridad de la aeronave y la toma de datos de la misión asignada. Aunque se han invertido recursos para convertir los RPA en Aeronaves Autónomas (AA) que detecten y evadan obstáculos con eficacia, no se han implementado estos cambios debido a encontrarse en una fase temprana de desarrollo. Por tal motivo el desarrollo actual de Aeronaves Autónomas adelantadas por entidades del estado como las Fuerzas Militares, se centra en disminuir las limitaciones operacionales y mitigar los riesgos implícitos de la pérdida del control.

¿Cuál es la funcionalidad que tiene la implementación de un sistema de control autónomo de bajo costo, para detección y evasión de obstáculos en la operación de un dron de ala fija, validado mediante pruebas en tierra?

2.1 Objetivos

2.1.1 Objetivo General

Implementar un prototipo de sistema de detección y evasión de obstáculos de bajo costo para drones de ala fija para pruebas en tierra.

2.1.2 Objetivos Específicos

- Realizar la simulación y el montaje del circuito electrónico para la detección de obstáculos mediante sensor de distancia infrarroja, Sensor LiDAR y cámara de video.
- Seleccionar el sistema de control y actuación de las superficies primarias de control de vuelo del dron, a partir de un algoritmo de decisión implementado en un microcontrolador.
- Usar un modelo ya existente de dron de ala fija y modificarlo según la necesidad del proyecto en un programa de diseño.
- Ensamblar el sistema de detección de obstáculos para la implementación de un prototipo de dron de ala fija, escogido para realizar pruebas en tierra.

2.2 Justificación

El presente proyecto nace de las ideas planteadas en el semillero de investigación en Drones perteneciente al programa de Ingeniería Aeronáutica de la Fundación Universitaria Los Libertadores y se desarrolla en el marco del proyecto de investigación llamado “Desarrollo e implementación de un sistema de control de vuelo de bajo costo para detección y evasión autónoma de obstáculos en una aeronave no tripulada tipo multirotor.”

En la actualidad el uso de drones ha incursionado de forma directa en espacios cotidianos y han sido programados para realizar labores específicas brindando apoyo en ambientes laborales, recreativos, educativos, de comunicaciones, vigilancia, salvamento y los de uso militar, entre otros. Por eso es pertinente la implementación y el uso de tecnologías para mejorar la efectividad y eficacia de estas aeronaves, principalmente en la planeación de trayectorias de control, la realización de tareas cada vez más automatizadas que se realicen con seguridad operacional mitigando riesgo de colisión con estructuras o seres vivos. Con el fin de mitigar los impactos de estos riesgos, la Fuerza Aérea Colombiana (FAC) y la Fundación Universitaria Los Libertadores (FULL) han unido esfuerzos para plantear un proyecto de investigación que permita implementar a futuro una solución de bajo costo, a los problemas actuales de evasión y detección que se presenten.

Capítulo 3

Marco Teórico

3.1 Antecedentes Históricos

Los drones se remontan a finales del siglo XIX, cuando el inventor serbio-americano Nikola Tesla demostró el uso de frecuencias de radio para controlar a distancia pequeños dispositivos. A lo largo del siglo XX, los militares incorporaron y ampliaron el concepto de control remoto de Tesla, con el primer avión no tripulado al final de la Primera Guerra Mundial y las bombas radiocontroladas, los equipos con cámara y el designador láser se utilizaron cómo estrategia militar [7]. Hasta 1907 cuando los hermanos Louis y Jacques Bréguet inventan el primer cuadricóptero. Aunque este primer acercamiento tenía la gran deficiencia de tener que ser tripulado por 4 personas, una por cada motor, por lo que no podría ser considerado como un dron. El vehículo de los hermanos llegó a levantarse un par de pies del suelo. Un invento que claramente sirvió de base para los siguientes trabajos realizados en este sector de la aeronáutica[8]. El ingeniero aeronáutico T. Claude Ryan el cual funda una empresa llamada “Compañía Aeronáutica Ryan” para 1922, construyen varios aviones de importancia histórica y técnica, incluyendo cuatro diseños innovadores de V / STOL , pero su avión de producción más exitoso fue la línea Ryan Firebee de drones no tripulados utilizados como drones objetivo y vehículos aéreos no tripulados, esta línea fue uno de los primeros drones propulsados a chorro siendo uno de los drones más utilizados jamás construidos[9].

3.2. Tarjeta Arduino

En el mundo de la electrónica y la programación existe una gran variedad de tarjetas programables o microcontroladores que varían en características y costos, destacándose el microcontrolador Arduino (ver Figura 1a) que presenta una relación costo beneficio más que aceptable; según la información encontrada en [10] indica que Arduino es un microcontrolador de código abierto y gratuito que no requiere membresía o licencia para ser utilizado. Se estableció en el 2005 para estudiantes, entusiastas de la programación para que lo usaran sin inconvenientes conectando sensores, actuadores, y demás componentes; tiene un amplio uso en proyectos sencillos y presenta una característica importante que es la de enviar y recibir información a través de Internet. Utiliza un hardware conocido como placa de desarrollo Arduino y un software para desarrollar código llamado Arduino IDE (Integrated

Development Environment). Por otro lado, la página del fabricante lo define como una plataforma de desarrollo libre que tiene incluido un microcontrolador reprogramable, cuenta con pines de fácil conexión con cables dupont (ver Figura 1b), su alimentación puede ser con cable USB o cargador. Existen varias versiones con características generales iguales, pero con particularidades como lo es la cantidad de pines de salida digitales o análogas y distintos tamaños, pero a pesar de las varias placas que existen, todas pertenecen a la misma familia (microcontroladores AVR marca Atmel), esto significa que comparten la mayoría de sus características de software, como arquitectura, librerías y documentación. Entre esas tarjetas se encuentra (Arduino Uno, Arduino Mega y Arduino Nano entre otros). Se recomienda de manera amplia el uso de esta tecnología porque es multiplataforma, su lenguaje de programación es de fácil comprensión, es de bajo costo y su consumo de energía es pequeño [11].

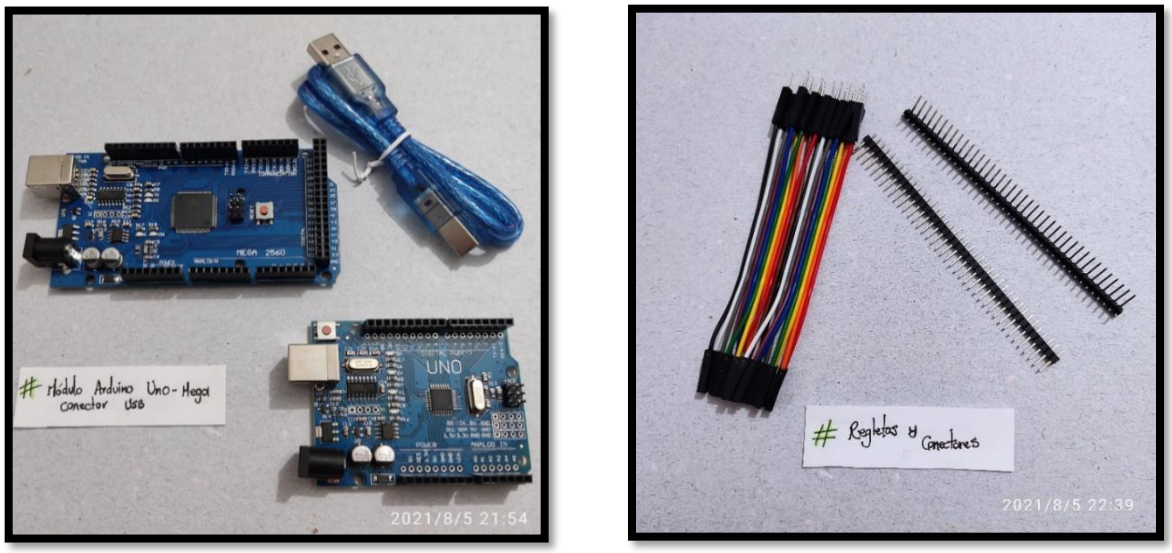


Figura 1. a) Placa Arduino y b) Cables tipo dupont o jumper
(Elaboración Propia)

Con base en la información consolidada, se selecciona esta placa (ver Figura 1a) para la realización del proyecto, por su lenguaje sencillo, tamaño adecuado, y salidas necesarias para manejar y/o controlar más de un sensor. Para trabajar con esta tarjeta programadora se adicionan otros componentes como los cables dupont (hembra-hembra, hembra-macho, o macho-macho), resistencias, sensores usados para (Técnicas de detección), cable de poder o tipo USB para poder energizarse. Arduino maneja una plataforma con un lenguaje de fácil entendimiento (ver Figura 2), al ser un software libre y que sus usuarios modifiquen su información (librerías) mediante sus necesidades, lo hace un sistema versátil, y de manejo sencillo.

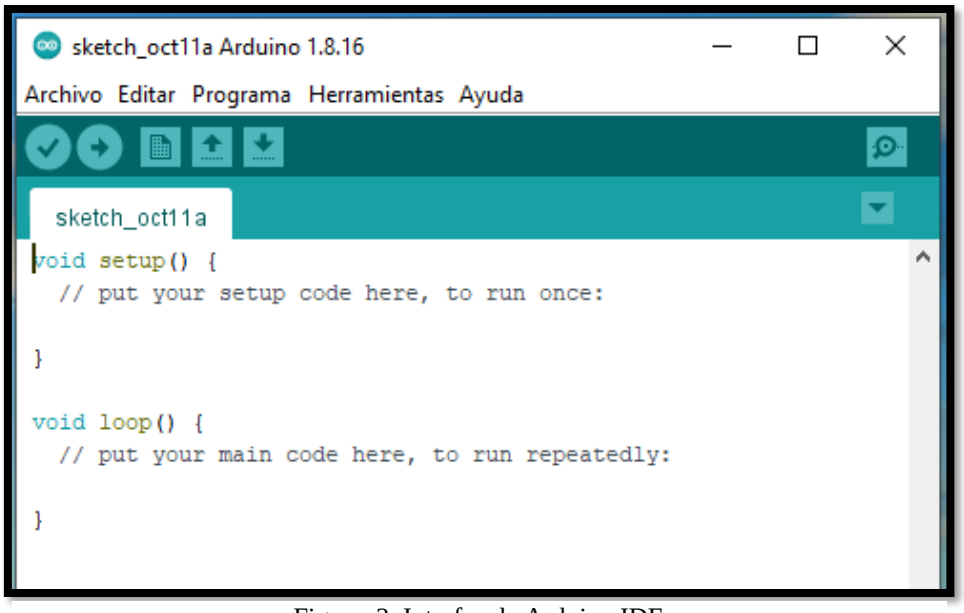


Figura 2. Interfaz de Arduino IDE
Elaboración Propia

3.3 Dron

Un dron por sus siglas en inglés UAV ("Unmanned Aerial Vehicle") o Vehículo aéreo no tripulado, es un dispositivo que no requiere un ocupante dentro de él, sino que vuela mediante programaciones o inteligencia artificial y se puede maniobrar a grandes distancias de manera totalmente autónoma siguiendo waypoints o puntos de referencia para realizar una tarea programada. En el documento «Joint Publication 1-02, Department of Defense Dictionary» editado por el Ministerio de Defensa de los Estados Unidos define UAV como:

«Un vehículo aéreo motorizado que no lleva a bordo a un operador humano, utiliza las fuerzas aerodinámicas para generar la sustentación, puede volar autónomamente o ser tripulado de forma remota, que puede ser fungible o recuperable, y que puede transportar una carga de pago letal o no. No se consideran UAV a los misiles balísticos o semibalísticos, misiles crucero y proyectiles de artillería».[12]

Existen drones para carreras, acuáticos, para realizar inspección, vigilancia, entretenimiento y en otras aplicaciones. El tipo más común es el cuadricóptero o de cuatro motores que pertenece a la categoría de ala rotatoria con despegue y aterrizaje vertical, en la figura 3 se visualiza una breve descripción según sus alas, su control y su uso y en la figura 4 se muestra un diagrama de sus componentes e interconexiones básicas.

Figura 3. Tipos de Drones [13]

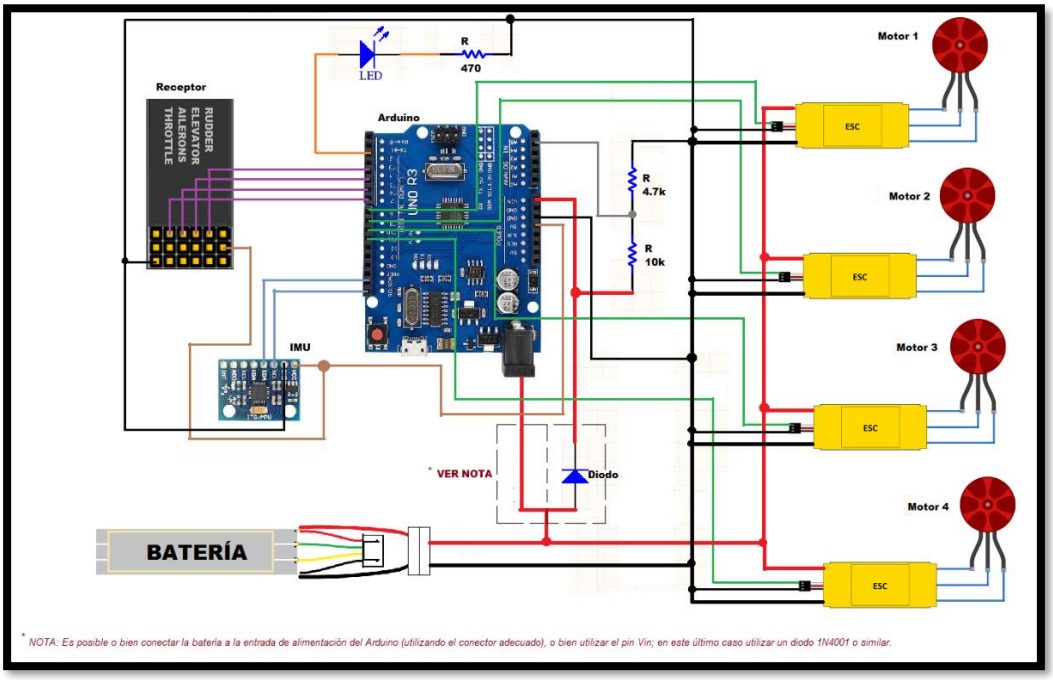


Figura 4. Las conexiones entre los componentes principales de un multirrotor [14]

3.3.1 Ala fija

Los drones de ala fija (ver Figura 5) son aeronaves que poseen un perfil alar que permite que la aeronave pueda moverse a través del aire y sea capaz de generar fuerzas sustentadoras para mantenerse en el aire. La principal característica de este tipo de drones es la gran autonomía que ofrecen ya que pueden estar volando varias horas gracias a su eficiencia aerodinámica. Los drones de ala fija son ideales para mapear grandes superficies de terreno ya que con una única batería se cubren grandes extensiones de espacio. Por este motivo son drones muy utilizados en trabajos de agricultura de precisión y de fotogrametría; a diferencia de los drones de ala rotatoria, con este tipo de drones no es posible realizar vuelos estacionarios, por lo tanto, no es adecuado realizar trabajos que requieran que el dron este volando fijo a una altura determinada como pueden ser, por ejemplo, los trabajos de inspección. Otra particularidad de este tipo de drones es que no pueden despegar ni aterrizar en vertical. Para el despegue de un dron de ala fija se necesita de una pista para carrera de despegue, una persona que se encargue de lanzarlo a mano o disponer directamente de una catapulta. La gran mayoría de los drones de ala fija actuales ya son capaces de realizar aterrizajes de forma autónoma, pero hay que tener en cuenta que se necesita una superficie lo suficientemente grande y en buen estado para que el dron no sufra ningún percance.[15], su diseño funciona adecuadamente para tomarlo como banco de prueba en tierra. En este proyecto el tipo que se usa es uno de ala fija, por comodidad, facilidad de ubicación de electrónica en su cuerpo y maniobrabilidad.

En los drones de ala fija tipo ala Zagi, la forma ‘v’ es la característica que más llama la atención a simple vista, ya que no tiene cola, esta forma es, por tanto, un factor de estabilidad, dando al modelo esa 'longitud' que tiene naturalmente un avión con fuselaje. Sin embargo, el ángulo de esta 'v' debe ser adecuado para la envergadura del modelo. Cuanto más "ancho" es el modelo, más tiende a ensancharse horizontalmente la "v". Entonces el modelo se vuelve más corto. La medida a la cual se refiere estas características se llama 'flecha'. Con respecto al perfil, sus medidas, diseños, y claves es más fácil remitirse a manuales en dónde se puede escoger con respecto a las cualidades aerodinámicas, el lift, drag, la envergadura, el ángulo de ataque con respecto a la dirección de vuelo, el borde de fuga por donde el aire pasa por toda el ala que se acoplen a las necesidades del proyecto; este tipo de dron cuenta con superficies primarias de control como los tienen los aviones, sin embargo, las más utilizadas y que varían de diseño son los alerones encargados de brindar el movimiento de alabeo y en algunos casos los winglets ubicados en la puntas del ala teniendo como principal objetivo disminuir la resistencia del aire.[16]

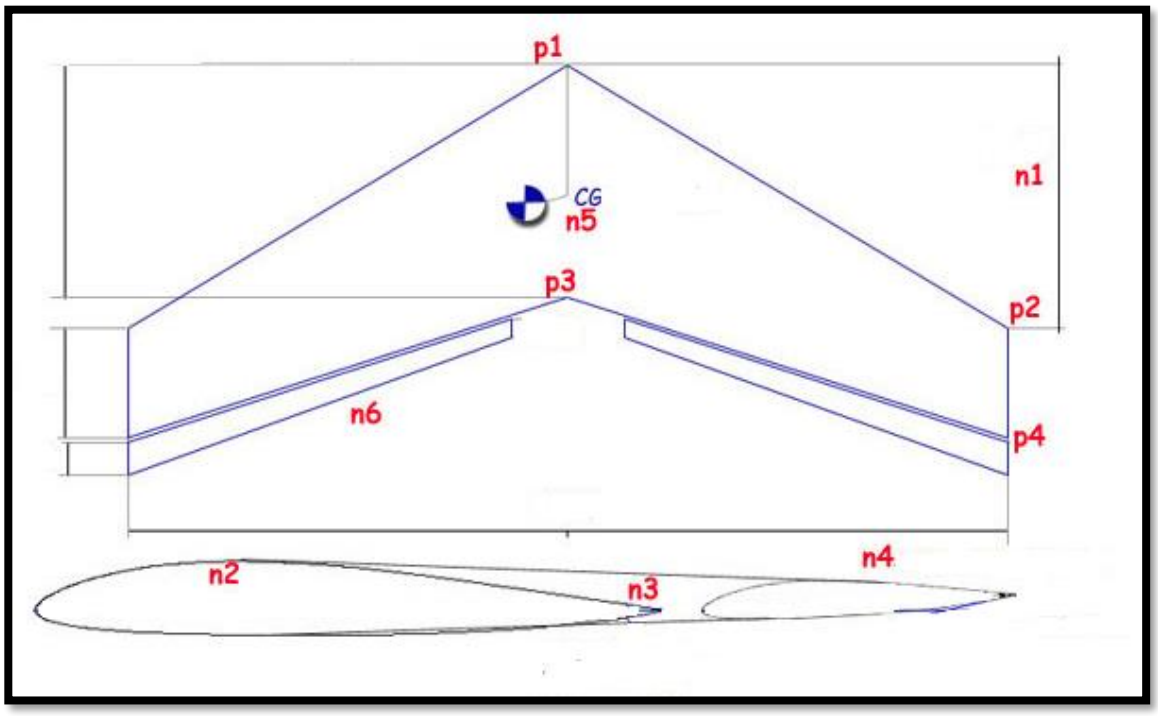


Figura 5. Ejemplo de plano de un ala Zagi[16]

En la Figura 5 se evidencia un ejemplo de plano que tienen cómo parámetros conocidos:

- N5= Centro de gravedad
- N6= Alerones
- N1= Cuerda aerodinámica
- P1-P3 = Cuerda máxima
- P2-P4= Cuerda mínima
- N3= Borde de salida
- N2= Sección vertical del ala

Para este proyecto se planteó un prototipo como el mostrado en la Figura 6, modificando el compartimento central donde irá ubicada toda la parte electrónica del sistema conservando el peso y balance del modelo original.

Figura 6. Dron ala fija [17]

3.4 Técnicas de detección

Por las coyunturas actuales de la sociedad el tema de seguridad cobra cada vez más importancia y se busca mejorar los métodos de vigilancia usando inteligencia y visión artificiales, enfocándose en la detección de individuos y objetos que puedan afectar la integridad de un ser, u objeto en particular, estas mejoras van ligadas a hardware y software que faciliten esta tarea y así lograr prevenir eventos dañinos. Este proceso es arduo y se requiere clasificar y analizar la información detectada con respecto a la necesidad que se plantee con ayuda de un sistema de visión artificial de toma de decisión y predicción automático, esto se implementa mediante de un sensor visual cómo lo son las cámaras de alta resolución [18].

Los diferentes tipos de sensores que se utilizan para la detección de obstáculos, embarcados en los vehículos autónomos (AV) deben realizar varias tareas de conducción autónoma, como el seguimiento de objetos, (los elementos dinámicos de la carretera (peatones, ciclistas, vehículos, esto supone un reto mayor debido a su ubicación y comportamiento continuamente cambiantes), la estimación de trayectorias y la evitación de colisiones, han sido las modificaciones que se han realizado en UAV [19].Para la acción de detectar obstáculos, se puede de diferentes maneras tipo sonoro, infrarrojo, visual etc.

3.4.1 LiDAR

LiDAR es un acrónimo que significa Laser Imaging Detection and Ranging, el cual permite medir distancias desde un emisor láser a un objeto o superficie. El funcionamiento de este sensor parte de la emisión y recepción de un haz de luz el cual se refleja en un espejo con una inclinación que cambia su trayectoria, este haz de luz viaja por el espacio hasta toparse con un objeto, cada haz determina la distancia que hay desde el sensor y el objeto a partir del tiempo de vuelo, esta tecnología LiDAR es utilizada para métodos no invasivos de mapeo de superficies, para obtener una digitalización rápida y precisa en formato nube de puntos. Mediante la ecuación 1 se describe su funcionamiento.

$$X = \frac{t\ x\ C}{2} \qquad \text{Ecuación 1}$$

Donde:

$X = Distancia\ entre\ el\ objeto\ y\ el\ sensor$
$t = Tiempo\ de\ vuelo\ Tof$
$C = Velocidad\ de\ la\ luz$

Estos sensores son precisos y fiables al momento de detectar un objeto gracias a que tiene una onda completa.[20]. Los principales usos de LiDAR se dan en:

- Arqueología: La capacidad de penetración del LiDAR a través de la vegetación ha permitido encontrar desde ciudades ocultas en la selva hasta yacimientos romanos.
- Hidrología: La alta resolución de los modelos digitales del terreno ayudan a crear mejores modelos predictivos de inundaciones, escorrentía, etc.
- Ordenación del territorio: Los datos LiDAR también se pueden usar para el planeamiento urbano ya que nos permiten obtener una "fotografía" tridimensional del momento de la captura de las ciudades.
- Forestal: Uno de los usos más extendidos del LiDAR es para el Inventario Forestal y la Ordenación de Montes.

En el caso del LiDAR aerotransportado, los dispositivos van equipados con un GPS que sirve para conocer en cada momento la posición del emisor y de este modo saber exactamente las coordenadas de cada retorno del láser con precisiones centimétricas. Sin embargo, el GPS no es suficiente para obtener una alta precisión sobre la posición de los datos. Además, es necesario un sistema llamado "IMU" de sus siglas en inglés "Inertial Measurement Unit" que se acopla al avión para registrar los giros, rotaciones e inclinaciones realizadas por el vehículo aéreo. Finalmente, muchos sistemas LiDAR, además permiten almacenar otros atributos necesarios para procesar los retornos de manera eficiente como por ejemplo el ángulo en el que el pulso fue "disparado" desde el emisor, el tiempo

GPS, número de vuelo y posición del retorno respecto al barrido [21].

3.4.2 Cámara

Es un dispositivo de captura y grabación de imágenes 2D, con ayuda de programación en lenguajes complejos o sencillos dependiendo el nivel del programador y la necesidad a suplir con ellas se logra obtener la información requerida como una captura y reconocimiento de rostros, de objetos, o grabación en tiempo real. Las cámaras pueden ser análogas o digitales, la principal diferencia entre ambas tecnologías es la forma en el cual el video es transmitido, y forma en que la información se almacena y comprime. Según [11], algunas características a tener en cuenta para seleccionar la tecnología adecuada son:

- Calidad de video
- Confiabilidad
- Precisión
- Latencia
- Cableado
- Seguridad
- Costos

La detección de objetos mediante imágenes puede ser una tarea compleja porque principalmente cuando la imagen no está limitada a tener un solo objeto, sino que puede contener múltiples objetos. La tarea es clasificar y localizar todos los objetos en la imagen. Aquí nuevamente la localización se realiza utilizando el concepto de cuadro delimitador [22].

3.4.3 Infrarrojo

Los sensores infrarrojos son una tecnología que inició en los años 90s. Siendo su principio de funcionalidad la detección de la radiación emitida por los cuerpos vivos e inertes, que posteriormente es transformada en una señal eléctrica. El sensor infrarrojo requiere de una comunicación lineal entre transmisor y receptor, lo que hace impredecible la línea de vista para su efectiva transmisión, por lo tanto, siempre será uno a uno, dejando de lado las configuraciones multipunto. Estos dispositivos están diseñados especialmente para la detección, clasificación y posicionado de objetos; la detección de formas y diferencias de superficie, incluso bajo condiciones ambientales extremas. Este componente puede tener la apariencia de un LED normal como se aprecia en la figura 7; la diferencia radica en que la luz emitida por él no es visible para el ojo humano, únicamente puede ser percibida por otros dispositivos electrónicos. Adicionalmente, el uso de estos sensores se puede evidenciar en aparatos tales como: aire acondicionado, drones, o en la apertura de puertas.

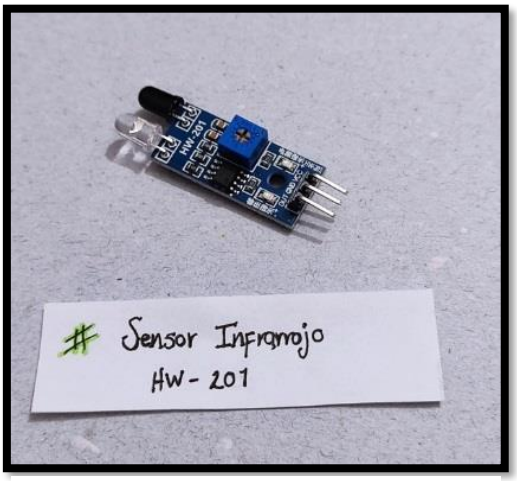


Figura 7. Sensor Infrarrojo HW – 201
Elaboración Propia

Es importante precisar que estos sensores emiten un haz de luz el cual tiene una abertura angular, utilizando este principio de funcionamiento para mantener la posición del objeto de acuerdo a la intensidad de la luz que reciba [23].Esta tecnología también se usa para detectar la distancia en la que se encuentra el objeto a medir dentro de un robot móvil, afirmando lo mencionado anteriormente.

3.5 Modulación PWM

Modulación por ancho de pulso o por sus siglas en inglés Pulse Width Modulation (PWM). Es una señal de onda cuadrada que se repite a una determinada frecuencia. Cada ciclo se denomina período, y el porcentaje de tiempo que la señal PWM está encendida durante un período específico determina el ciclo de trabajo, podemos ajustar una tensión continua constante a diferentes niveles de tensión. Esto nos ayuda a controlar un motor a distintas velocidades.

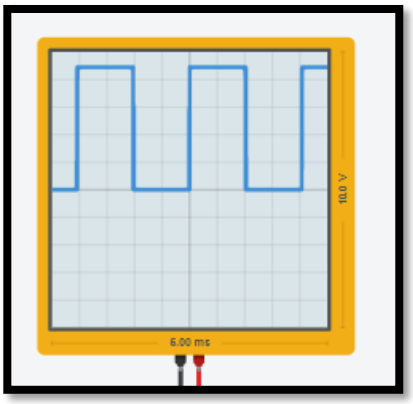


Figura 8. Gráfica de PWM [24]

Otro parámetro característico de una señal PWM es el denominado ciclo de trabajo (duty cycle), que hace referencia al porcentaje de tiempo que el pulso (tensión entregada) está en activo durante un ciclo. [25]

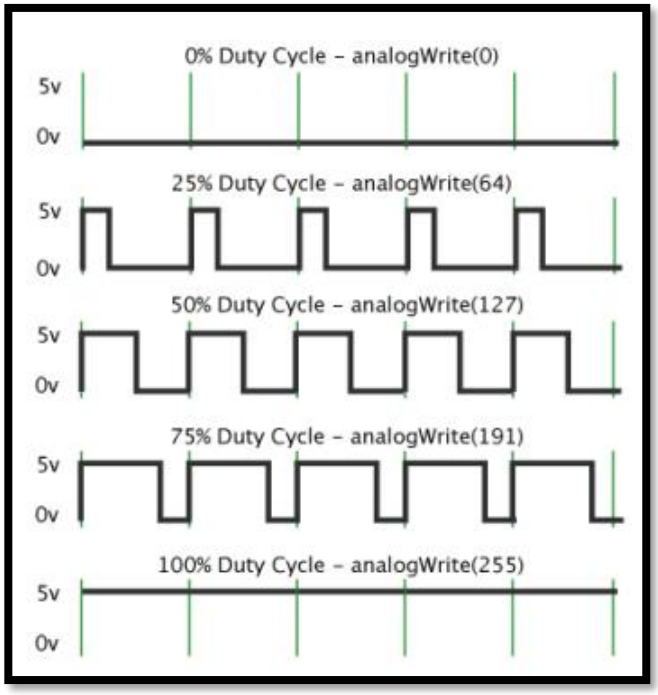


Figura 9. Gráfica de Duty cycle (Modulación PWM) [25]

Capítulo 4

Metodología

4.1 Tipo de investigación

Este proyecto tiene un enfoque de investigación cualitativo de tipo exploratorio teniendo en cuenta y siguiendo como ejemplo lo establecido por Elsy Bonilla y Penélope Rodríguez en el libro “Más allá del dilema de los métodos”[26], donde lo que se buscó fue especificar mediante la exploración y la evaluación, cómo a partir de la implementación de un sistema de detección y evasión de obstáculos de bajo costo para drones de ala fija, se puede innovar con el manejo integral de sensores, mejorando así la funcionalidad de los drones con prueba en tierra. Es exploratoria porque no se tiene conocimiento que dentro de los proyectos presentados en la Fundación Universitaria Los Libertadores se hayan realizado trabajos investigativos que den cuenta de la incorporación de diferentes tipos de sensores (LiDAR, infrarrojo, cámara) en un solo prototipo. Es evaluativa, debido a la incorporación de diferentes tipos de sensores (LiDAR, infrarrojo, cámara) obliga a realizar pruebas ilimitadas buscando la efectividad del propósito del presente proyecto, en el diagrama de flujo de la Figura 10 se muestra la metodología desarrollada.

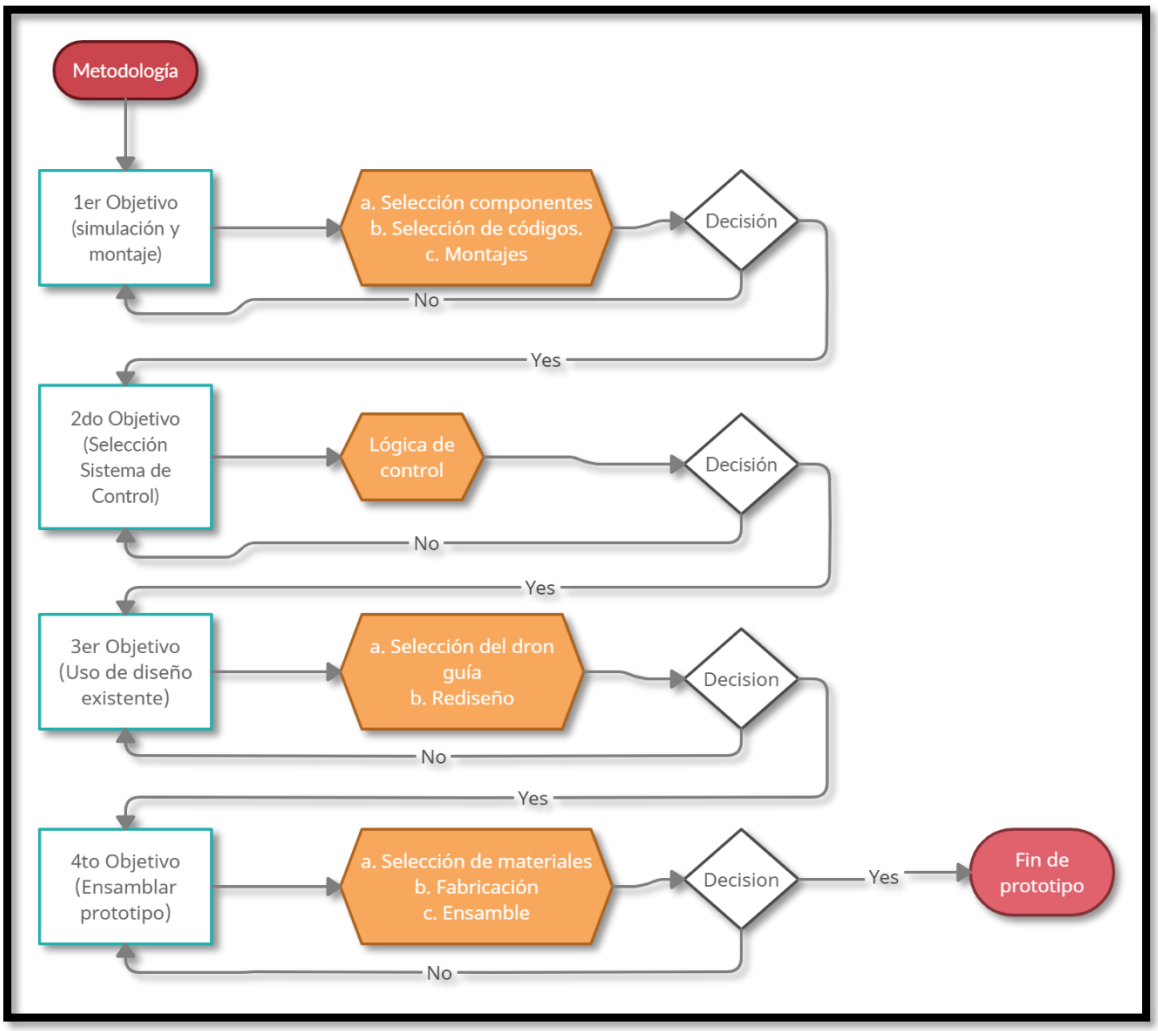


Figura 10. Diagrama de flujo de la metodología. *Elaboración propia*

4.2 Diseño Preliminar

Durante esta fase se definen los requisitos que puede tener el usuario al momento de interactuar con el prototipo del ala Zagi en un banco para pruebas en tierra, tales como, programación, detección de obstáculos, riesgos inherentes a la operación, y demás consideraciones que a nivel de ingeniería se requieran con el fin de cumplir las exigencias y objetivos planteados en este proyecto.

4.2.1 Misión

La principal tarea del dron es ejecutar una adecuada detección de objetos, para lograr esto se debe establecer el rango que ofrece cada sensor (LiDAR, infrarrojo y cámara) y que, mediante la información captada por ellos, se realice el movimiento de las superficies de control (alerones) y del

motor brushless, que simularan las maniobras de evasión ante un objeto detectado.

4.2.2 Requisitos

Los requerimientos que se establecieron para la funcionalidad del dron se determinaron mediante una tabla de verdad (ver tablas 1 a 4), tomando como referente dos entradas provenientes de la información de los sensores (LiDAR, infrarrojo) con sus respectivas salidas, gobernadas y /o controladas por dos servos y un motor brushless). En la tabla 1 se evidencia las entradas del sistema de control establecidas por la cantidad de sensores en este caso dos (LiDAR e infrarrojo), generando cuatro combinaciones posibles donde por medio de números binarios (1; 0) se determina la lógica de detección de la siguiente manera: sensor detecta = 1, sensor no detecta = 0. En la Tabla 2 se evidencian las salidas generadas por las diferentes combinaciones de las entradas establecidas en la Tabla 1, a lo cual ejecutará la acción indicada para los motores (motor brushless y servomotores) donde motor activo = 1, motor inactivo =0.

Tabla 1. Tabla lógica de verdad con los sensores como entrada. *Elaboración propia*

Tabla de verdad		
Combinaciones	**Entrada 1 (Sensor Infrarrojo Corto alcance)**	**Entrada 2 (Sensor LiDAR (Largo alcance)**
1	0	0
2	0	1
3	1	0
4	1	1

Tabla 2. Salidas (Motores Brushless y servos). *Elaboración Propia*

Salida 1 **Motor Bruslehh**	**Salida 2** **Servo Aleron 1 (Derecho)**	**Salida 3** **Servo Aleron 2 (Izquierdo)**
1	1	1
1	1	1
1	1	1
0	1	1

Tabla 3. Dirección de los alerones. *Elaboración Propia*

Dirección de los Alerones
Alerones abajo (Pitch arriba)
Aleron derecho arriba - aleron izquierdo abajo (Alabeo izquierda)
Aleron derecho abajo - aleron izquierdo arriba (Alabeo derecha)
Alerones arriba (Pitch abajo)

Tabla 4. Posibles escenarios. *Elaboración Propia*

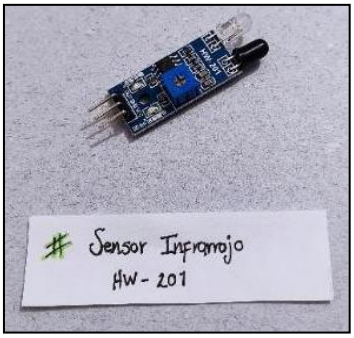
Escenario
Avión en vuelo recto y nivelado sin obstáculos (50 rev)
Obstáculo muy cercano, alabeo y cambio de rpm de motor para evadir (65 rev)
Obstáculo muy cercano, alabeo y cambio de rpm de motor para evadir (55 rev)
Apague sistema de propulsión alerta (30 rev)

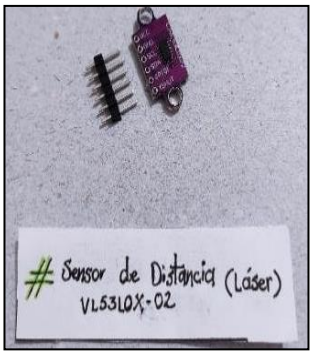
Con base en las tablas de la 1 a la 4, que representan los parámetros de funcionalidad del dron establecidos para cumplir la misión, se hace un análisis en conjunto, con simulación, montaje y selección de un diseño de dron que se adapte a la electrónica a implementar (espacio y/o ubicación de los sensores y actuadores.

4.3 Descripción y selección de los sistemas de detección y componentes electrónicos.

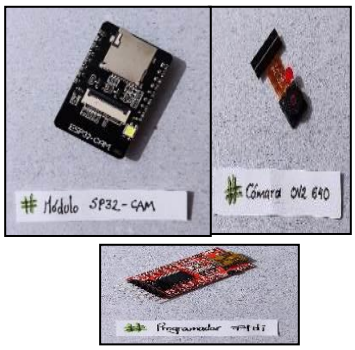
En la tabla 5 se describen los materiales y componentes a usar para el desarrollo del dron, para ello, se evalúan materiales resistentes, livianos, de fácil manipulación y económicos.

Tabla 5. Descripción de componentes del proyecto. *Elaboración propia*

ITEM	**NOMBRE COMPONENTE**	**FIGURA**	**DESCRIPCION**
1	Sensor Infrarrojo HW-201		Dispositivos opto electrónicos capaces de medir la radiación electromagnética infrarroja de los cuerpos en su campo de visión [27]

ITEM	NOMBRE COMPONENTE	FIGURA	DESCRIPCION
2	Sensor de distancia Láser VL53L0X-02		Este tipo de sensor detecta la posición del objeto. Esto se logra mediante el uso de un sistema de triangulación o uno de medición de tiempo. Es parte de la nueva generación de **sensores de distancia** por tiempo de vuelo (ToF: Time of Flight). Posee un empaque ultra pequeño, ofrece mediciones exactas sin importar la superficie reflectante, con un rango de medición de hasta 2m. [28]
3	Módulo Arduino UNO		El Arduino es una placa que tiene todos los elementos necesarios para conectar periféricos a las entradas y salidas de un microcontrolador. [29]
4	Módulo Arduino Mega		Un microcontrolador con la configuración de “sistema mínimo” (El término “sistema mínimo” se refiere a que solo se utilizan los componentes indispensables para el microcontrolador). [30]

ITEM	NOMBRE COMPONENTE	FIGURA	DESCRIPCION
5	Motor Brushless EMAX XA2212		Es un motor eléctrico que no emplea escobillas para realizar el cambio de polaridad en el rotor.[31]
6	Controlador de Velocidad ESC EMAX 30A		Está diseñado para controlar motor brushless en drones, haciendo que el control sea muy fácil a través de una señal de PWM, este módulo además puede proveer 5V, para energizar tu sistema de control con hasta dos amperios. [32]
7	Batería Lipo Turnigy 2200 mAh		Esta batería LiPo (polímero de iones de litio) de descarga alta es una excelente manera de alimentar cualquier proyecto RC, robótico o portátil. Es una excelente opción para cualquier cosa que requiera una batería pequeña con mucha fuerza [33]
8	Cargador de Batería Lipo		Carga y balancea con precisión baterías de polímero de lito, de LiFe (A123), de NiCd y de NiMH, con el cual puedes cargar hasta 6s (LiPo/LiFe) muestreando el voltaje de cada celda en tiempo real. [34]

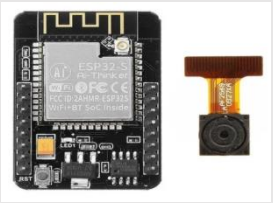
ITEM	NOMBRE COMPONENTE	FIGURA	DESCRIPCION
9	Cámara SP32 CAM		Plataforma ideal para transmitir vídeo e imágenes por internet de forma sencilla y en diseño compacto.[35]
10	Servomotor sg90		Es un servo miniatura de gran calidad y diminutas dimensiones, además es bastante económico. Funciona con la mayoría de las tarjetas electrónicas de control con microcontroladores y además con la mayoría de los sistemas de radio control comerciales. [36]

Los componentes para implementar son acordes y coherentes con el objetivo general del proyecto, por tamaño, peso, prestaciones técnicas y precio; así mismo seguir encaminados al logro de cumplir los objetivos propuestos. Para ello fue realizada una búsqueda de sensores de bajo costo que fueran compatibles con Arduino, debido a la variedad en el mercado en las tablas 6,7 y 8 fue realizada una comparación de los componentes (cámara, LiDAR e infrarrojo), que facilitara la elección de estos.

4.3.1 Cámaras y sensores

Son usadas para la visualización, se determina que la cámara más pertinente para utilizar es la cámara ESP32 CAM, para visualizar objetos y hay varias disponibles para usar con Arduino, se pueden conectar tanto vía bluetooth o wifi, el ángulo de visión es mayor en comparación con la OV7670, al igual que la resolución de la imagen; la tarjeta al tener salidas digitales se le pueden agregar más dispositivos si se cree necesario, adicional a esto al tener memoria se puede guardar la información capturada. En la tabla 6 se realiza una comparación de dos cámaras.

Tabla 6. Comparación de cámaras compatibles con Arduino. *Elaboración propia*

CÁMARAS PARA ARDUINO				
Características	**ESP 32 CAM OV2640**		**OV7670**	
Componente		**favorabilidad**		**favorabilidad**
Voltaje de alimentación	5v	+	3.3v	−
Módulo wifi/bluetooth	802.11b/g/n - 4.2BR	+	N/A	−
Resolución	1600X1200 (imagen) 1080p30, 720p60, 680X480p90 (video)	+	640X480 Imagen video no especifica	−
Flash	Si	+	N/A	−
SRAM	520 kb	+	N/A	−
Interfaz de Control	UART / SPI / I2C /PMW / ADC /DAC	+	SCCB / I2C	−
	30 - 60 fps	+	30 fps	−
Pines	16	−	18	+
Ángulo de visión	120° - 160°	+	25°	−
Peso	20g	−	10g	+
Dimensiones	27 x 40.5 x 6 mm	−	34 × 25 × 35 mm	+
Temperatura	-40 - 85°C	+	-30 - 70°C	−
Precio	$ 42.000,00	−	$ 25.000,00	+

Los sensores son usados para detección de obstáculos, en este proyecto se usaron de dos tipos LIDAR e infrarrojo y en las tablas 7 y 8 fue realizada una comparación de dos tipos de cada uno de estos sensores, con el fin de seleccionar el más adecuado a las necesidades del proyecto.

Tabla 7. Comparación de sensores Infrarrojo. *Elaboración propia*

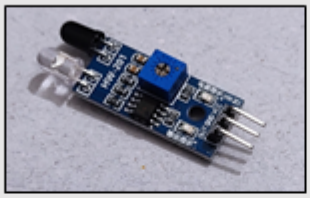
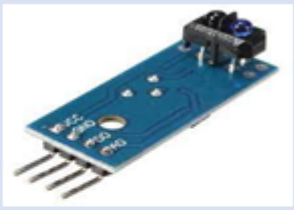
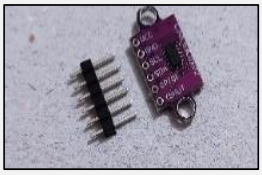
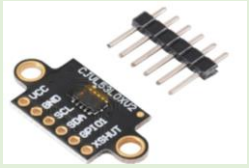
SENSORES INFRARROJOS				
Características	**Hw-201**		**Tcrt5000**	
Componente		**Favorabilidad**		**Favorabilidad**
Voltaje de alimentación	3.3 a 5 volts	+	5V	−
Voltaje de salida digital	5V	+	1.25 V	−
Dimensiones	31 mm x 15 mm x 7 mm	+	37 mm x 14 mm	−
Distancia de detección	20 a 300 mm	−	0,3 m - 12m	+
Angulo de detección	35°	+	1,5 °	−
Conexiones	VCC, OUT, GND		VCC, GND, DO, AO.	
Precio	$ 8.000,00	+	$ 20.000,00	−

Tabla 8. Comparación Sensor LiDAR. *Elaboración propia*

SENSORES LIDAR						
Características	**VL53L0X-02**		**LiDAR TF-Luna**		**VL53L0X**	
Componente		**Favorabilidad**		**Favorabilidad**		**Favorabilidad**
Voltaje de alimentación	2.6 ~ 5.5v	+	5v	−	3 ~ 5 V	−
Distancia:	2m	+	0.2-8m	−	2m	+
Dimensiones	4,4 x 2,4 x 1,0 mm	−	35mm*21.25mm*13.5mm	+	25mm x 10.3mm x 3.5mm	+
Angulo de medición:	25 °	+	2 °	−	25 °	+
Temperatura de operac	-20°C a 70°C	+	-10℃~60℃	−	-10°C a 70°C	−
Pines	VCC, GND, SDA, SCL, GPI01, XSHUT	+	ABS+PC	−	VCC, GND, SDA, SCL, GPI01, XSHUT	+
peso	1.2 g	+	＜5g	+	1.4 g	+
Emisor Laser Infrarrojo	940 nm	+	850nm	−	940 nm	+
Tipo de laser	VCSEL	+	VCSEL	+	VCSEL	+
Precio	$ 24.000,00	+	$ 97.110,00	−	$ 44.000,00	−

Es importante tener en cuenta los componentes adecuados para el ensamble del dron. El proyecto se basa en las conexiones de la Figura 4, la cual muestra las partes básicas y necesarias para armar un dron como lo son: i) La estructura (frame ala Zagi), ii) Controladora de vuelo (placa Arduino, ESC), iii) la propulsión, que son los motores (servomotores y motor brushless) y iv) el sistema de detección (cámara, LiDAR, infrarrojo).

4.3.2 Controladora de vuelo

La controladora de vuelo se encarga de generar señales de control hacia los motores con base en la información leída de los sensores. En este caso el Arduino hace las veces de controladora de vuelo. Para el proyecto se tienen dos opciones Arduino Uno y Arduino Mega (ver figura 11a), ambas tarjetas son funcionales, sin embargo, se elige comparando las prestaciones de cada una principalmente tamaño, cantidad de salidas análogas y digitales.

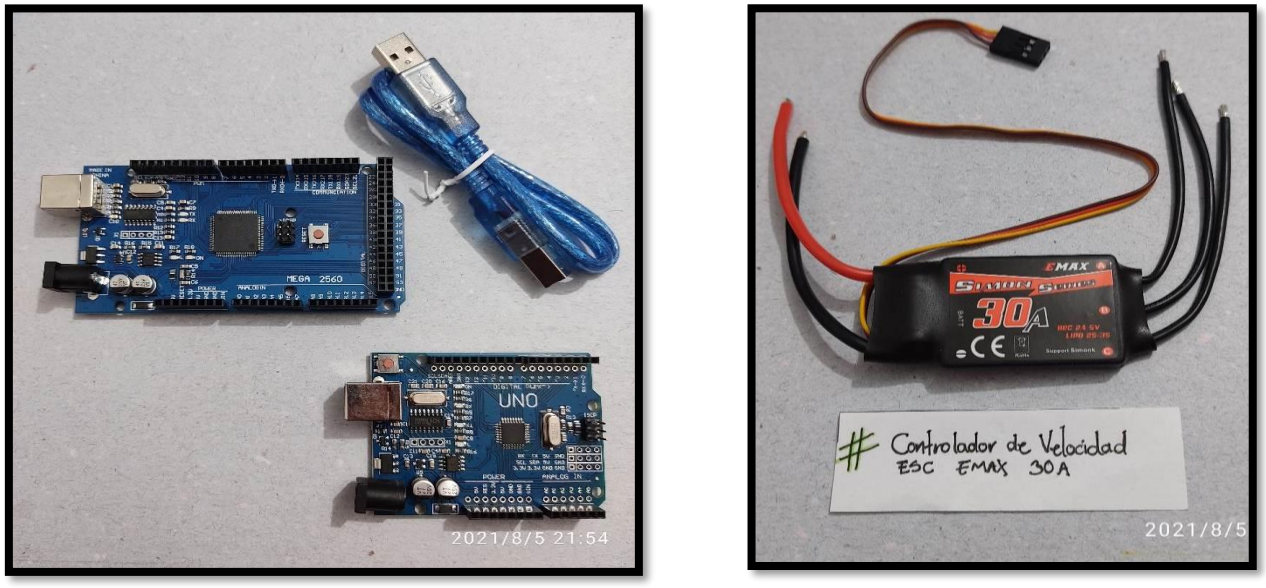


Figura 11. a) Tarjeta Programable y b) variador de velocidad
Elaboración Propia

4.3.3 Controlador electrónico de velocidad

Los controladores electrónicos de velocidad ESC se encargan de gestionar la corriente eléctrica para variar las RPM del motor brushless, este dispositivo de control se alimenta por medio de una batería Lipo, en el mercado se encuentra un sinfín de opciones, desde la marca, fabricante, diseño, corriente y celdas soportadas. Se eligió la ESC mostrada en la figura 11b que tiene una capacidad de gestionar 30 A de corriente, soporta una carga de voltaje (batería) de 2 a 3 Celdas, acorde al motor, sus pines de conexión se adaptan a los que se conectan a la batería, se suelda un conector T60 macho, los que se

conectan al motor se colocan pines de contacto macho y los pines de control son compatibles con los cables dupont para el Arduino.

4.3.4 Batería Lipo

Para energizar los motores y el proporcionar energía eléctrica al dron se usa una batería LiPo (Polímero de Litio) compuesta por tres celdas, cada una de ellas debe contener un ánodo, un cátodo y un electrolítico. El ánodo es una ultradelgada lámina de litio metálico que hace la función de fuente de iones de litio (descarga) o como colector (carga), el cátodo es un material compuesto con capas intercaladas de óxido de vanadio, sal de litio y polímeros, todo ello laminado sobre una hoja de aluminio que sirve de colector. El aspecto que hace únicas a las baterías de polímero de litio es el electrolito confeccionado a partir de una membrana que sirve de separador entre las láminas de ánodo y cátodo, es un sólido de textura gomosa, que puede estar constituido por diversos compuestos según fabricante. La actual tendencia a la hora de diseñar baterías de ion de litio, consiste en añadir al electrolito un gel que mejora las propiedades térmicas y disminuye la impedancia interna de la batería, mejorando su capacidad de descarga, el comportamiento elástico del polímero permite que la superficie de contacto con los electrodos sea la adecuada [37]. Fue seleccionada una batería de polímero de litio mostrada en la figura 10a porque las baterías convencionales no entregan la corriente suficiente para hacer mover el motor. Las principales características de esta batería son que tiene tres celdas, una capacidad de 2200 mAh y un voltaje de 11.1v, con una tasa de descarga de 35C, esto significa que es la rapidez con la que una batería se poder descargar de forma segura. Con el dispositivo de carga mostrado en la Figura 10b, se realiza un proceso de carga de forma balanceada para la batería, optimizando el suministro de corriente y voltaje adecuado para ella. Es un equipo de carga multipropósito ya que dispone de varios conectores dando la posibilidad de cargar diferentes tipos de batería.

Ecuaciones para calcular el tiempo de descarga de la batería teniendo en cuenta las características de la misma.[38]

Batería Lipo Turnigy

- 2200 mAh
- 3cell
- 11.1 v
- 35C

$$CxmAh = mAh\ de\ descarga \quad \text{Ecuación 2}$$

$$^{mAh}/_{1\ min} = mAmin \quad \text{Ecuación 3}$$

$$mAmin\ x\ C = mAmin \quad \text{Ecuación 4}$$

$$^{mAmin}/_{mAhBat} = \min de\ descarga \quad \text{Ecuación 5}$$

$$35Cx2200mAh = 77000\ mAh\ de\ descarga$$

$$2200\ {}^{mAh}/_{60} = 36.66\ mAmin$$

$$36.66\ mAmin\ x\ 35C = 1283.33\ mAmin$$

$$1283.33\ {}^{mAmin}/_{2200\ mAhBat} = 5.83\ \text{min}\ de\ descarga$$

La batería tendrá un tiempo de descarga de 5.83 min.

Figura 12. a) Batería Lipo y b) cargador de batería
Elaboración propia

Es importante realizar medición de voltaje por cada celda y en conjunto (con las 3 celdas conectadas) antes y después del proceso de carga como se muestra en la Figura 13, determinando el estado de la batería y si es o no necesario ponerla a cargar.

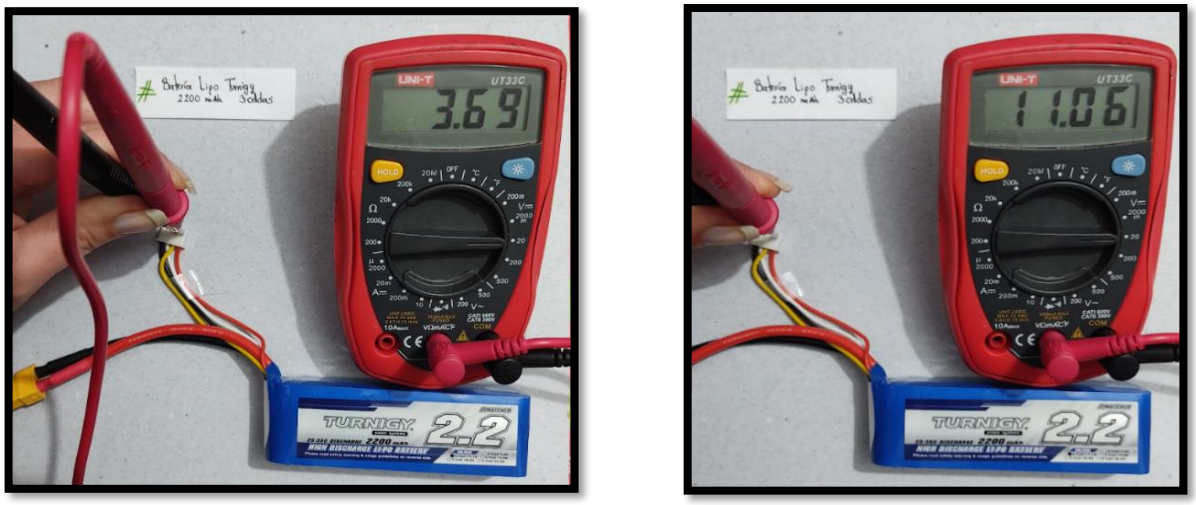


Figura 13. Voltaje de una celda (3.69V) y Voltaje de las celdas (11.06 apróx 11.1 V)
Elaboración propia

De acuerdo con las pruebas realizadas e ilustradas en la figura 13, y mediciones se determina que la batería se encuentra en buen estado y lista para usarla en los montajes del circuito. El fabricante indica que, para prolongar su vida útil, se debe cargar en su totalidad, sin embargo, es mejor no permitir que su curva de descarga supere los 3 voltios, (9.1v), adicional se sugiere guardarla en empaques herméticos y con protección contra incendios.

4.3.5 Motores para propulsión y control de superficies

Para dar propulsión a este tipo de drones se usan motores Brushless, que significa sin escobillas, son los elementos mecánicos utilizados en la conmutación de los motores de corriente directa tradicionales y se van desgastando con el uso [39]. Para este tipo de trabajos los que no tienen escobillas son más fiables, más potentes y tienen más protección, por ello el seleccionado es el EMAX 2212 de 1400 KV mostrado en la figura 14a, que tiene las siguientes características: buen diseño de succión siendo esta succión fuerte, la temperatura de funcionamiento del motor es baja, buen rendimiento soporta baterías 2-3S, tiene una larga vida útil y gran potencia.[40]

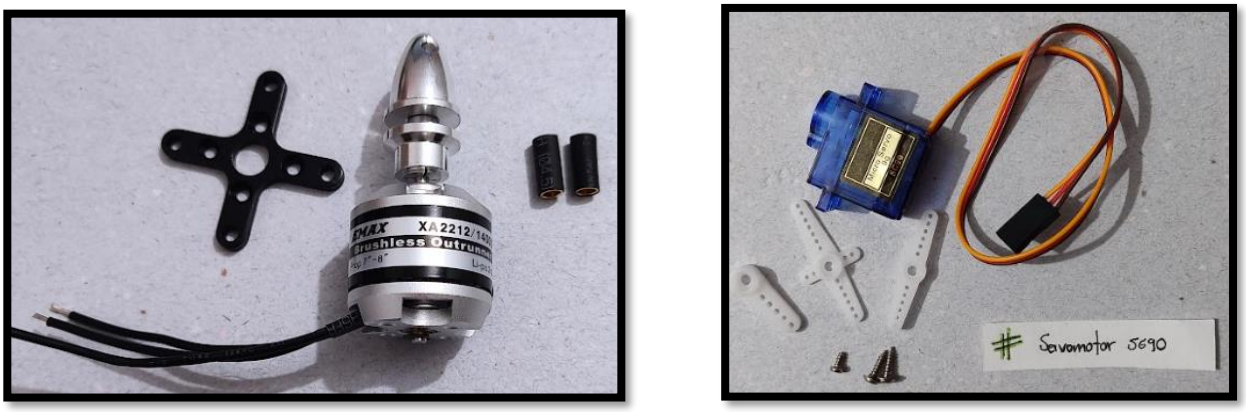


Figura 14. a) Motor Brushless y b) Servomotor SG-90
Elaboración propia

Con respecto al motor brushless se realiza una simulación para validar la señal PWM generada por el Arduino, ya que las salidas de modulación de ancho de pulso van desde 0 a 5V generando un pulso con un tiempo determinado y así mismo con ayuda del controlador de velocidad se hace una amplificación de señal para mover el motor adecuadamente.

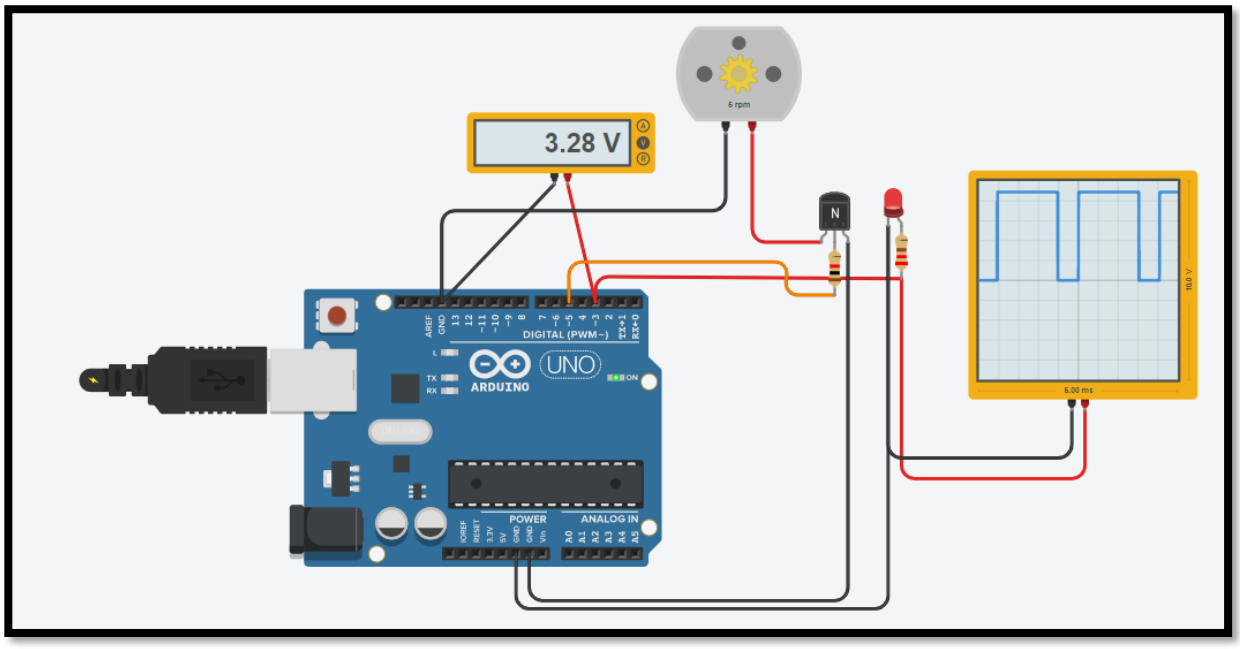


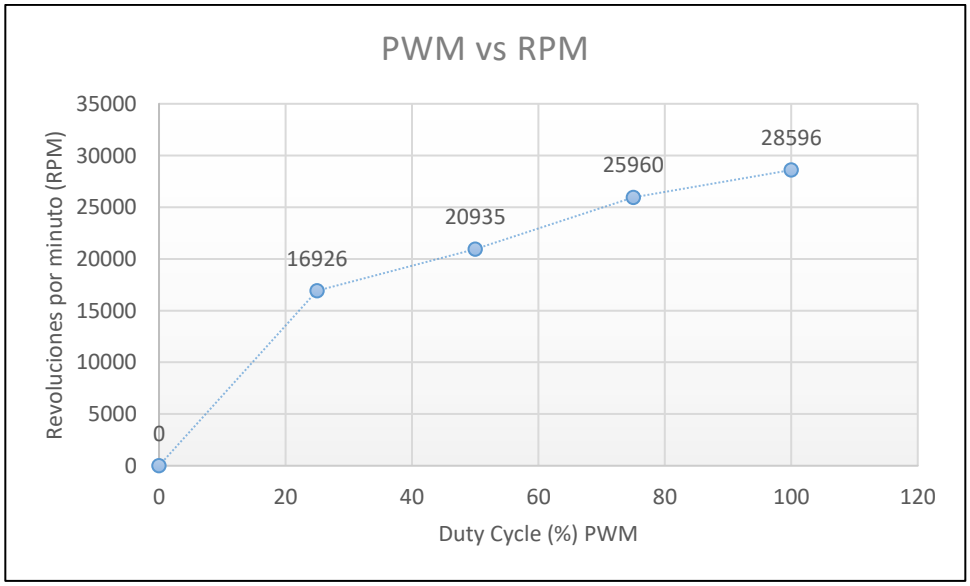
Figura 15. Simulación en TinkerCAD para obtener la señal PWM. [24]

De igual manera se hace el montaje (ver figura 16) para validar las RPM y las PWM del motor brushless seleccionado obteniendo así la gráfica 1, mediante la siguiente tabla, los porcentajes se establecieron teniendo en cuenta la Figura 7 siendo el 0% 0.5 ms y el 100% 2.5 ms como ancho de

pulso predeterminados por el fabricante.

Tabla 9. Pruebas de motor brushless en banco de pruebas. *Elaboración propia*

Corriente (A)	Duty Cycle (%) PWM	RPM	KV	Voltaje (v)
0	0	0	0	12,03
1,19	25	16926	1469	11,44
2,24	50	20935	1921	10,86
4,26	75	25960	2378	10,13
6,81	100	28596	2808	9,39



Gráfica 1 Modulación por ancho de pulso con respecto a las revoluciones por minuto. *Elaboración propia*

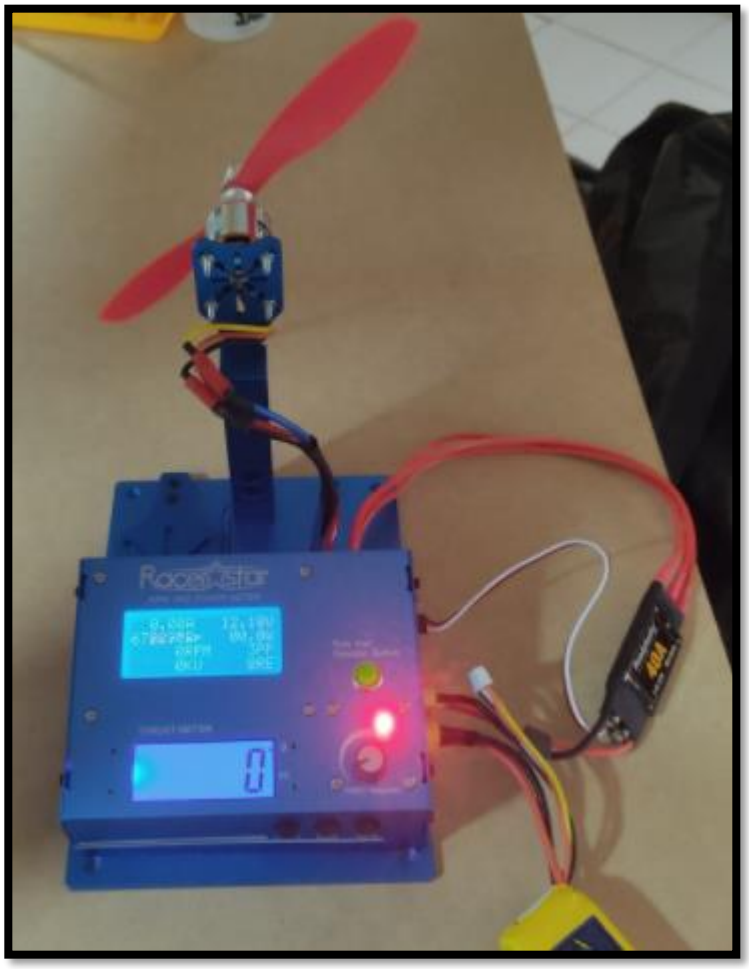


Figura 16. Montaje para obtener la señal PWM.
Elaboración propia

Nota: Como se visualiza en la gráfica se puede determinar que el ciclo de trabajo del motor brushless a mayor porcentaje las RPM también aumentan, sin embargo, no son del todo lineal, al igual al aumentar la corriente sucede lo mismo con los RPM y los KV, pero son inversos al consumo de voltaje siendo este menor al aumentar la velocidad como se evidencia en la Tabla 9.

Para el control de las superficies primarias de control (alerones) se utilizan servomotores (ver figura 14b), estos motores tienen un giro controlado del eje y requieren de una señal que consiste en una serie de pulsos, siendo esta la duración que indica el ángulo de giro del servo, a lo cual cada uno tiene sus márgenes de operación que corresponden al ancho del pulso ya sea máximo o mínimo. Los tiempos de duración del pulso es de 0.5 ms a 2.5 ms, que corresponden a la posición de los dos extremos del servo. De acuerdo con la información recolectada el servo escogido es el SG-90 siendo de fácil conexión, tamaño, peso y costo razonable. Tiene ventajas apreciables como: par elevado, fiabilidad de

funcionamiento, bajo mantenimiento, gran exactitud en el control de velocidad y posición, capacidad de velocidades muy altas, pérdidas en el rotor muy bajas, rotor con poca inercia, construcción cerrada, es útil para trabajar en ambientes sucios y presenta amplia gama de potencias (de 100 W a 300 KW).

4.3.6 Hélices

Para generar sustentación se usan hélices de plástico como las mostradas en la figura 17, esto como medio de propulsión aerodinámico fundamental que brinda empuje al dispositivo basándose en las revoluciones del motor, es decir, los motores brushless tienen una característica especial que son los KV (número de revoluciones por voltio) y a partir de esa medida se selecciona el tamaño adecuado de las hélices, en este caso que el motor proporciona 1400 KV y el fabricante sugiere usar una hélice de tamaño de 6” a 7” de largo, datos extraídos del datasheet del fabricante.

Figura 17. Hélices
Elaboración propia

4.4 Pruebas y validación de códigos en Arduino para Servomotores

La funcionalidad de los servomotores fue validada con el Arduino, mediante códigos simulados en Tinkercad como se observa en la Figura 18a. Inicialmente se buscan códigos para gobernar los servos y simular su movimiento antes de realizar el montaje. Esta validación consta de dos pasos: i) Modificación primaria al código Arduino [41] para comprobar comportamiento del servomotor y ii) Simulando en el programa de libre acceso “TinkerCad” siendo una herramienta gráfica e intuitiva, se realizó el montaje de los servomotores para verificar el funcionamiento. Una vez validada la simulación se procede hacer el montaje para comprobar el funcionamiento como se muestra en la figura 18b teniendo en cuenta las conexiones hechas tanto en la simulación como lo programado en el código mostrado en la figura 19.

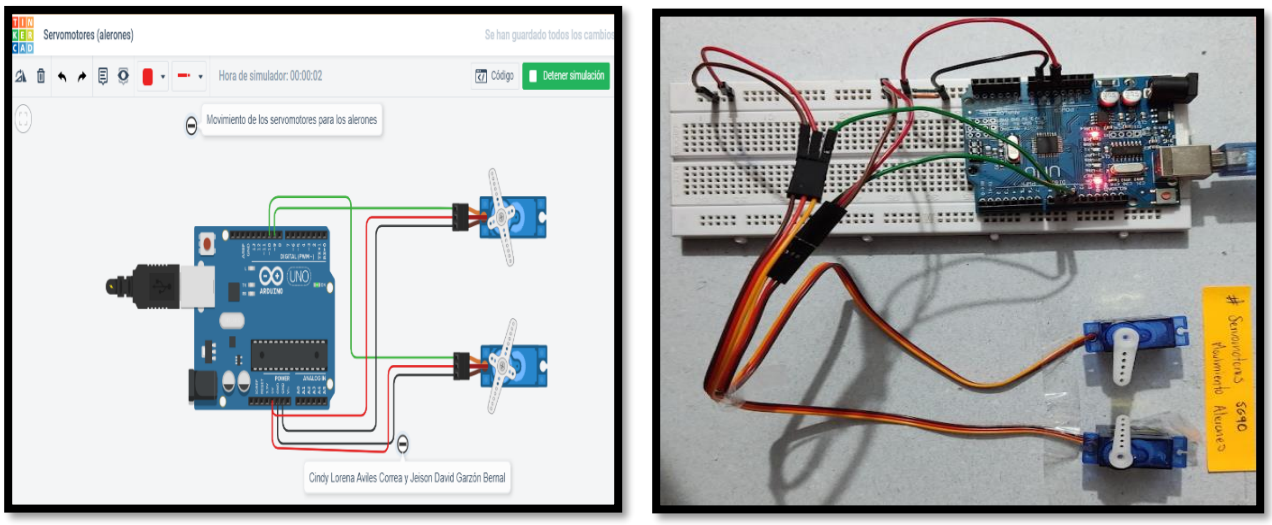


Figura 18. a) Simulación Servomotores en TinkerCad y b) Montaje servomotor para alerones
Elaboración propia

Servos-alero Arduino 1.8.15

Archivo Editar Programa Herramientas Ayuda

Servos-alero

```
#include <Servo.h> // Se incluye librería para el control de los servomotores


// Se indican las variables
Servo aleron1;
Servo aleron2;

void setup()
// Se indica la posición de los servos con las salidas PWM del arduino
{
  aleron1.attach(9);
  aleron2.attach(10);
}

void loop()
// Se le asigna la posición inicial a los servos (0°)
{
  aleron1.write(0);    // Posición abajo
  aleron2.write(180);  // Posición arriba
  delay(2000);       // Una espera de 2 seg
  // Cambio de posición de los servos
  aleron1.write(180);   // Posición arriba
  aleron2.write(0);     // Posición abajo
  delay(2000);
}
```

Compilado

Figura 19. Código para funcionamiento del servomotor [41]
Elaboración propia

También se corrobora la funcionalidad del servomotor que controla el movimiento del sensor LiDAR, manejando la misma metodología de modificar un código ya existente.[41][42], teniendo en cuenta los requerimientos del proyecto , como se muestra en la figura 20. A diferencia de los servomotores de los alerones, el servomotor Lidar debe estar moviéndose de un lado a otro constantemente para realizar un barrido y cubrir un área más amplia para la detección.

Figura 20. Código Servomotor LiDAR[42]

Como el sensor LíDAR se instalará en un servomotor para generar un efecto de sensado tipo radar el Servomotor LiDAR también es simulado como se observa en la figura 21a y probado como se observa en la figura 21b.

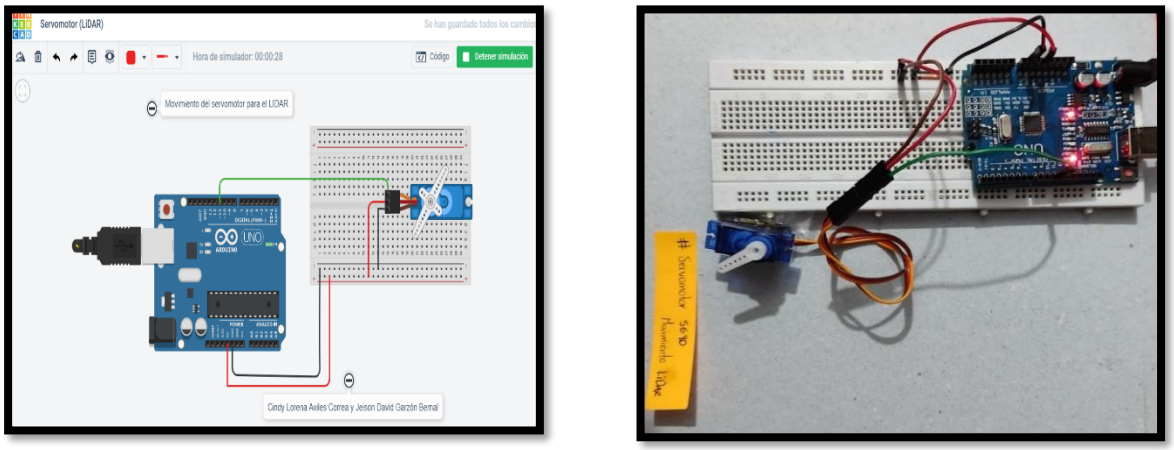


Figura 21. a) Simulación Servomotor LiDAR y b) Montaje Servomotor LiDAR
Elaboración propia

4.5 Validación de funcionalidad del motor Brushless.

Hay diversos códigos, diferentes opciones y formas de conexión para los motores Brushless, como por ejemplo, el uso de un potenciómetro para variar las revoluciones hasta el uso de una ventana serial que proporciona el programa Arduino IDE obteniendo el mismo resultado en ambos casos, de acuerdo a esto, se selecciona el código que permite el uso de la ventana serial que proporciona el programa porque es más simple , esta consiste en aumentar o disminuir las revoluciones del motor, disminuyendo

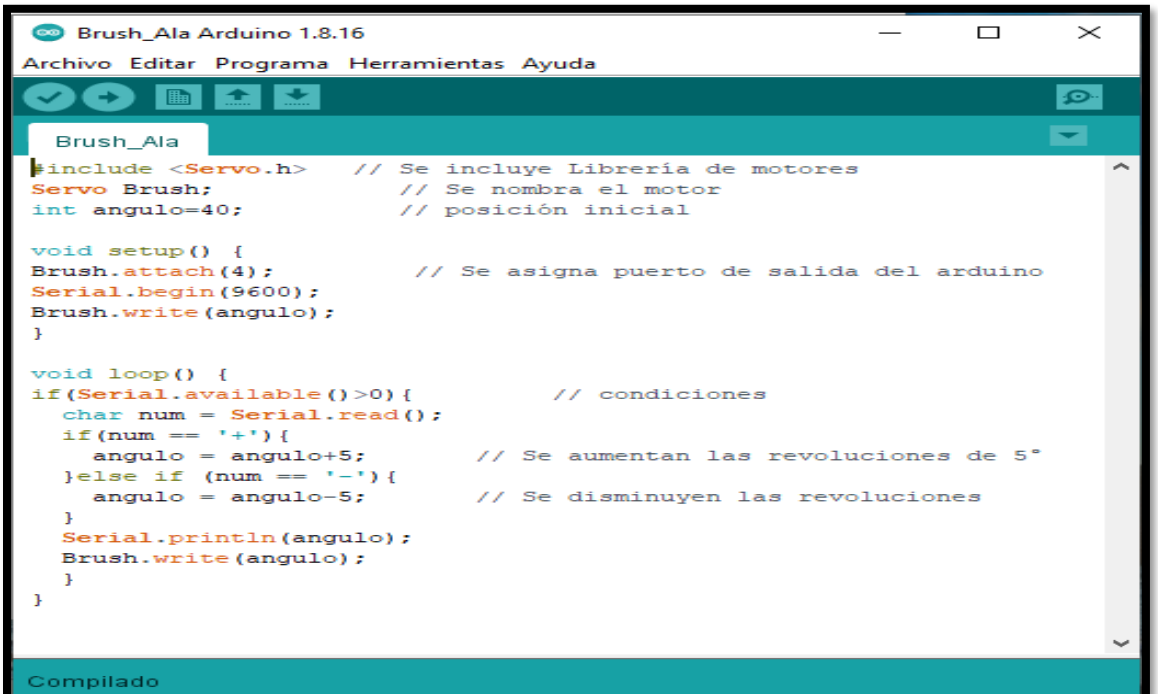


Figura 22. Código control de revoluciones

así las líneas de código y los componentes. En la Figura 22 se muestra el código del motor Brushless, este código se tomó de un curso de armado de drones brindado por la plataforma Udemy [43], siendo sencillo, fácil de montar y de entender.

Para el montaje real se tuvo en cuenta las sugerencias del fabricante, de conectarlo inicialmente sin hélices, ya que la fuerza de empuje ejercida por el motor sobre la hélice podría resultar riesgoso por desprendimiento de esta, por ello siempre se sugiere probar primero los motores sin hélices y ubicar el motor en un soporte por cuestiones de vibración.

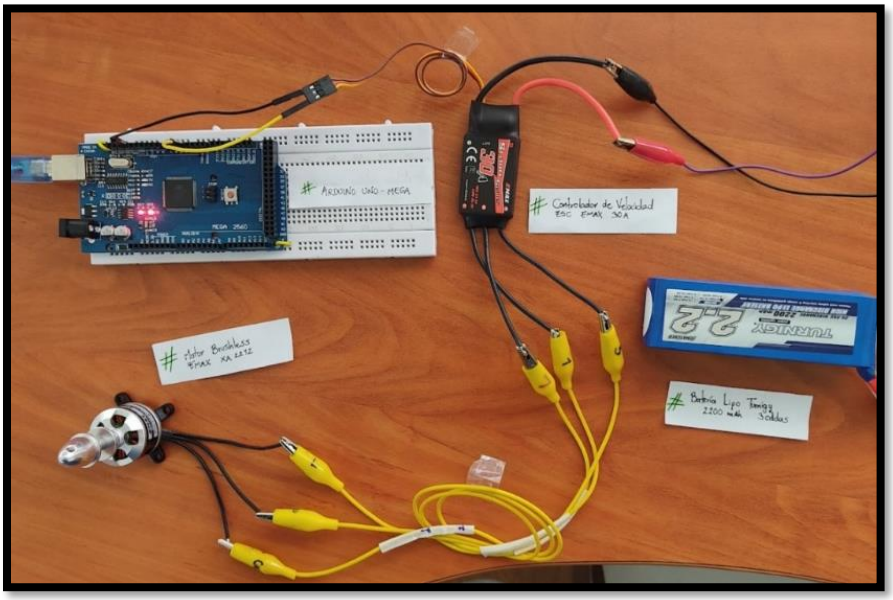


Figura 23. Montaje Motor Brushless

Elaboración propia

4.6. Sistema de detección (LiDAR, infrarrojo, cámara)

El proyecto se basa en la implementación de un sistema de detección de obstáculos, el cual está compuesto por tres sensores con características diferentes entre sí, pero que se complementan el uno al otro; por eso es importante realizar la comprobación de funcionalidad, determinar el rango de detección de cado uno (largo, corto y mediano alcance) y darle la ubicación correspondiente, con el propósito de validar los escenarios planteados en las tablas de verdad.

4.6.1 Sensor Infrarrojo

Cómo se explicó en el marco conceptual es un sensor de proximidad el cual mide la radiación emitida por otros cuerpos, teniendo de base un led receptor y otro transmisor, de acuerdo con la comparación

hecha en la Tabla 6, se escoge el sensor HW-201, para validar su funcionalidad se probaron varios códigos de programación, se ejecuta la simulación en el software Proteus porque tiene caracterizado el sensor en sus librerías. En la figura 24 [44] se muestra el código para realizar la validación del sensor infrarrojo y en la Figura 25 se muestran los montajes virtuales realizados en el módulo ISIS de Proteus.

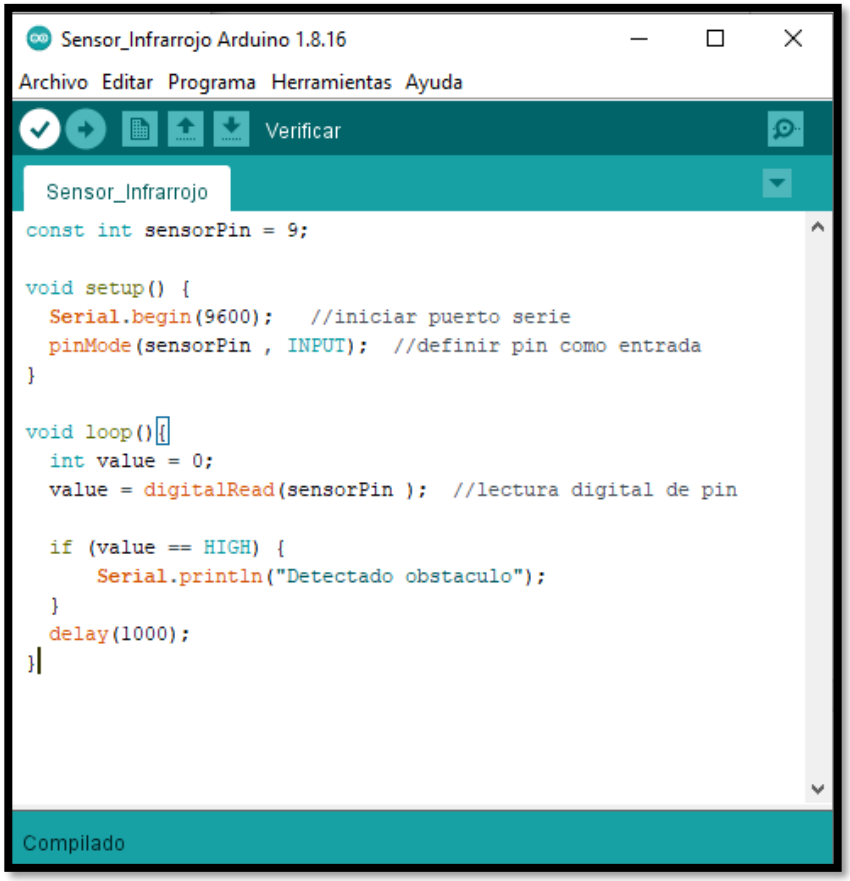


Figura 24. Código Sensor Infrarrojo [27]

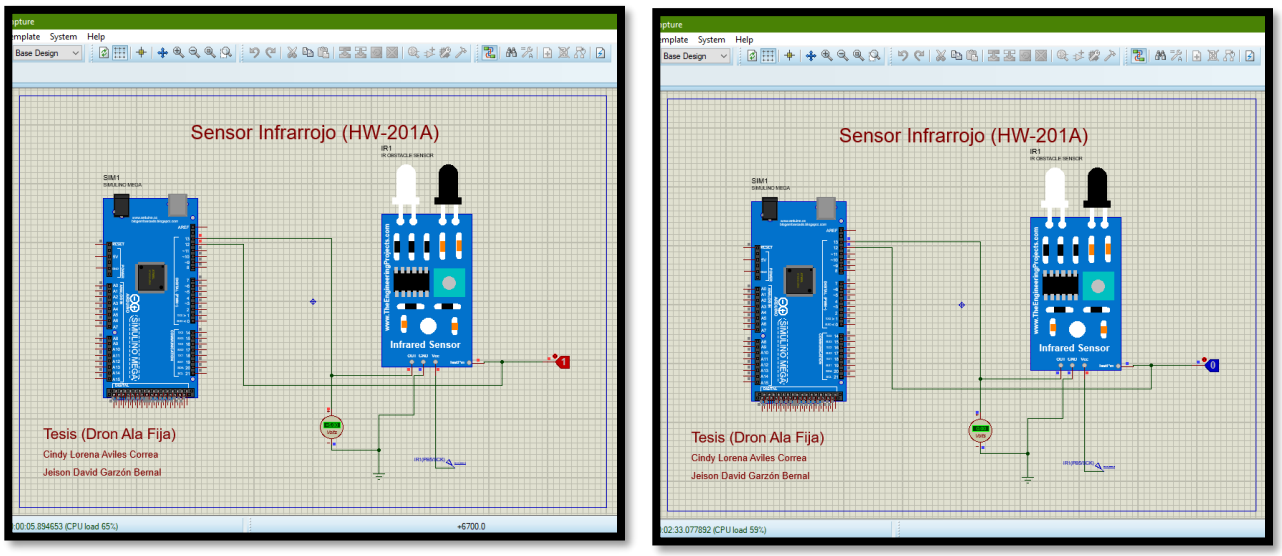


Figura 25. Con obstáculo y sin obstáculo
Elaboración propia

En la ilustración 25 al tener un valor binario de 1=activo en la entrada lógica, significa que está reconociendo un obstáculo. Para probar físicamente el rango de detección del montaje se realizó el circuito mostrado en la Figura 26, midiendo con una escuadra con resolución de milímetros, la distancia máxima de detención. En el montaje se confirma el rango de detección y se encuentra entre 0,3 a 150 mm o 15 cm, convirtiéndolo en el sensor de corto alcance.

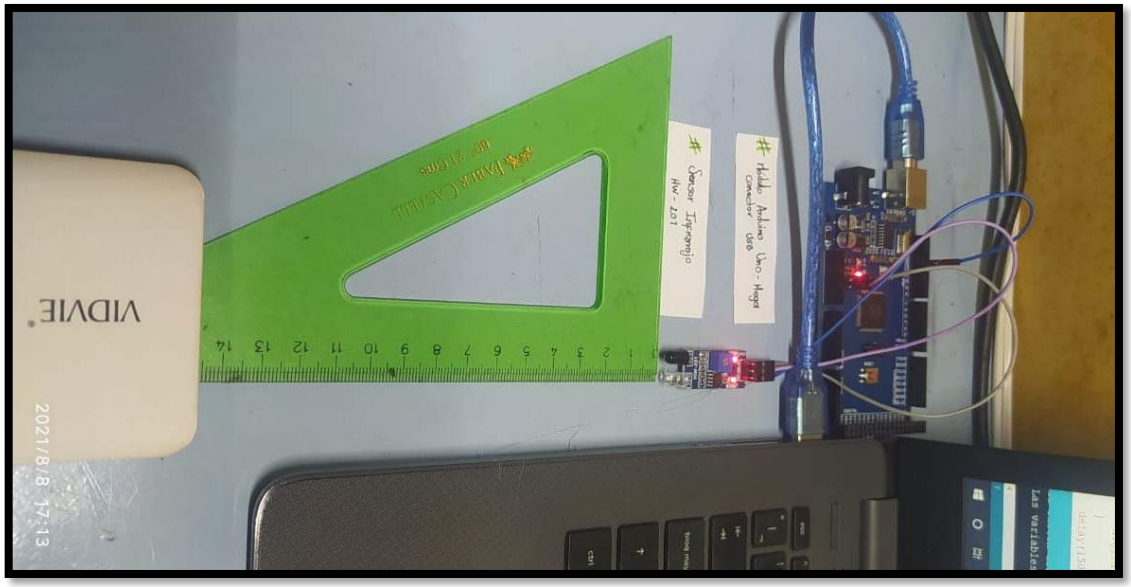


Figura 26. Montaje Sensor Infrarrojo

Elaboración propia

4.6.2.1 Código cámara SP32-CAM

La cámara seleccionada viene con una tarjeta (SP32-CAM), esta permite que la visualización se pueda ver online, ingresando a la dirección IP que genera al subir el código en el Arduino. Esta tarjeta cuenta con diferentes librerías que optimizan su funcionalidad, y un ejemplo de esta para comprobar que el dispositivo funcione adecuadamente.

Se verificaron dos códigos, uno con una red wifi que tenga navegación [45]como el mostrado en la figura 27b y otro en el que la misma tarjeta crea un punto de conexión wifi sin navegación [46]como el mostrado en la figura 27a. Los códigos completos se encuentran en el ANEXO número 1 y 2.

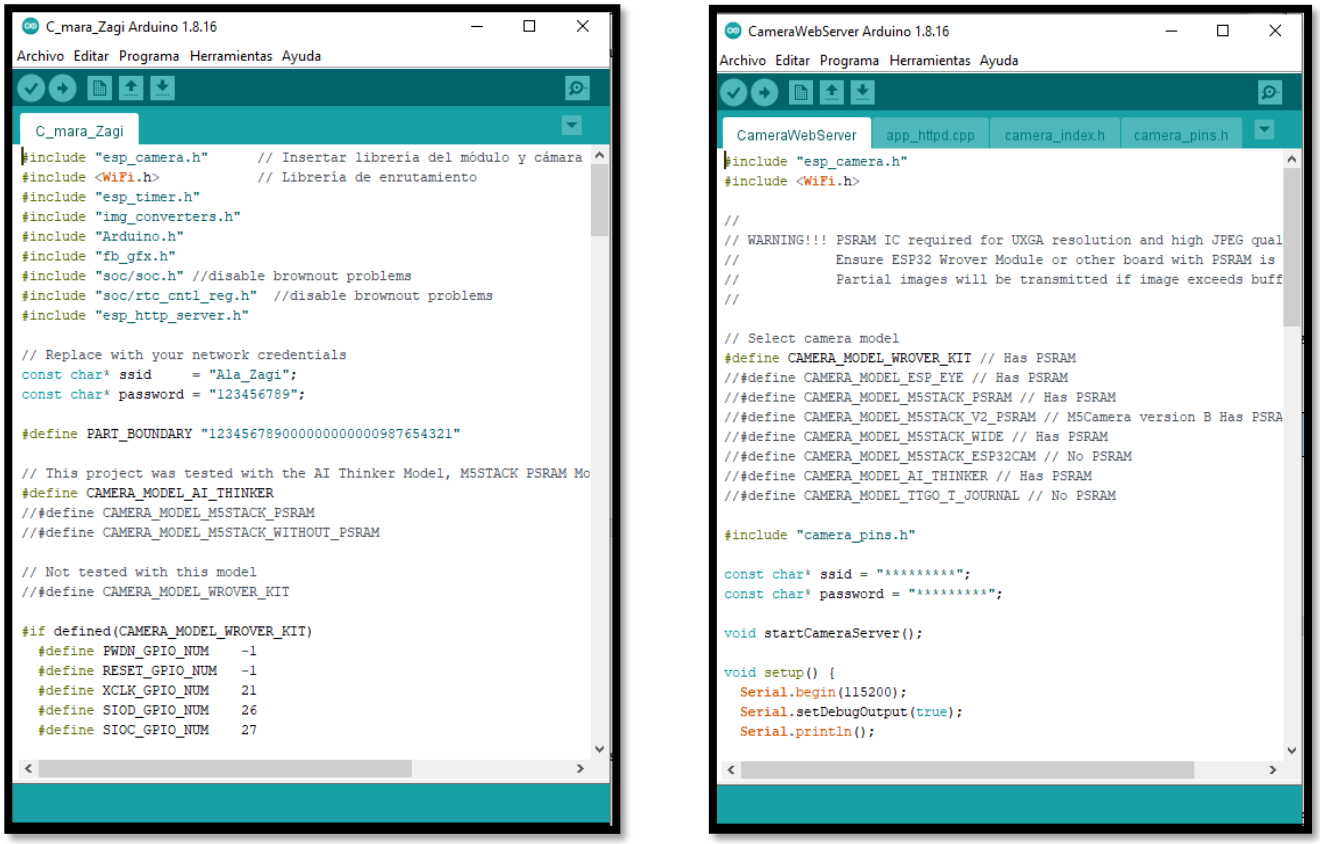


Figura 27. a) Código cámara (punto de acceso sin navegación) y b) Código 2 cámara con navegación

La imagen generada por el código sin acceso a internet de la Figura 27a, solo visualiza lo que está grabando, es decir, sólo muestra la imagen como se ve en la figura 28a y no tiene interfaz de usuario.

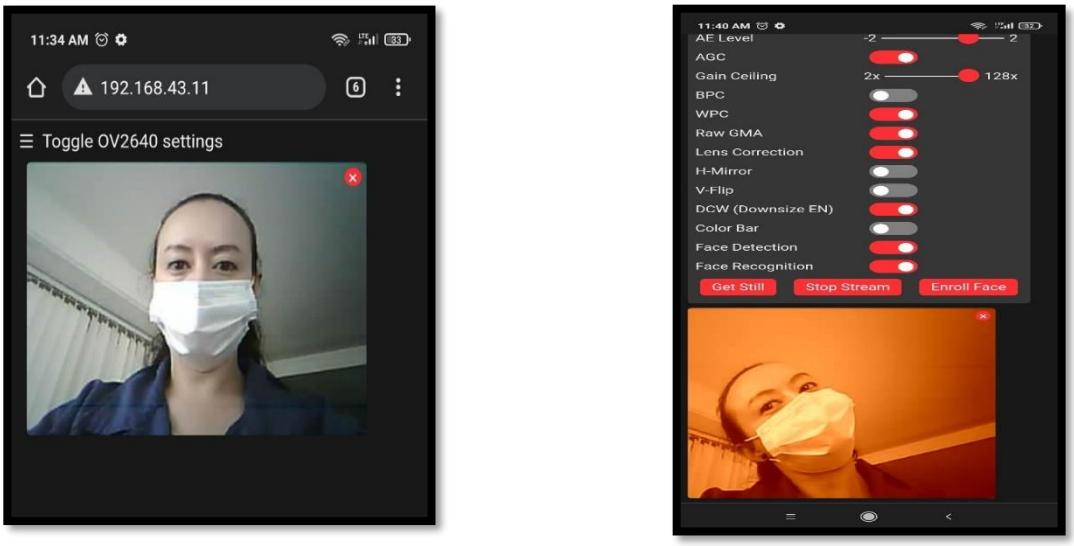


Figura 28. a) Visualización punto de acceso sin navegación y b) visualización con navegación e interfaz de usuario
Elaboración propia

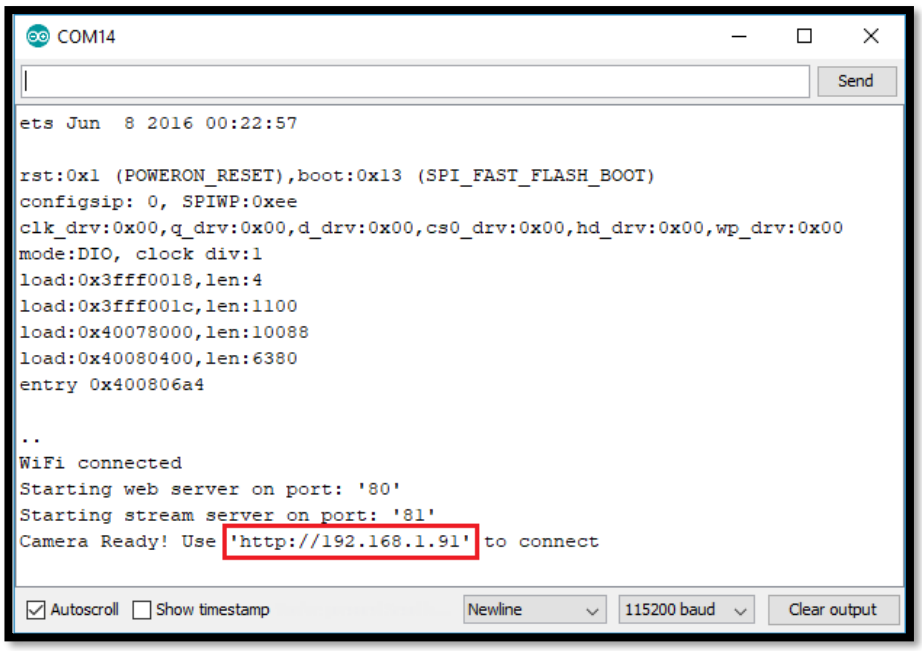


Figura 29. Monitor serial (dirección IP)[45]

Para el segundo código el cual el punto de acceso a la red si tiene navegación como se muestra en la figura 27b, al conectarse a una red wifi externa, con la dirección IP que genera el código en el monitor serial ver figura 28b, al colocarla en el navegador de la Tablet, computador o celular aparece una interfaz de usuario que permite modificar aspectos de la imagen como se observa en la Figura 27b. El código completo se encuentra en el ANEXO 2, es mucho más elaborado y tiene más opciones de tratar las imágenes. Finalmente, en la figura 30 se ilustra el montaje físico de la cámara articulada al Arduino.

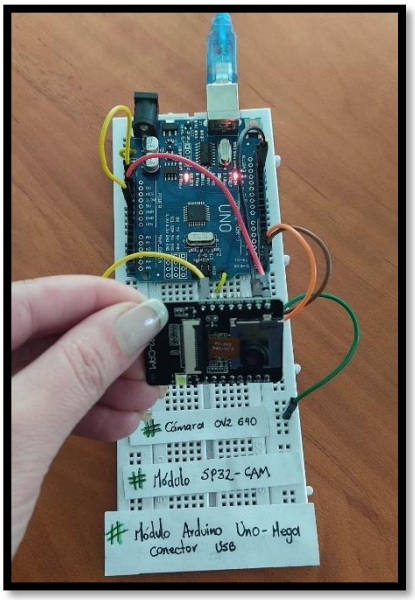


Figura 30. Montaje con cámara
Elaboración propia

4.6.3 LiDAR

Para este sistema primero fue realizada la comprobación de la funcionalidad del sensor láser, utilizando una variedad de códigos para Arduino, en donde muchos de ellos, la sintaxis, su escritura, sus librerías no funcionaban adecuadamente, se tomó el que mejor respondió y dio mejor funcionalidad al sensor, a continuación, en la figura 31a se visualiza el código, en la figura 31b simulación en TinkerCad , en la figura 32a el montaje físico y en la figura 32b el montaje funcionando y con los códigos articulados.

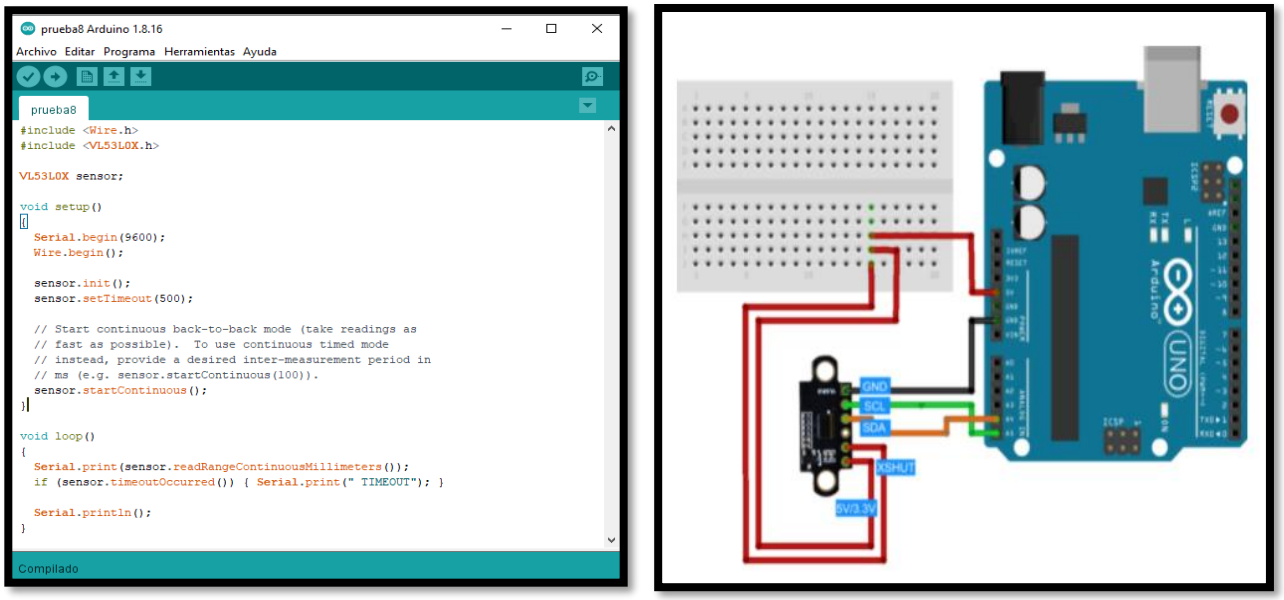


Figura 31. a) Código Sensor LiDAR [28][29] y b) Simulación Sensor LiDAR

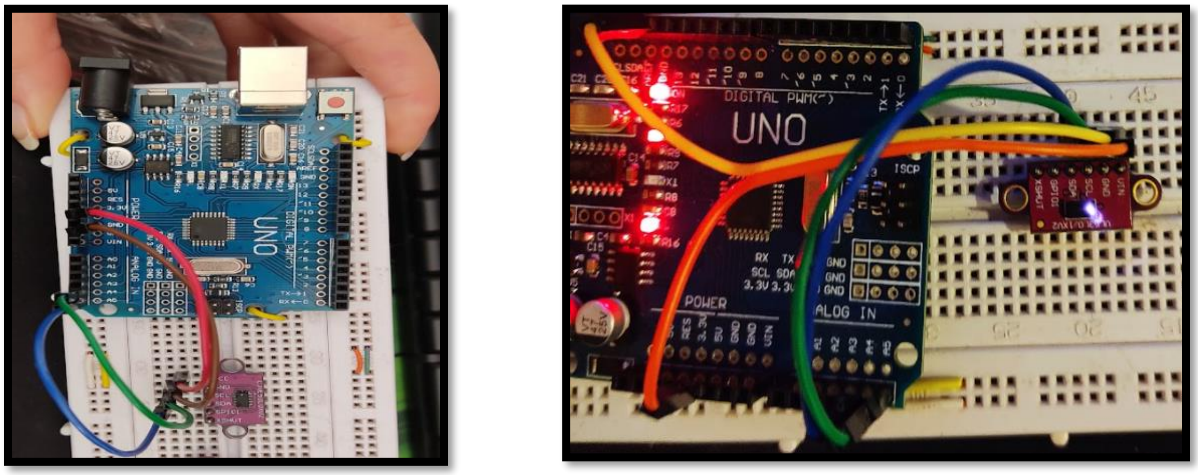


Figura 32. a) Montaje Sensor LiDAR y b) Sensor LiDAR (láser encendido)
Elaboración propia

Una vez fue comprobado de manera individual la funcionalidad de los componentes, se dispone a verificar del cómo lograr que todos funcionen de manera simultánea y se cumpla la tabla de verdad con respecto a los escenarios establecidos.

4.7 Construcción ala Zagi

Usando un modelo ya existente de dron de ala fija y la modificación del mismo según la necesidad del

proyecto, adaptada en un programa de diseño. Se tomó el diseño del dron ala fija Trimble uX5 [17] hecho en Solid Edge un software de diseño 3D, ver figura 33 [17], se dimensiona, y se determina que se va a ampliar las medidas tanto de las alas como del centro del dron. Se rediseñó y fabricó mediante el siguiente procedimiento.

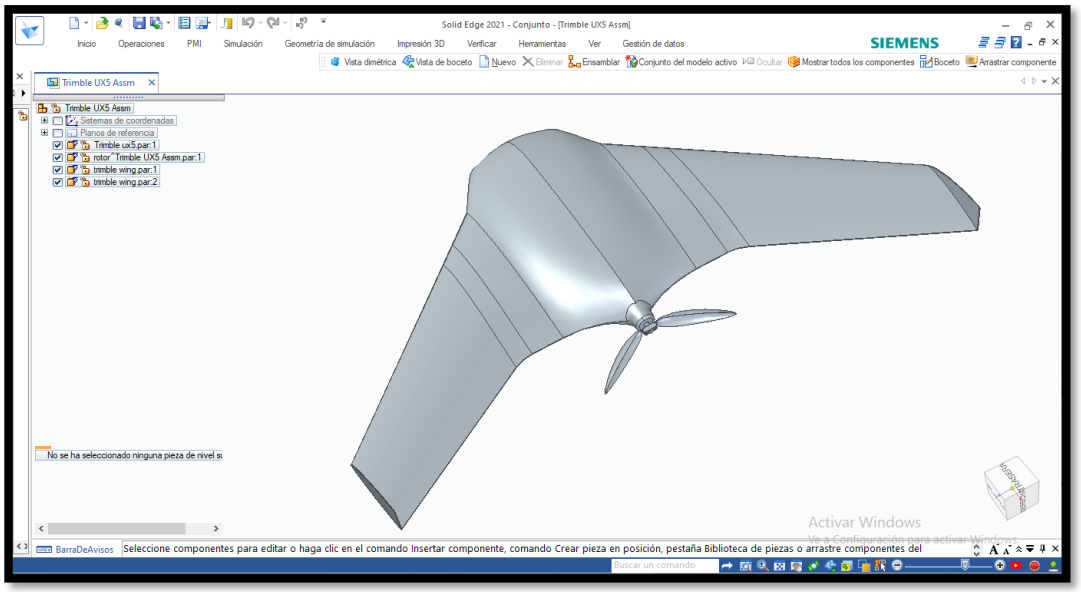


Figura 33. Diseño seleccionado Trimble uX5

Paso 1. Diseñar el ala volante con medidas como la envergadura, tipo de perfil, longitud y espesor, se procede con el siguiente método para la ubicación de las medidas visto desde un ángulo superior, se plantea tomar inicialmente 22cm que está dado por el perfil, a 50cm de distancia se hace una recta dando espacio para la reducción gradual al segundo perfil, donde este cuenta con una reducción de su tamaño del 30% comparándolo con el inicial de 22 cm, ubicado a ¼ superior de la línea longitudinal así como lo muestra la siguiente figura 34.

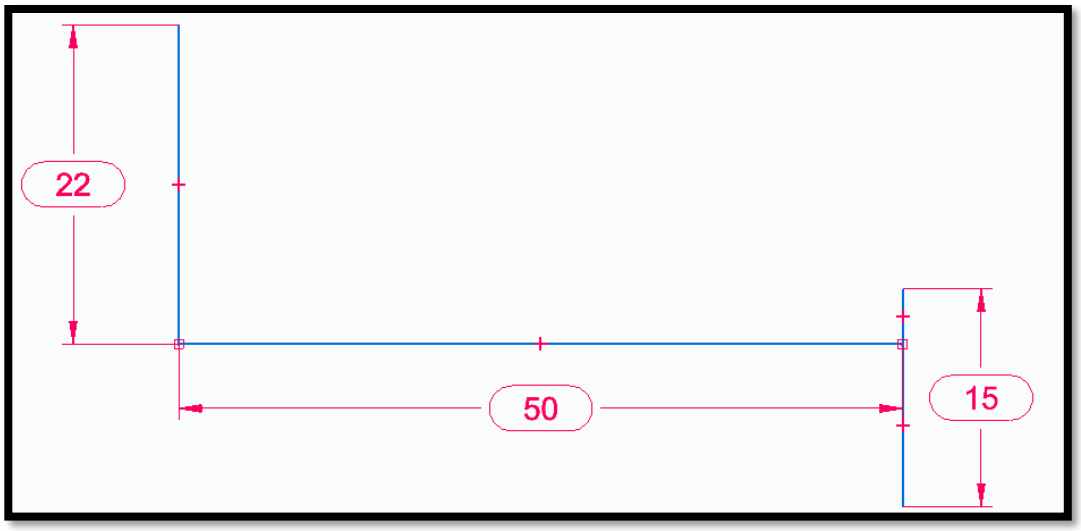


Figura 34. Diseño primera extensión de medidas del ala. *Elaboración propia*

Paso 2. Posteriormente se unieron las puntas, para completar la forma del ala como se ve en la figura 35.

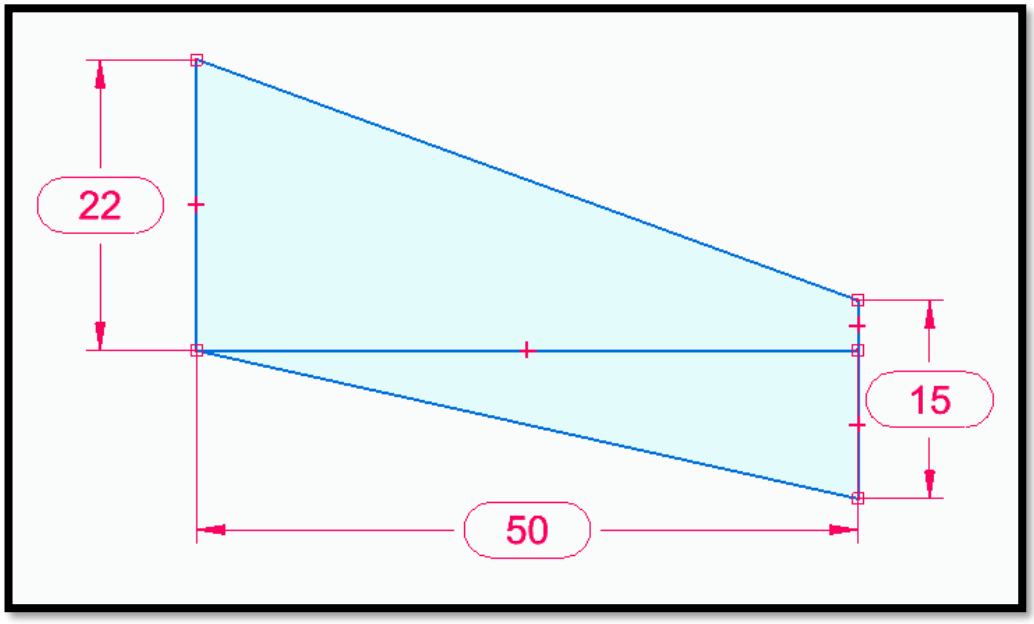


Figura 35. Diseño forma del ala. *Elaboración propia*

Paso 3. Para la construcción del ala volante se compararon dos tipos de perfiles a) un perfil simétrico y b) un perfil asimétrico, se seleccionó un perfil asimétrico ya que genera una mayor velocidad. Específicamente se escogió la referencia 2410 [47]. Como se muestra en la figura 36.

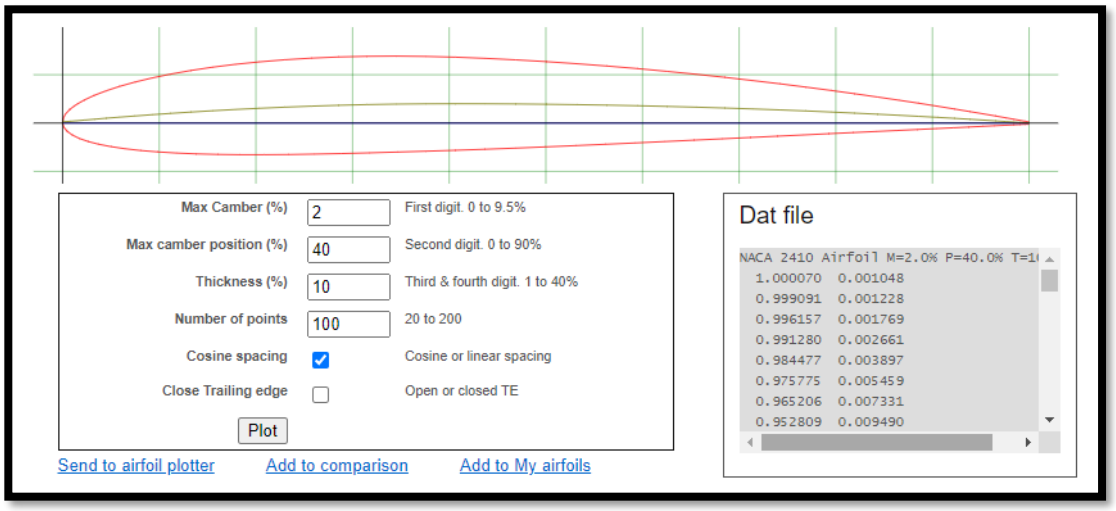
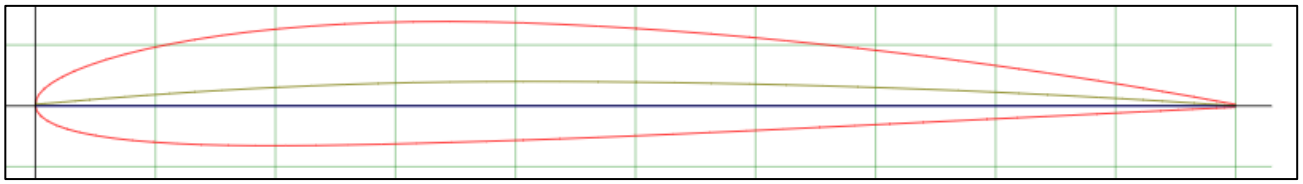


Figura 36. Selección perfil alar

Paso 4. Luego de esto se tomó el borde sombreado en rojo trasladándolo a un archivo formato PDF, para luego imprimirlo sobre acrílico; se hizo un molde de 22 cm de largo con 4cm de alto y 3mm de espesor y otro 15cm de largo con 2,5 cm de alto y 3mm de espesor. Como se observa en la figura 37.

Perfil alar 2410 de 22 cm



Perfil alar 2410 de 15 cm

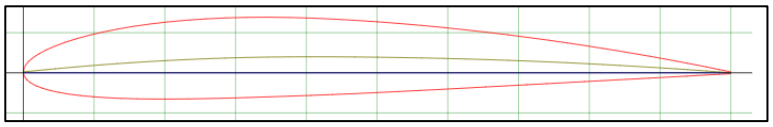


Figura 37. Perfiles
Elaboración propia

Paso 5. A partir de los moldes de dichos perfiles estos fueron cortados en acrílico con cortadora laser para una mayor precisión y simetría. Teniendo como resultado lo evidenciado en la figura 38.

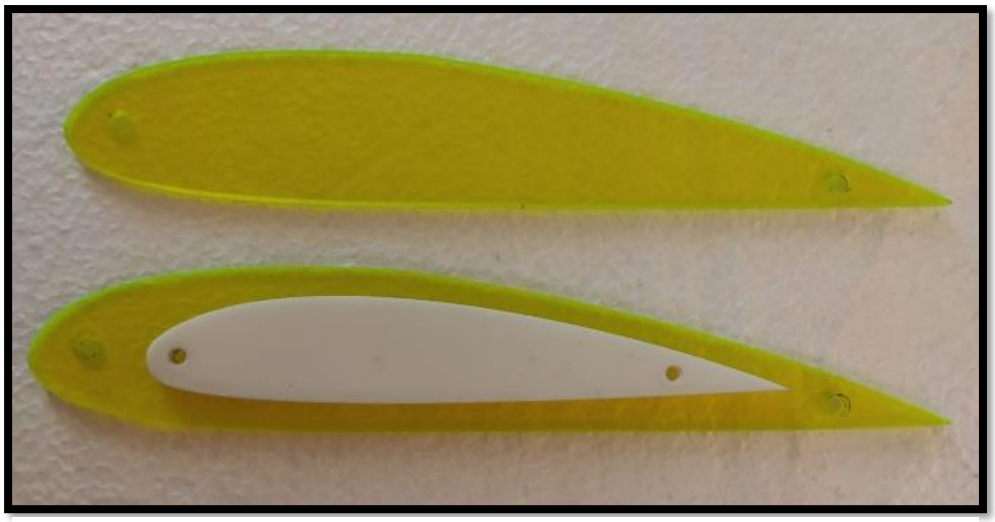


Figura 38. Perfiles alares en acrílico
Elaboración propia

Paso 6. Fueron realizados los esquemas en la placa de icopor efectuando cortes cuadrados y así delimitar el uso del material como se ve en la figura 39a usando los diagramas de la figura 39b.

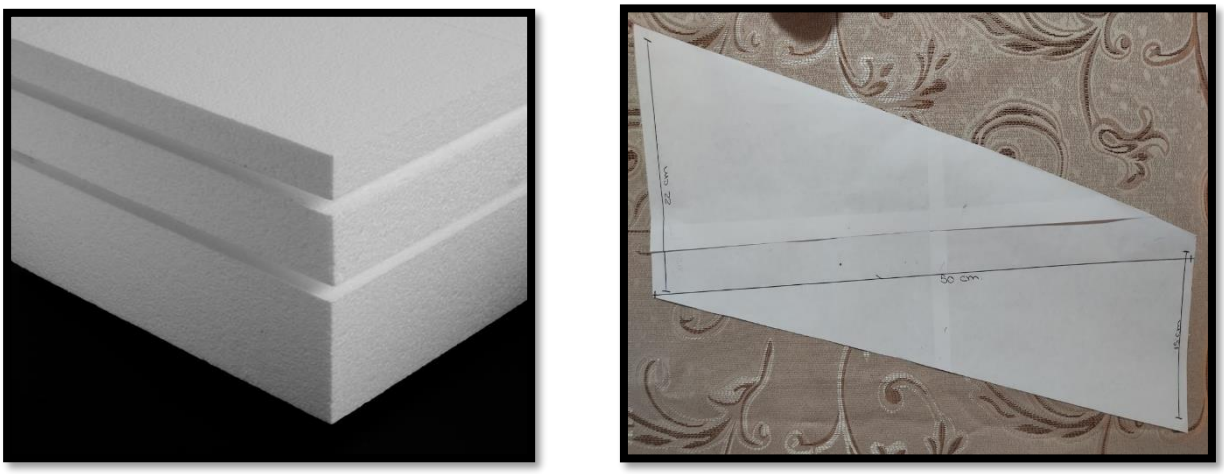


Figura 39. a) Placa de icopor y b) Plano ala (físico)
Elaboración propia

Paso 7. Se construyó una maquina cortadora de icopor como se aprecia en la imagen 40, para esta se usó; una roseta, una bombilla, cable dúplex, dos palos de madera y 1m de cable ferroníquel, se hace el montaje con conexión en serie para generar una resistencia por parte del bombillo, haciendo que el ferroníquel se caliente lo suficiente para cortar el icopor.

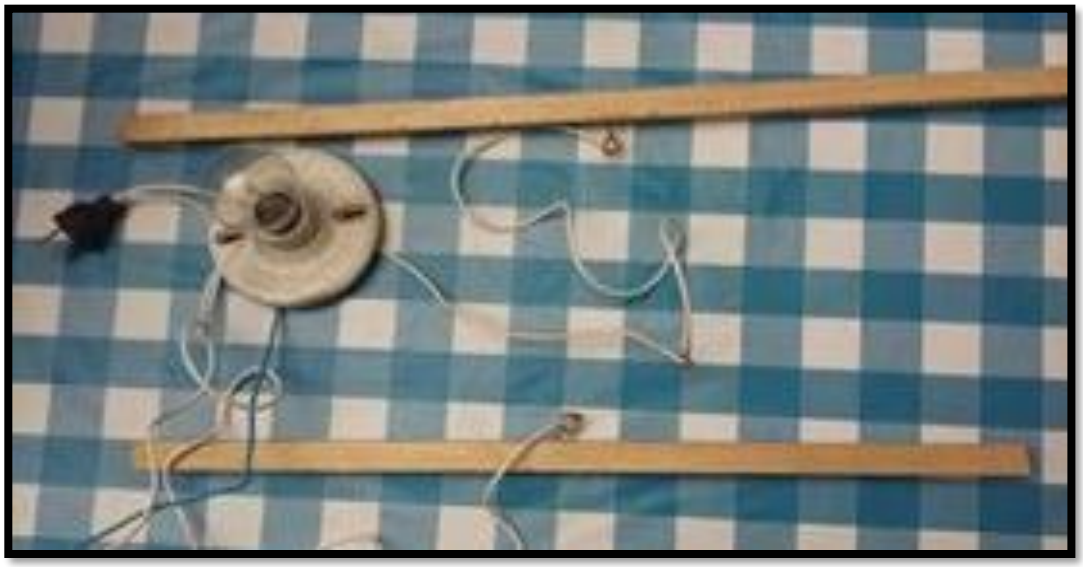


Figura 40. Máquina cortadora de icopor casera, elaboración propia
Elaboración propia

Paso 8. Se anclaron los perfiles alares a los bordes del ala con clavos para poder darle la forma al ala con la máquina de cortar icopor como se observa en la siguiente ilustración 41. Se mide el centro para colocar los perfiles y se usa la máquina para darle la forma del perfil alar 2410, estos cortes se ejecutan pasando el ferroníquel de lado a lado guiándose por el perfil que se anclo.

Perfil alar 22 cm

perfil alar 15cm

Figura 41. Proceso de corte Elaboración propia

Elaboración propia

Paso 9. Tomando como molde los perfiles fijados al icopor y pasando el ferroníquel queda como resultado lo que se aprecia en la figura 42 en la cual se pulen sus imperfecciones con una lija tipo 200 (grano fino). Se juntan los dos perfiles de 22cm a una distancia de 17 cm uno con respecto al otro, formando una caja base donde van los componentes eléctricos, junto con esto se ensamblan dos vigas estructurales de madera en cada ala generando un mejor acople y agarre entre las piezas.

Figura 42. Perfil alar en icopor
Elaboración propia

Paso 10. Seguidamente se unen con silicona líquida dejándolo secar por 24 horas, para garantizar una buena firmeza en el pegante y en la estructura, una vez ya seca el ala se hacen los cortes de los alerones 4 cm de ancho con 30 cm de largo (con 1,5 cm de tolerancia para su movimiento suba y baje) figura 43, y la caja en donde va el circuito procediendo a forrarla con papel Contac. Como se aprecia en la figura 44.

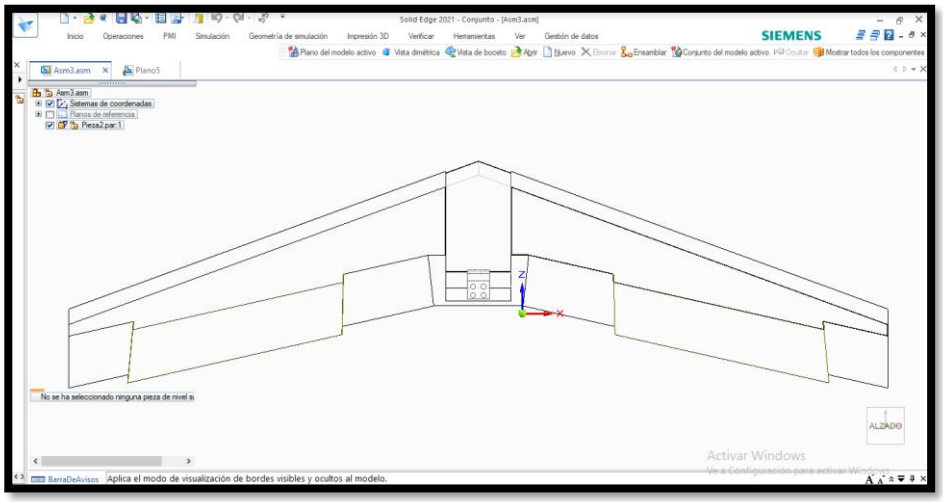


Figura 43. Diseño alerones
Elaboración propia

Figura 44. Forrado con Contac
Elaboración propia

Paso 11. Se cubre la caja base y junto con ella se ancla en la parte trasera una base con forma de L acoplada con tornillos para la ubicación del motor adicional se hace una tapa en forma de globo que cubre la parte superior del sistema asegurado con un botón de presión como se observa en la figura 45.

Figura 45. Forrado del dron
Elaboración propia

En la figura 46, se aprecia el resultado final del diseño y fabricación del ala Zagi.

Figura 46. Resultado del ala volante sin ningún componente eléctrico
Elaboración propia

4.7.1 Construcción base ala Zagi

Paso 1. Seleccionando los materiales los cuales fueron 3m de tubo PVC de diámetro de 1 pulgada, 2 conectores tipo T, 2 codos de 3 entradas, 4 codos de 90°.

Paso 2. Se toma como referencia la cavidad interna del dron que brinda un espacio de 17 cm, se corta 4 secciones del tubo a una distancia de 35 cm que dará como larguerillos de soporte, y 3 vigas laterales de 20 cm y 2 atravesaños de 20 cm.

Paso 3. Una vez cortados se procede a conectarlos donde toma una referencia a una base en forma de "C"
En la siguiente figura (47) se evidencia el diseño en SolidEdge de la base con cotas de mm.

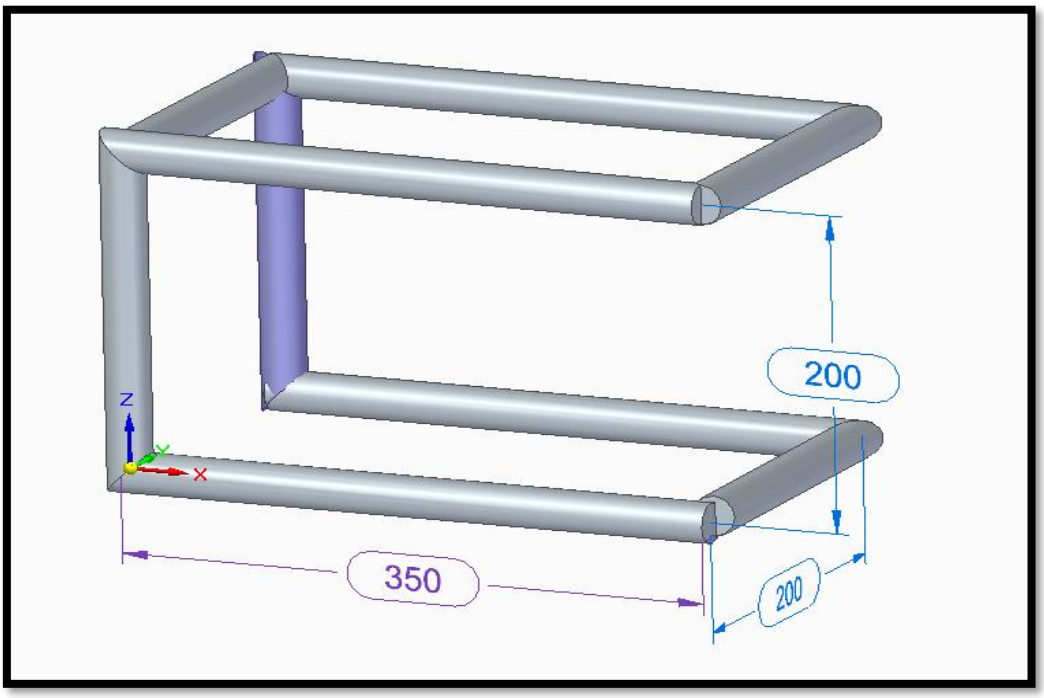


Figura 47. Diseño base tierra
Elaboración propia

En la figura 60 se refleja la base en físico junto con el dron ya terminado.

Capítulo 5

Análisis y Resultados

5.1 Simulación y montaje

Realización de la simulación y montaje del circuito electrónico para la detección de obstáculos mediante sensor de distancia infrarrojo, sensor LiDAR y la cámara de vídeo.
De acuerdo a los pasos y proceso hecho en la sección de la metodología, la simulación de los sensores, el montaje en el simulador y el modelo construido, fueron articulados los tres sensores para lograr la detección y evasión de obstáculos, esto se logró mediante un proceso de ensayo y error, utilizando los diferentes códigos encontrados para cada uno de los componentes adicional a los sensores, fueron programados también los servomotores que controlarán el movimiento de los alerones, el servomotor que le dará movimiento al sensor LiDAR para que su rango de detención sea de 80° continuos y el control de las revoluciones del motor brushless determinadas por la tabla de verdad número 2. Todo fue acoplado a una sola tarjeta programable Arduino UNO seleccionada por el tamaño reducido con respecto al Arduino Mega, mediante una librería de Arduino IDE llamada Protothreads en la cual se adapta la configuración de tiempos en el que los códigos inician de manera sincrónica y que se puede manipular la información de todos los dispositivos en el monitor serial, confirmando la correcta implementación y articulación sinérgica de las señales de los sensores. A continuación, se muestra lo realizado con cada uno de los sensores hasta llegar al código final.

5.1.1 Unión de códigos

Antes de iniciar con el acoplamiento de los diferentes códigos mostrados en el documento, se investigó cómo se puede ejecutar dichas tareas de forma simultánea sin que el código final no genere sobre procesamiento, en este caso y de toda la información recopilada se encuentra el uso de una librería en Arduino llamada Protothreads [48], mediante información escogida desde la plataforma GitHub ver figura 48 y modificada a los requerimientos del proyecto, de una manera sencilla, ayuda a unificar varios comportamientos asignados a sensores, actuadores o simplemente a unos LED, fue la adecuada para condicionar y hacer funcionar adecuadamente los sistemas de detección, parametrizando los delays (retardos) entre cada código, implementando las diferentes librerías y variables declaradas.

Figura 48. Estructura de código Librería Protothreads [48]

5.1.2 LiDAR

En esta sección se unen los dos códigos de los componentes que forman el sistema de detección con un sensor láser conocido por sus siglas LiDAR. Para asegurar que el dispositivo sea de bajo costo se implementa el sensor sobre un servomotor que le da movimiento para que su rango de cobertura sea mayor y su tiempo de rotación sea el adecuado y funcional para la misión. En la figura 49a se muestra el encabezado del código y en la figura 49b el montaje de los dos elementos (servo + sensor), compilando el código no refleja fallas.

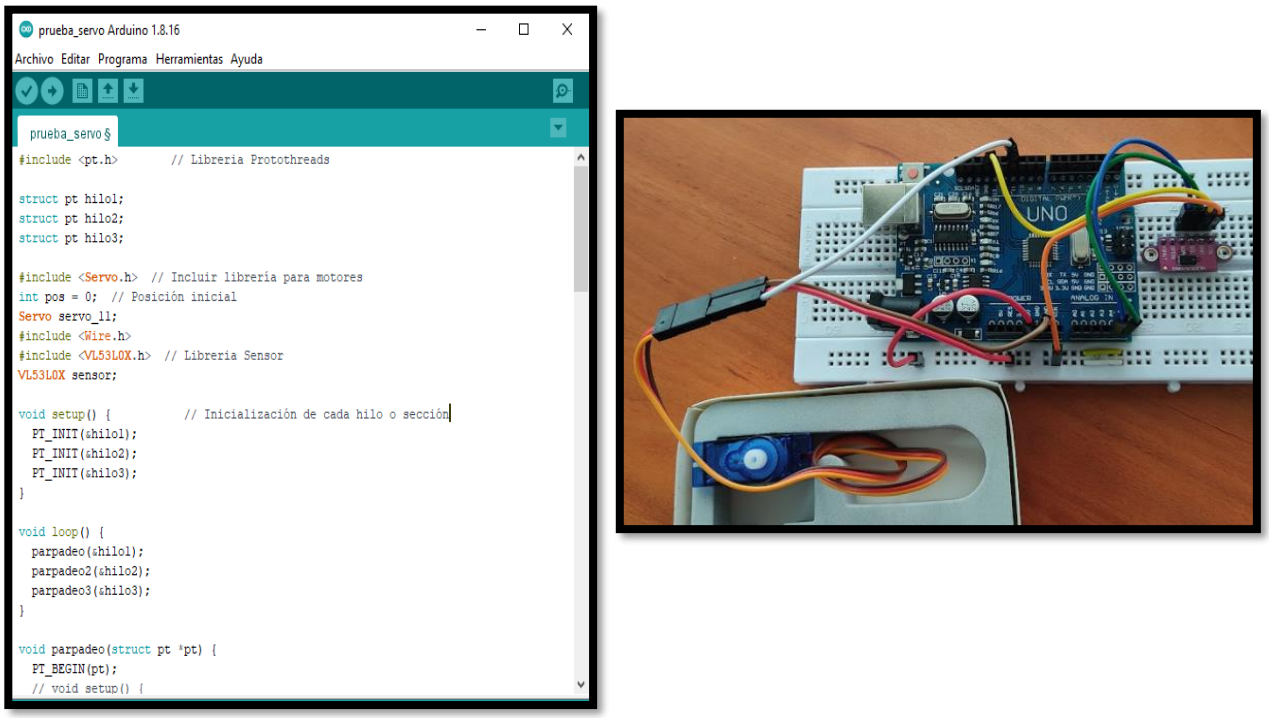


Figura 49. a) Código y b) Montaje, Servomotor + LiDAR, *Elaboración propia*

5.1.3 Infrarrojo

Con respecto al sensor infrarrojo, se tomó el código de la figura 20 siendo este funcional para cumplir los requerimientos del proyecto y se adicionó a la librería de Protothreads ingresando un bucle que se ejecuta en simultaneo con los códigos del sensor LiDAR y servomotor parte del código se visualiza en la figura 50a, por consiguiente, se hace el montaje en protoboard de los diferentes componentes para validación de conexiones y funcionalidad ver imagen 50b.

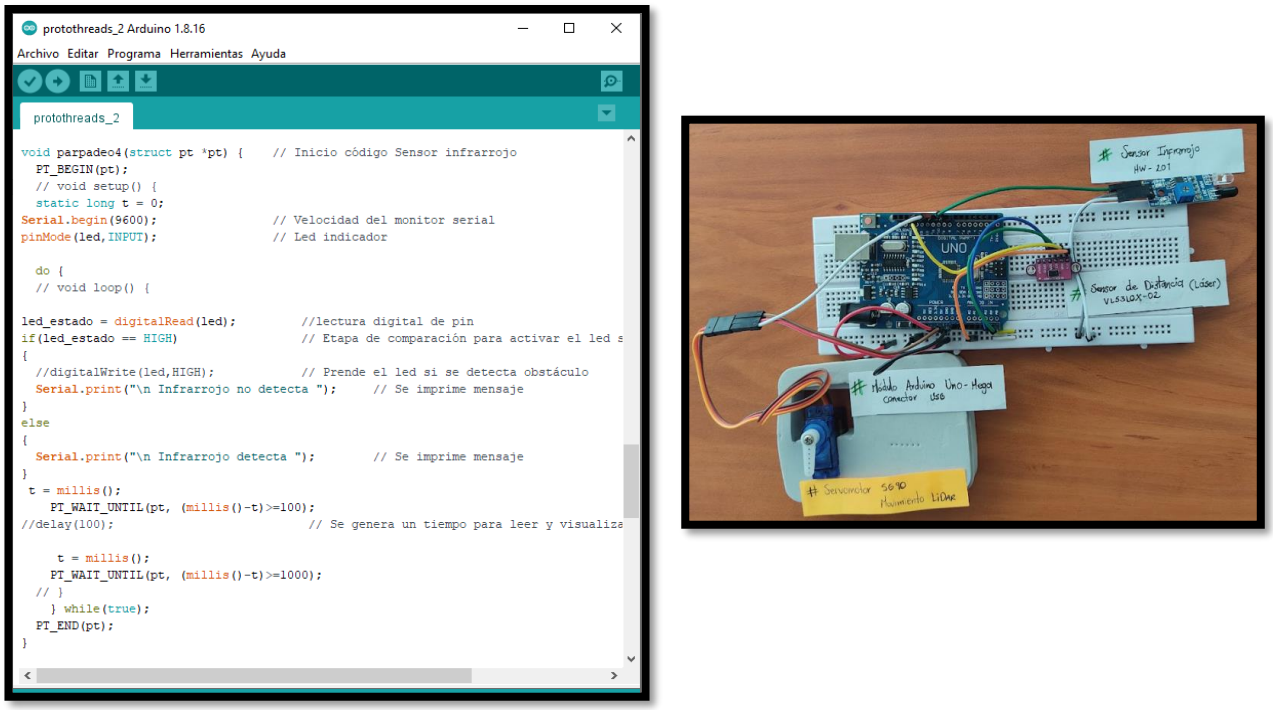


Figura 50. a) Código infrarrojo en Prototheads y b) Montaje LiDAR + Infrarrojo
Elaboración propia

5.1.4 Alerones

De igual manera los alerones tienen un código independiente (ver figura 18), siendo validado su correcto funcionamiento con montaje individual, se dispone a acoplar el código en la librería de prototheads y hacer el montaje correspondiente, en la siguiente figura 51 se puede visualizar la sección en donde estará ubicada la programación y el montaje en físico de los sensores y los servomotores (3); siendo compilado sin errores de sintaxis se logra el cargue adecuado, teniendo como resultado el movimiento de los 3 servomotores y los dos sensores con su correspondiente detección.

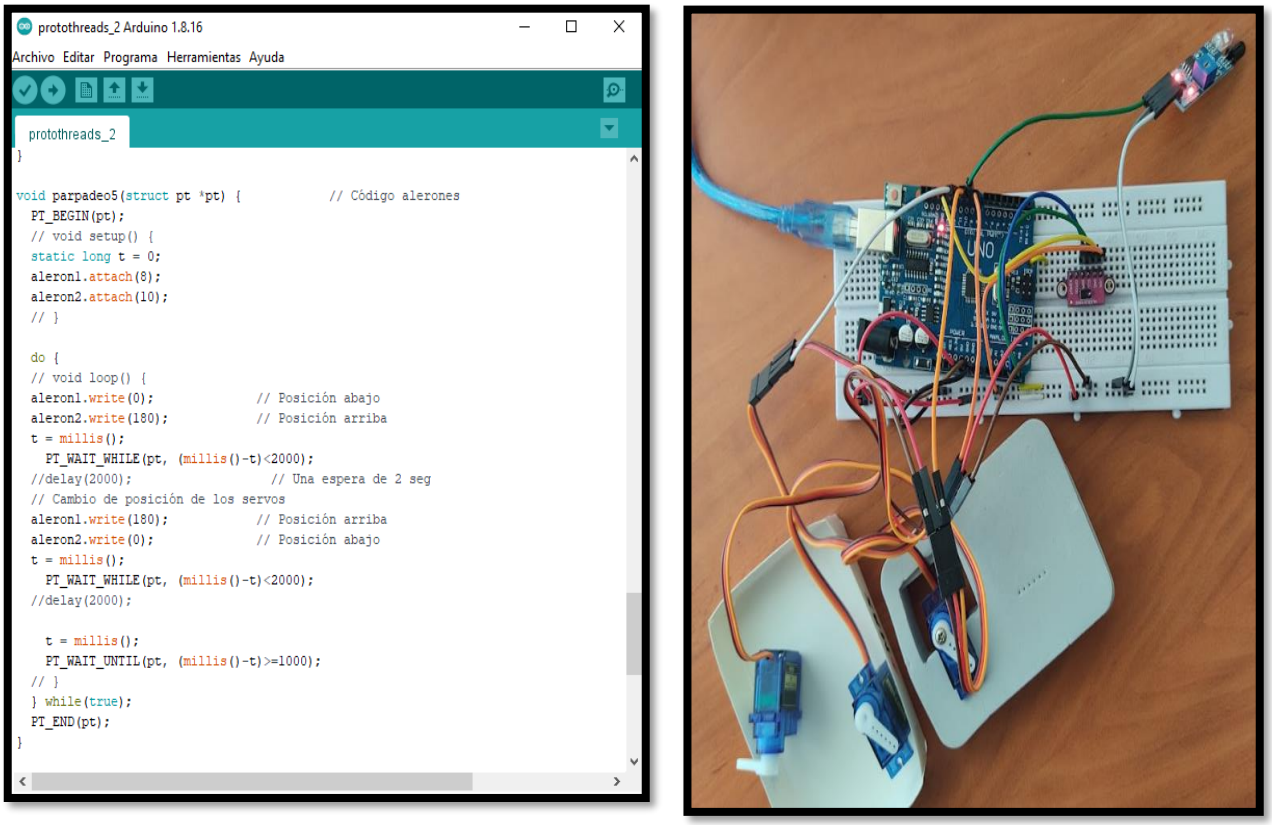


Figura 51. a) Código LiDAR, infrarrojo, alerones y b) Montaje LiDAR, Infrarrojo, Servomotores
Elaboración propia

5.1.5 Motor Brushless

El movimiento del motor sin escobillas se determinó por medio del monitor serial, dando el aumento o disminución a las revoluciones, el motor empieza a girar con velocidad constante sin acelerar a las 50 revoluciones, dato confirmado cuando se valida el funcionamiento del motor de forma independiente en la metodología, en la siguiente imagen (figura 52a), se puede observar la ubicación de la programación correspondiente al igual que el montaje mostrado en la figura 52b, sin errores en la compilación y la labor ejecutada por el sistema sin anomalías.

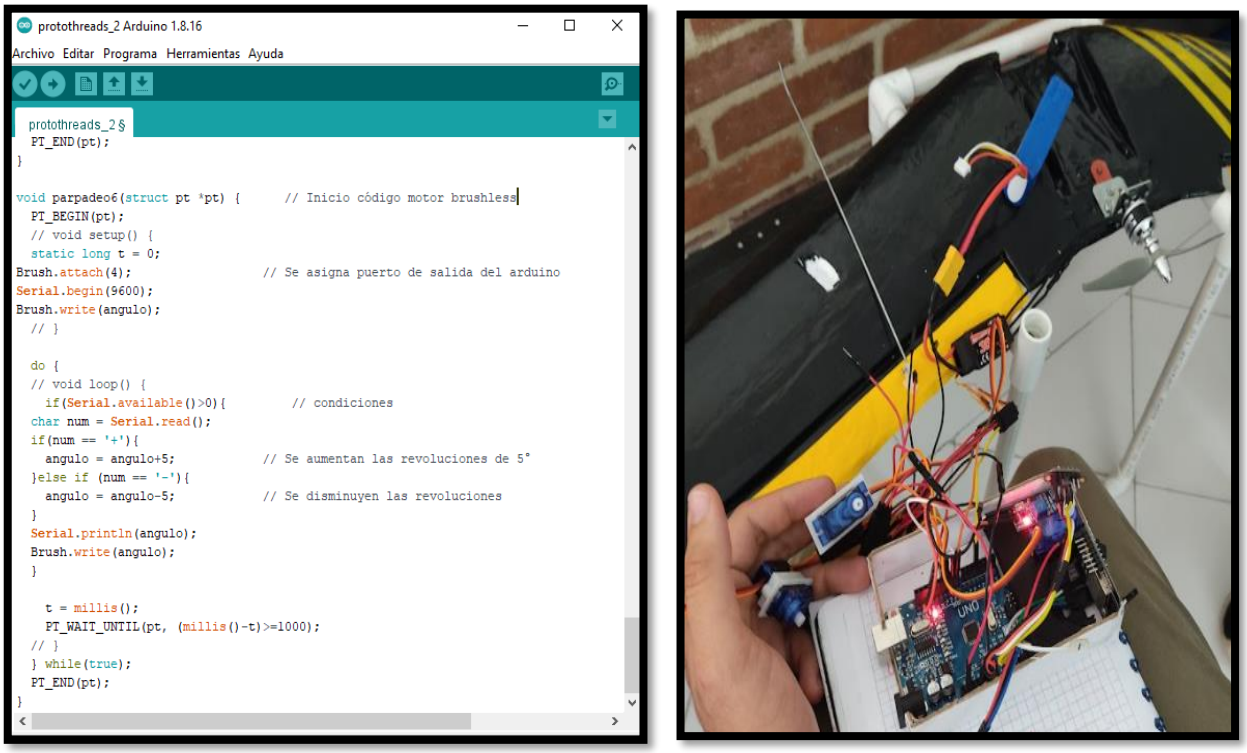


Figura 52. a) Código PT LiDAR, Infrarrojo, servomotores y brushless y b) Montaje
Elaboración propia

5.1.6 Cámara

Al adicionar el código de la cámara, este al momento de compilar en la librería protothreads y al hacer las debidas conexiones físicamente, afecta negativamente a la funcionalidad de los demás dispositivos ya que su conexión y cableado necesita un reset constante en el sistema para la navegación y generación de dirección IP, pausando la continuidad de las líneas afectando los demás circuitos. Por ello se decide implementar el conector Ftdi que venía incluido en la compra de la tarjeta SP32CAM, creando un sistema independiente y funcional sin interrupciones con el mismo, para ello la alimentación también será independiente, por medio de un cable mini USB.

5.2 Sistema de control

Selección del sistema de control y actuación de las superficies primarias de control de vuelo (alerones) del dron.
Esta selección se realizó a partir de un algoritmo de decisión implementado en un micro controlador, Arduino UNO, para darle el control a la decisión de cada movimiento se le atribuye a las condicionales establecidas a partir de las Tablas 1 y 2, parte del código visualizado en la figura 53, código completo

en anexo 3 de manera sencilla se cumple con lo establecido, para la optimización de líneas, capacidad de procesamiento y cantidad de información. A continuación, se muestra la comprobación de las tablas (1 a la 4) con los escenarios correspondientes.

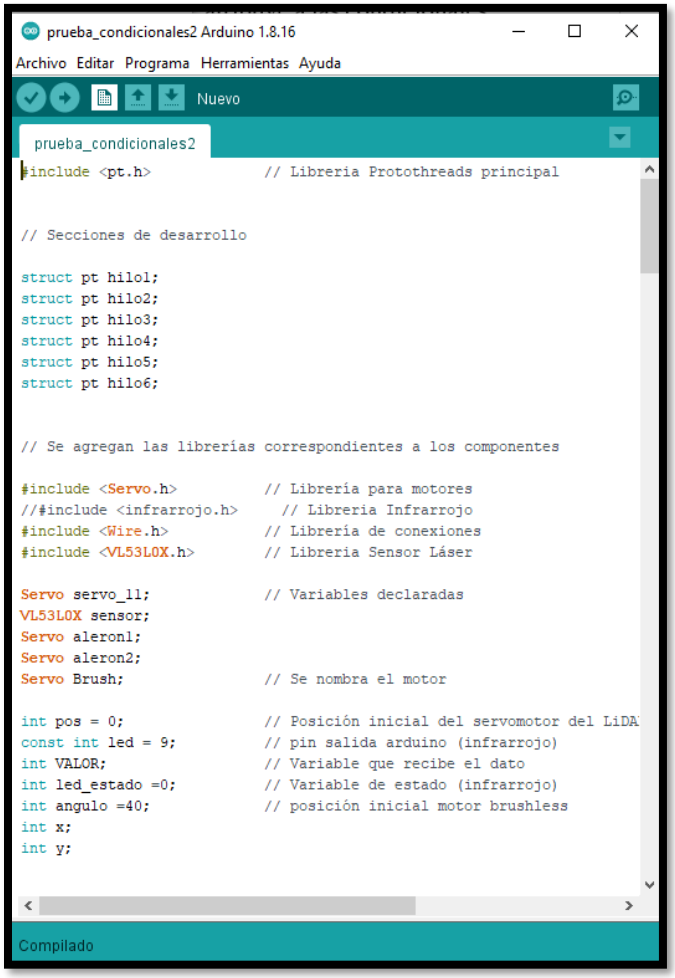


Figura 53. Código PT Final
Elaboración propia

Al realizar la compilación del código (ver figura 53) y el cargue del mismo a la tarjeta controladora, en el monitor serial la información que muestra no es clara, y no corresponde a las medidas del sensor LiDAR, cuando se inicia la comprobación de la tabla 1 y 2 genera error por lo cual se decide cambiar el código inicial del sensor LiDAR (ver figura 27) por el siguiente código visualizado en la figura 54, código completo en anexo 4 [49], validado individualmente teniendo mejor respuesta en el monitor serial con respecto a la distancia medida, al ser acoplado a la librería Prothothreads continúa fallando la compatibilidad de los códigos siendo la causa de la falla los retardos de cada uno, optando por hacer el código final uniéndolos de manera secuencial sin generación de hilos.

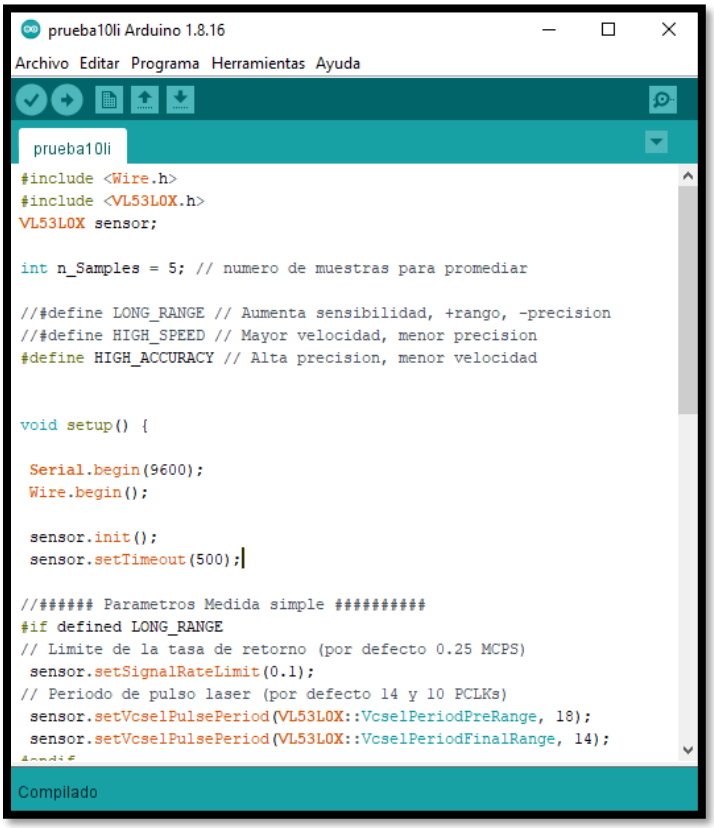


Figura 54. Código Sensor LiDAR Final. [49]

Después de ajustar, acoplar y organizar todos los códigos en una nueva programación se inicia la validación de los escenarios, teniendo como resultado con visualización en el monitor serial las siguientes ilustraciones, iniciando con el código definitivo en la figura 55, código completo en anexo 5.

Figura 55. Código Final (condicionales)
Elaboración propia

En la figura 56 se puede evidenciar la información en el monitor serial con respecto a la primera condición de la tabla de verdad número 1 (0:0) con resultados de la tabla 2, en el código para el primer escenario los alerones están hacia abajo (con posición de 180°) indicando pitch arriba como comportamiento del dron. En donde la ventana del monitor serial indica Fuera de rango (d > 2 m) como condición del LiDAR es decir no detecta, el aviso de no detecta obstáculo es emitido por el sensor infrarrojo y el número 50 significa las revoluciones del motor (vuelo constante), confirmando así el primer escenario (ver Tabla 4).

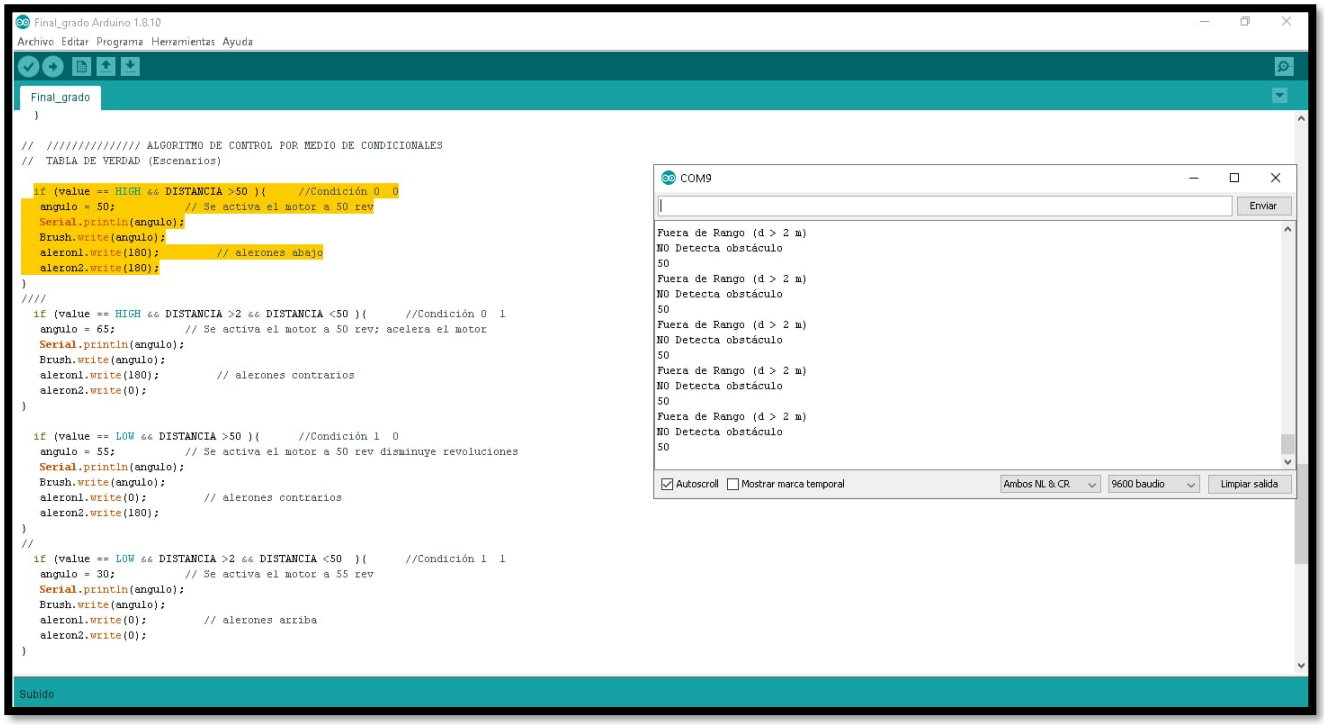


Figura 56. Condición 1 (0:0)
Elaboración propia

En la figura 57 se puede evidenciar la información en el monitor serial con respecto a la segunda condición de la tabla de verdad número 1 (0:1) con resultados de la tabla 2; en el código para el segundo escenario el aleron derecho está en posición abajo y el aleron izquierdo con posición arriba efectuando un alabeo como respuesta evasiva. En donde la ventana del monitor serial muestra una medida dada en cm indicando que el LiDAR detecta, el aviso de no detecta obstáculo es emitido por el sensor infrarrojo y el número 65 significa el aumento de las revoluciones del motor, confirmando así el segundo escenario.

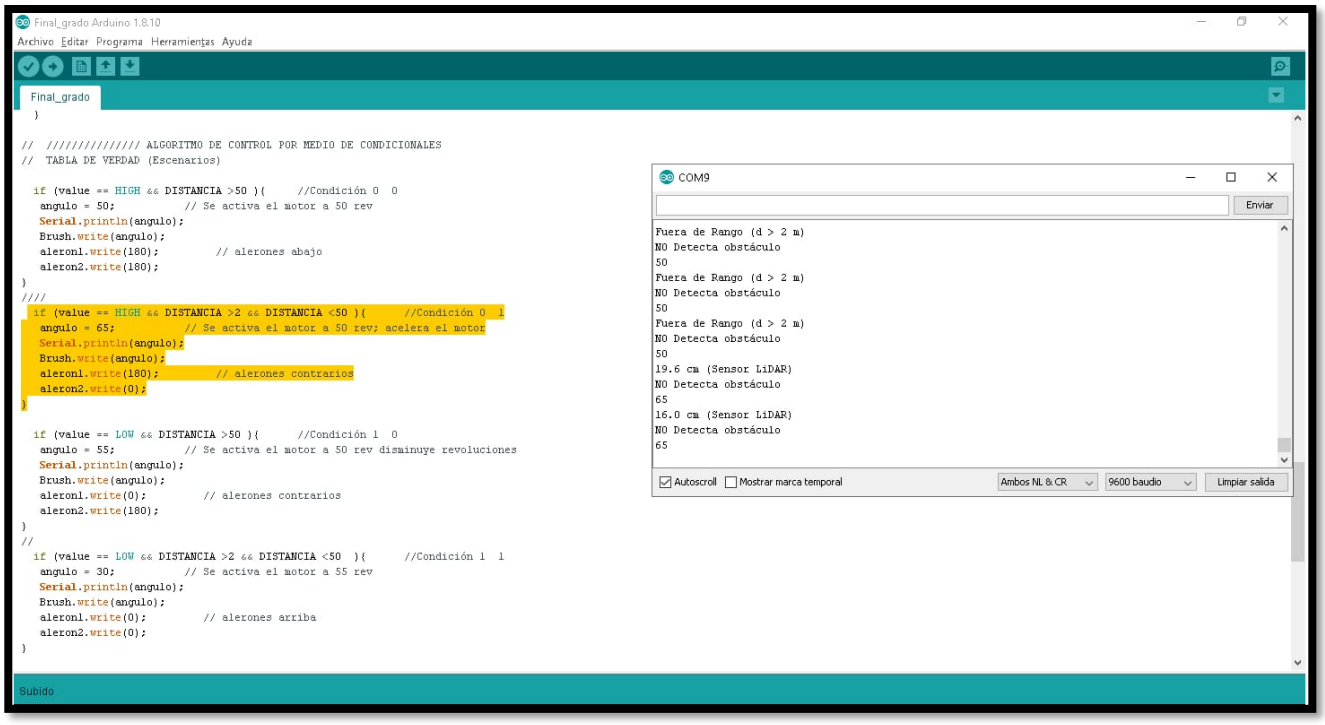


Figura 57. Condición 2 (0:1)
Elaboración propia

En la figura 58 se puede evidenciar la información en el monitor serial con respecto a la tercera condición de la tabla de verdad número 1 (1:0) con resultados de la tabla 2; en el código para el tercer escenario el aleron derecho está en posición arriba y el aleron izquierdo con posición abajo efectuando un alabeo como respuesta evasiva. En donde la ventana del monitor serial muestra Fuera de rango (d > 2 m) como condición del LiDAR es decir no detecta, el aviso “detectado obstáculo por infrarrojo” resultado de la reacción del sensor modificando así la disminución de las revoluciones del motor (55 rev), confirmando así el tercer escenario.

En la figura 59 se puede evidenciar la información en el monitor serial con respecto a la cuarta condición de la tabla de verdad número 1 (1:1) con resultados de la tabla 2; en el código para el cuarto escenario los alerones deben estar en posición arriba (con posición de 0°) efectuando un pitch abajo como respuesta evasiva. En donde la ventana del monitor serial muestra una medida dada en cm indicando que el LiDAR detecta, el aviso “detectado obstáculo por infrarrojo” resultado de la reacción del sensor modificando así las revoluciones del motor (30 rev) haciendo que este se detenga, confirmando así el cuarto y último escenario.

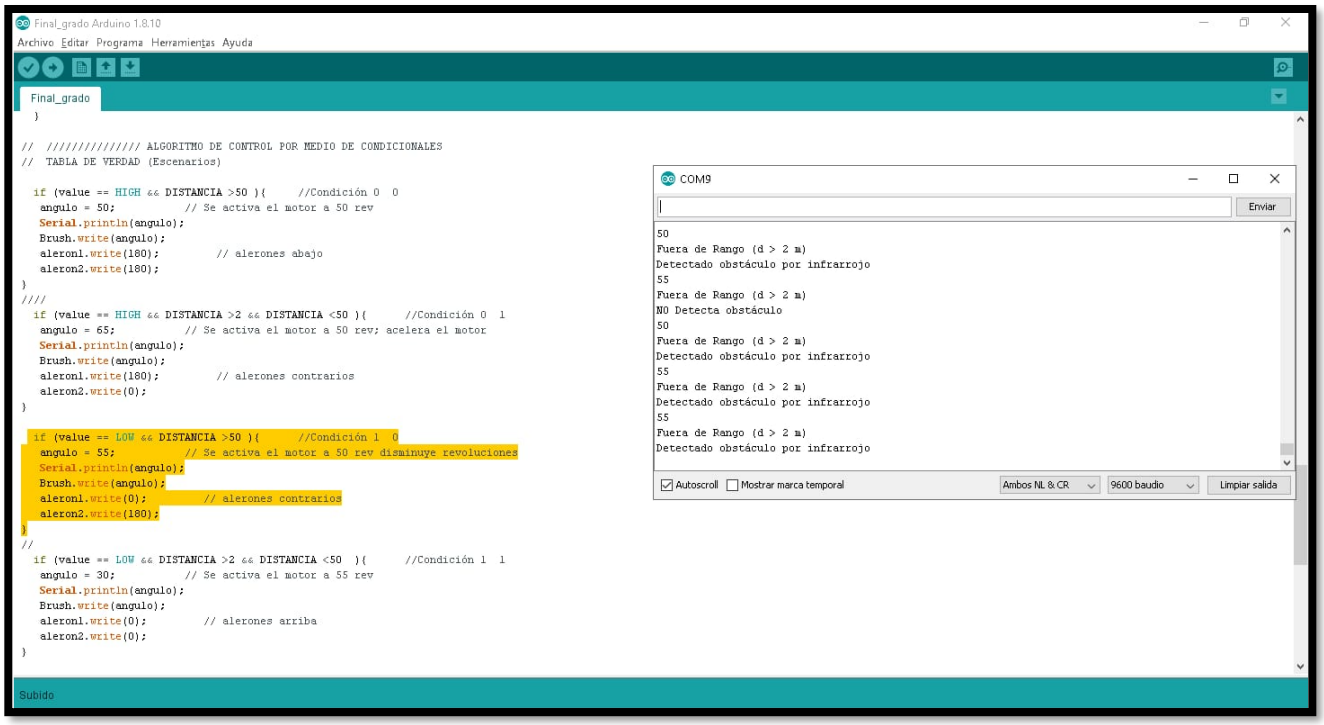


Figura 58. Condición 3 (1:0). *Elaboración propia*

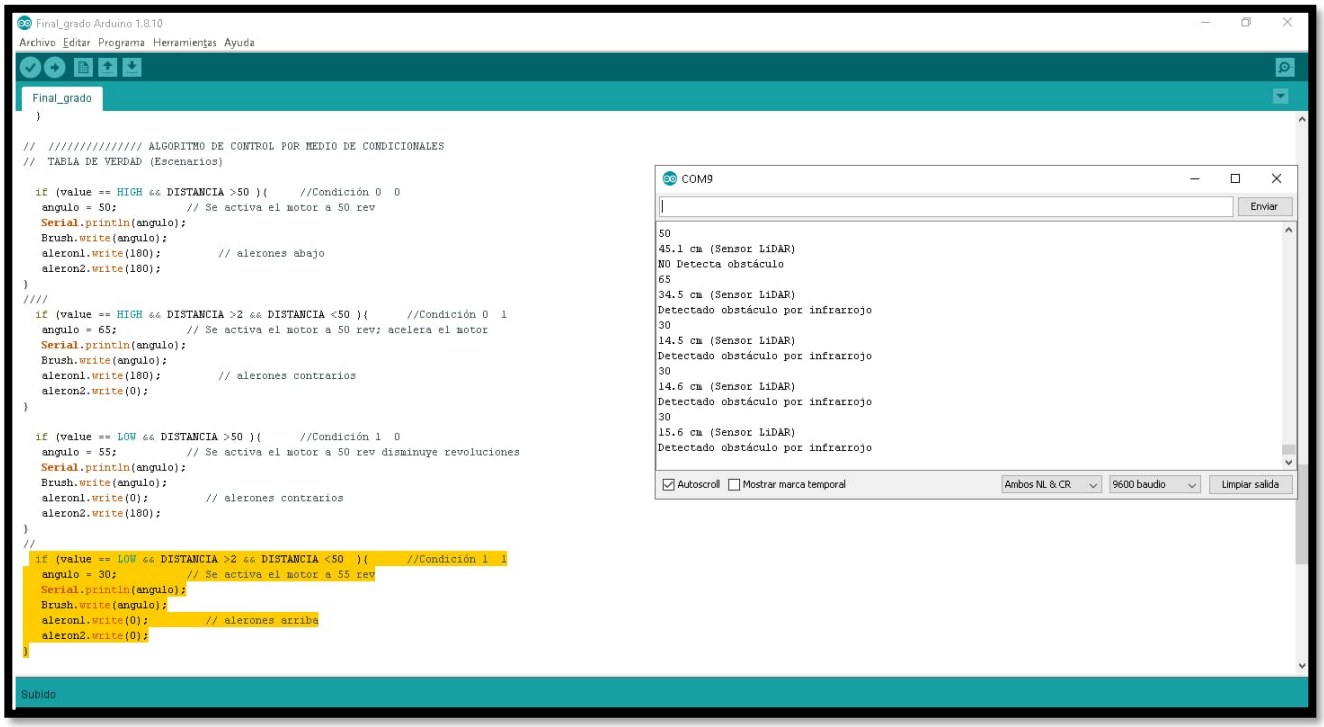


Figura 59. Condición 4 (1:1). *Elaboración propia*

Figura 60. Condición 1 (0:0)
Elaboración propia

De acuerdo a la evidencia dada por la figura 60 se confirma el funcionamiento del primer escenario en el montaje, con el sistema de detección completo visualizado por la cámara. En la figura 61 se puede confirmar el comportamiento del segundo escenario; al igual que en la imagen de la figura 62 con el tercer escenario y por último la confirmación del cuarto escenario mostrado en la figura 63.

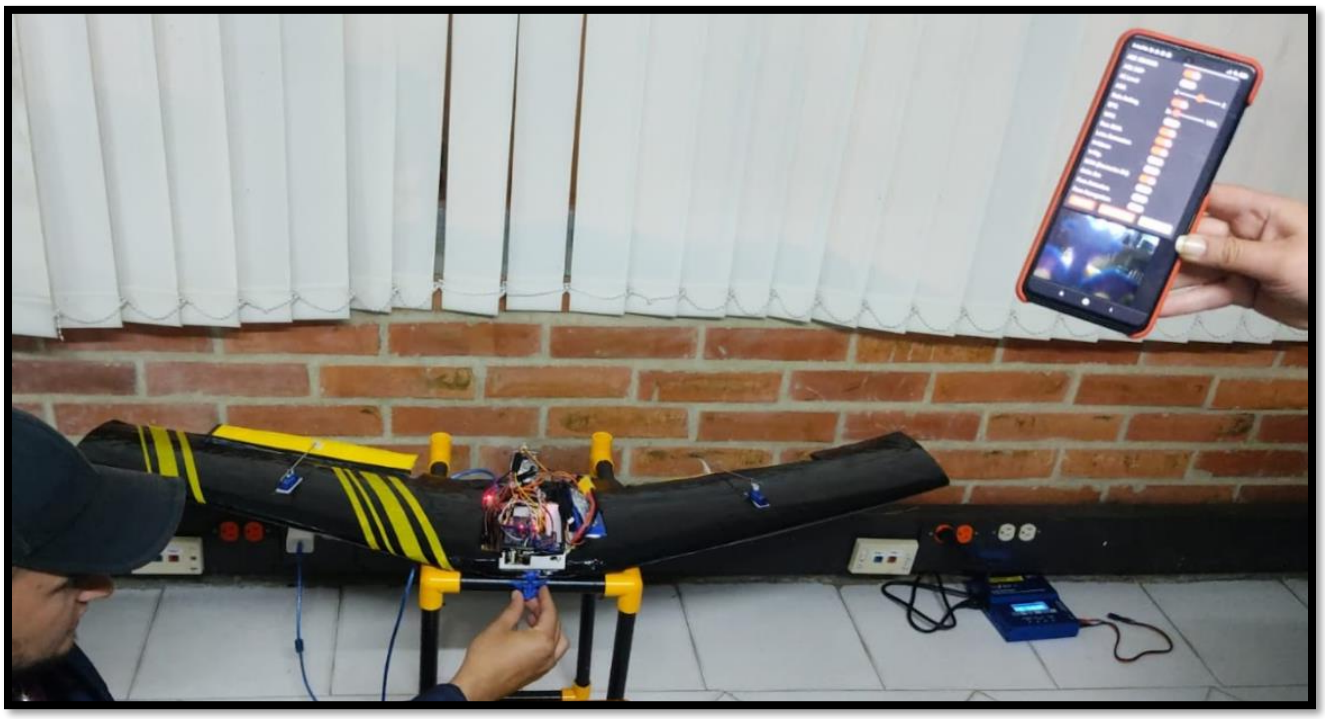


Figura 61. Condición 2 (0:1)
Elaboración propia

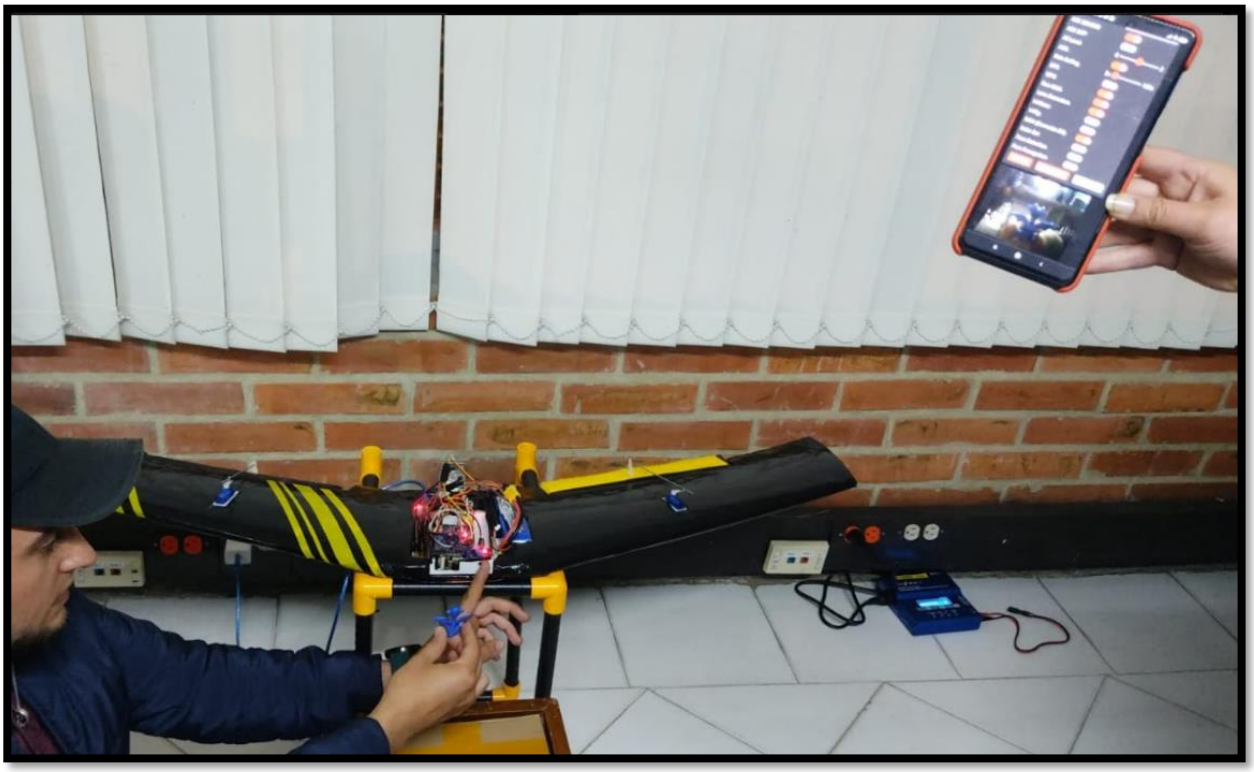


Figura 62. Condición 3 (1:0). *Elaboración propia*

Figura 63. Condición 4 (1:1). *Elaboración propia*

5.3 Rediseño

Usando un modelo ya existente de dron de ala fija y la modificación del mismo según la necesidad del proyecto, adaptada en un programa de diseño.

Se toma el diseño del dron ala fija Trimble uX5 ver figura 6, se dimensiona, y se determina que se va a ampliar las medidas tanto de las alas como del centro del dron obteniendo como resultado final lo siguiente. A partir de la figura 33, las modificaciones finales se visualizan en las siguientes imágenes en la que vemos el modelo final en el CAD mostrado en la figura 64, las dimensiones y las vistas se muestran en la figura 65.

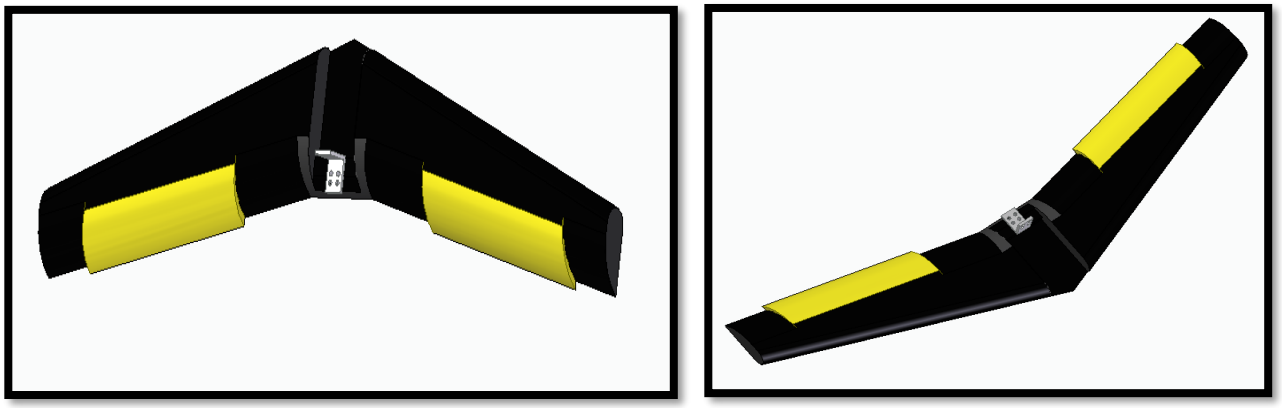


Figura 64. Diseño CAD final de Ala Zagi
Elaboración propia

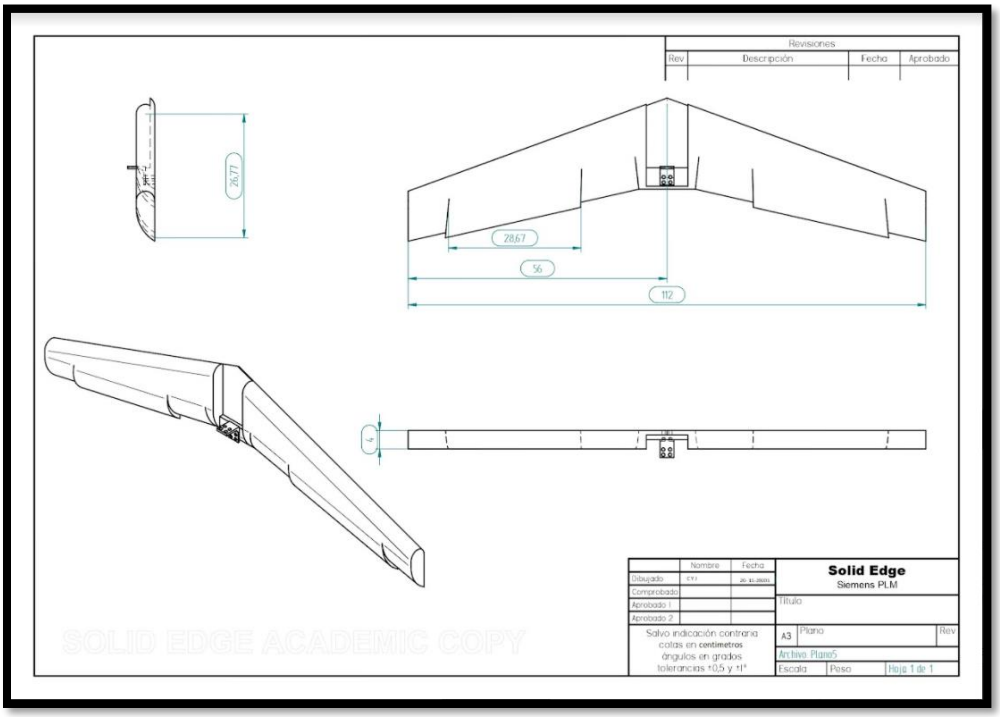


Figura 65. Plano 2D dimensiones y vistas del ala Zagi
Elaboración propia

5.4 Ensamble del sistema de detección de obstáculos y el ala zagi fabricada.

Ensamble del sistema de detección de obstáculos para la implementación del prototipo de dron de ala fija, escogido para realizar pruebas en tierra.
Se adecua todo el sistema electrónico, la batería Lipo, el ESC, el motor, en el espacio correspondiente obteniendo el siguiente resultado.

Figura 66. Dron Delta sin electrónica. *Elaboración propia*

Figura 67. Dron Delta con electrónica
Elaboración propia

Ya finalizando el ensamble del prototipo, se especifica los costos de los componentes con el valor comercial por medio online o por tienda física confirmando que es un proyecto de bajo costo, realizando la comparación con drones de ala fija que están en el mercado como los drones SenseFly que están en promedio de los 20.000 dólares.[50]

Tabla 10 Presupuesto del prototipo. *Elaboración propia*

Item	Nombre Componente	Cantidad	Valor Unidad	Valor Total
1	Sensor Infrarrojo HW-201	1	$ 6.700,00	$ 6.700,00
2	Módulo Arduino UNO	1	$ 28.000,00	$ 28.000,00
3	Módulo Arduino Mega	1	$ 54.950,49	$ 54.950,49
4	Motor Brushless EMAX XA2212	1	$ 141.923,00	$ 141.923,00
5	Regleta de pines en L	1	$ 800,00	$ 800,00
6	Regleta de pines recta	1	$ 500,00	$ 500,00
7	Controlador de Velocidad ESC EMAX 30ª	1	$ 70.000,00	$ 70.000,00
8	Batería Lipo Turnigy 2200 mAh	1	$ 132.900,00	$ 132.900,00
9	Cargador de Batería Lipo	1	$ 250.000,00	$ 250.000,00
10	Termo encogible (m)	2	$ 500,00	$ 1.000,00
11	Conectores protoboard (40 unidades)	1	$ 5.000,00	$ 5.000,00
12	Conectores batería	2	$ 4.000,00	$ 8.000,00
13	Cámara SP32 CAM	1	$ 44.000,00	$ 44.000,00
14	Servomotor sg90	3	$ 6.600,00	$ 19.800,00

Item	Nombre Componente	Cantidad	Valor Unidad	Valor Total
15	Multímetro	1	$ 64.000,00	$ 64.000,00
16	Atornillador estrella	1	$ 2.500,00	$ 2.500,00
17	Atornillador pala	1	$ 2.500,00	$ 2.500,00
18	Pinzas	1	$ 5.000,00	$ 5.000,00
19	Cortafríos	1	$ 5.000,00	$ 5.000,00
20	Hélice 7” u 8”	1	$ 10.500,00	$ 10.500,00
21	Batería cuadrada de 9v	1	$ 6.500,00	$ 6.500,00
22	Conectores protoboard macho – macho (40 unidades)	1	$ 5.000,00	$ 5.000,00
23	Sensor de distancia Láser VL53L0X-02	3	$ 24.000,00	$ 72.000,00
24	Contac (m)	2,5	$ 3.500,00	$ 8.750,00
25	Tubo PVC	3	$ 2.500,00	$ 7.500,00
26	Codos y uniones	8	$ 1.600,00	$ 12.800,00
27	Icopor	1	$ 10.000,00	$ 10.000,00
28	Cinta aislante negra (rollo)	8	$ 2.000,00	$ 16.000,00
29	Ferroníquel	5	$ 800,00	$ 4.000,00
30	bombillo 100 w	1	$ 2.000,00	$ 2.000,00
31	Roseta	1	$ 4.500,00	$ 4.500,00
32	Cable dúplex (m)	3	$ 1.200,00	$ 3.600,00
33	Conectores hembra y macho (tomacorriente)	1	$ 1.600,00	$ 1.600,00
34	Lata de pintura	1	$ 8.000,00	$ 8.000,00
TOTAL				**$ 1.015.323,49**

Muchos de estos componentes se disponen sin necesidad de comprarlos, haciendo el prototipo más económico.

Capítulo 6

Conclusiones y Recomendaciones

6.1. Conclusiones

- Respondiendo el objetivo general del proyecto, que a su vez hace parte inicial del objetivo general del semillero de investigación de la Universidad FULL, en donde se quiso comprobar la funcionalidad que tiene la implementación de un sistema de control autónomo de bajo costo, para detección y evasión de obstáculos en la operación de un dron de ala fija, validado mediante pruebas en tierra, siendo en términos generales viable partiendo de los recursos que se tuvieron disponibles, claro está que todo tiene la posibilidad de mejorar, tanto en componentes electrónicos, y métodos de programación más avanzados o con mejores prestaciones si se quiere llegar hacer volar el ala Delta.

- Se da respuesta el objetivo uno, donde mediante la cantidad de alimentación que se necesitó para cada uno de los sistemas, se implementa una placa (baquelita) siendo una parte positiva (5v) y una parte negativa, conectada a la alimentación de la placa del Arduino para una adecuada distribución de voltajes; al momento de conectar la batería Lipo se observó que la batería energizaba todo el montaje, sin quitar rapidez a los servos ni pérdida de latencia cómo sucedía sólo con la conexión del cable de impresora al computador facilitando así al usuario el poder ubicar el dron lejos de un ordenador siendo este alimentado de forma independiente sin afectar el tiempo de descarga de la batería.

- Concluyendo paralelamente al objetivo uno se puede decir que al momento de comprobar las características operacionales de los sensores cuando se compran al fabricante, en el momento de programar, ensamblar y energizar, los rangos de detección no corresponden a las especificaciones técnicas de los manuales, es decir, las características operacionales tanto del LiDAR como del infrarrojo cambian, en el LiDAR por ejemplo indicaban un rango de lectura de 2 m, pero en realidad el rango es de sólo 1 m dejando la precisión y cobertura a la mitad y en el infrarrojo se descalibra al conectarse en conjunto con los demás componentes reduciendo su cobertura de 15 cm a sólo 5 cm.

- Tal y como se pudo comprobar y dando respuesta al objetivo numero dos a partir de la unión de los diferentes códigos hay sistemas que no son compatibles entre sí, generando cambios en el montaje y la programación como lo sucedido con la cámara quedando finalmente con alimentación independiente, al ser energizada por el Arduino más las demás conexiones apagaba el circuito debido a un reset constante en su conexión.

- Aunque finalmente nuestra hipótesis no fue del todo cierta contemplando el objetivo

número dos, Con respecto al procesamiento de imagen, no se cumple el propósito de reconocimiento de obstáculos mediante el uso de la cámara porque no se cuenta con la información ni la capacitación necesaria sobre inteligencia artificial concluyendo que el tiempo invertido para lograrlo no fue suficiente, dejando la cámara sólo para visualizar el obstáculo detectado por los demás sensores mediante un navegador.

- Correspondiente al objetivo numero dos es propicio indicar que para llegar a los códigos de programación adecuados y así diseñar el código final se debió probar una gran variedad de los mismos, analizar cada una de las líneas, solicitar tutorías, modificándolos a raíz de lógica y extrayendo fragmentos hasta lograr una programación funcional coordinando de forma secuencial, llegando a la conclusión de que todo se basa en prueba y error.

- A causa del objetivo número tres respecto a la construcción es correcto concluir en el desarrollo de este proyecto con el propósito de hacer físicamente el prototipo del dron tipo ala delta o Zagi, surgen inconvenientes que van ligados a la parte de programación, fue ardua la tarea para lograr cumplir con los objetivos a cabalidad, sin embargo, no se logra al 100%.

- Gracias a todo lo anterior se puede ultimar que junto al objetivo número cuatro fue validada la funcionalidad que tiene la implementación de un sistema de control autónomo de bajo costo, para detección y evasión de obstáculos en la operación de un dron de ala fija, comprobando mediante pruebas en tierra que demostraron la viabilidad partiendo de los recursos que se tuvieron disponibles basando los resultados mediante porcentajes se logró un 85 % de lo establecido en los objetivos debido a la programación de la cámara la cual quedó independiente.

6.2. Recomendaciones

- Para futuras investigaciones, o de estudiantes que continúen con el hilo del semillero de investigación se sugiere y/o recomienda mejorar el tipo de tecnología implementada, dado que cada vez existen plataformas de diseño, simulación y programación con mayor alcance y que aportarían de manera positiva con el propósito final del dispositivo, debido a que el conocimiento y profundidad de manipulación, control y programación actuales fueron básicas y el tiempo no fue el suficiente para lograr una inteligencia artificial o un procesamiento de imágenes más arduo y completo hablando específicamente de la cámara, el dispositivo es funcional, sin embargo, su labor es sólo visual, se logró llegar hasta ese punto tomando ese sensor como una ayuda visual del sistema, en donde mostrará al usuario por medio de la interfaz el obstáculo que se encuentre al frente o en el rango de cobertura especificado por fabricante en tiempo real y con alimentación independiente.

- Con respecto al diseño del prototipo se sugiere tener en cuenta cálculos de peso y balance si el objetivo es hacer que vuele, adicionalmente la reconstrucción del mismo sea con materiales más resistentes mejorando la cavidad central con un monocasco traslucido en

la parte frontal para que no altere la funcionalidad de los sensores.

Bibliografía

[1] U. I. de Valencia, "RPAS, UAV y drones: ¿Cuáles son las diferencias? | VIU," 2018. https://www.universidadviu.com/es/actualidad/nuestros-expertos/rpas-uav-y-drones-cuales-son-las-diferencias (accessed Oct. 31, 2021).

[2] N. mundo BBC, "Cómo es el MQ-9 Reaper, el dron más letal y más usado por el ejército de Estados Unidos cuya información se compartió en la 'internet profunda' - BBC News Mundo," Jun. 12, 2018. https://www.bbc.com/mundo/noticias-44813793 (accessed Oct. 31, 2021).

[3] EXPANSION, "¿En qué usa 'drones' el Gobierno?," 2014.

[4] D. Rodriguez Brazón, "Tensión entre Venezuela y Colombia por drones y captura de sargentos - Venezuela - Internacional - ELTIEMPO.COM," 2021. Accessed: Oct. 31, 2021. [Online]. Available: https://www.eltiempo.com/mundo/venezuela/tension-entre-venezuela-y-colombia-por-drones-y-captura-de-sargentos-621630.

[5] SEMANA, "En video: el impresionante show con drones en la inauguración de los Juegos Olímpicos Tokio 2020," 2021.

[6] J. David, P. García, D. Armando, and S. Betancourt, "DISEÑO Y CONSTRUCCIÓN DEL TREN DE ATERRIZAJE PARA LA AERONAVE VANT SOLVENDUS," Fundación Universitaria Los Libertadores, 2017.

[7] N. Pons, "The Companion to International Humanitarian Law," *Companion to Int. Humanit. Law*, 2018, doi: 10.1163/9789004342019.

[8] J. A. Torres, "DISEÑO DE UN VEHÍCULO AÉREO CUADRIROTOR DE BAJO COSTE CON PLATAFORMA RASPBERRY PI," Universidad Politécnica de Valencia, 2021.

[9] M. R. Bernal, "Diseño y construcción de un avión no tripulado basado en sistemas y dispositivos COTS," 2013.

[10] L. Louis, "Working Principle of Arduino and Using it as a Tool for Study and Research," *Int. J. Control. Autom. Commun. Syst.*, vol. 1, no. 2, pp. 21–29, Apr. 2016, doi: 10.5121/IJCACS.2016.1203.

[11] Arduino, "¿Qué es Arduino? | Arduino.cl - Compra tu Arduino en Línea." https://arduino.cl/que-es-arduino/ (accessed Oct. 05, 2021).

[12] C. Cuerno-Rejado, L. García-Hernández, A. Sánchez-Carmona, A. Carrio, J. L. Sanchez-Lopez, and P. Campoy, "Evolución histórica de los vehículos aéreos no tripulados hasta la actualidad," *Dyna*, vol. 91, no. 3, p. 332, 2016, doi: 10.6036/7781.

[13] JUGUETECNIC, "¿Qué es un DRONE? Tipos, nombres y componentes." https://juguetecnic.com/blog/107_que-es-un-drone--tipos-nombres-y-componentes.html (accessed Oct. 11, 2021).

[14] HazTuDron, "Haz tu propio dron," 2019. http://haztudron.com/construccion2.html (accessed Nov. 20, 2021).

[15] Aerial insights, "▷ Tipos de drones: ¿Cuántos tipos de drones existen en el mercado?," *2019*. https://www.aerial-insights.co/blog/tipos-de-drones/ (accessed Oct. 06, 2021).

[16] TZagiCAD, "TZagi CAD - the project," *Blog*. http://tzagi.altervista.org/zcad.html (accessed Nov. 22, 2021).

[17] M. Dalvi, "Drone | Biblioteca de modelos CAD 3D | GrabCAD," *GrabCAD*, Jun. 06, 2021. https://grabcad.com/library/drone-497 (accessed Jul. 11, 2021).

[18] U. Fuentes Enriquez, J. A. Montero Valverde, M. Martínez Arroyo, and J. F. Gazga Portillo, "Detección de individuos en acceso principal | Phyton & OpenCV," *Ingeniantes*, vol. 2, no. 2, pp. 62–67, 2020, Accessed: Oct. 11, 2021. [Online]. Available: https://citt.itsm.edu.mx/ingeniantes/articulos/ingeniantes7no2vol2/9. Detección de Individuos Python & OpenCv.pdf.

[19] E. Khatab, A. Onsy, M. Varley, and A. Abouelfarag, "Vulnerable objects detection for autonomous driving: A review," *Integration*, vol. 78, pp. 36–48, May 2021, doi:

10.1016/J.VLSI.2021.01.002.
[20] L. Fernando and M. Rios, “Reconstrucción de Imágenes 3D a partir de un Sensor LiDAR de Bajo Costo,” Universidad de Ibagué, 2019.
[21] Geoinnova, “¿Qué es un sistema LiDAR? - Geoinnova,” *28/01/2021*. https://geoinnova.org/blog-territorio/que-es-un-sistema-lidar/ (accessed Oct. 05, 2021).
[22] F. Rojas Ramos, “Detección de objetos usando cámaras y sensores LIDAR,” BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA, Ciudad Universitaria, Puebla, 2020.
[23] G. A. Granada Marín and J. A. Valencia Garzón, “CONSTRUCCIÓN DE UNA PLATAFORMA DE LEVITACIÓN ELECTROMAGNÉTICA UTILIZANDO SENSORES INFRARROJOS,” Universidad Tecnológica de Pereira, Pereira, 2015.
[24] J. Hernandez_YP, “Tarea del PWM | Tinkercad,” 2021. https://www.tinkercad.com/things/9IhAHZFwRfr-tarea-del-pwm (accessed Dec. 13, 2021).
[25] M. Rodríguez, S. Sanz, and J. Andrés, “CONTROL DE VELOCIDAD DE MOTORES BRUSHLESS MEDIANTE MODULACIÓN PWM Tutor,” Universidad de Valladolid, 2018.
[26] E. Bonilla and P. Rodríguez, *Más allá del dilema de los métodos*, Tercera. 1995.
[27] Securitas Direct, “Sensores infrarrojos: ¿qué son y para qué se utilizan? - Protegiendo Personas,” 2018. https://protegiendopersonas.es/sensores-infrarrojos-que-son-y-para-que-se-utilizan/ (accessed Nov. 20, 2021).
[28] Keyence Corporation, “¿Qué es un sensor láser de tipo de reconocimiento de ‘posición’? | Fundamentos del sensor: Guía de sensores para fábricas clasificados por principios | KEYENCE,” 2021. https://www.keyence.com.mx/ss/products/sensor/sensorbasics/laser_location/info/ (accessed Nov. 20, 2021).
[29] BeJob, “Qué es la programación con arduino y para qué sirve – Bejob,” 2021. https://www.bejob.com/que-es-la-programacion-con-arduino-y-para-que-sirve/ (accessed Nov. 20, 2021).
[30] J. Guerra, “Arduino Mega 2560 el hermano mayor de Arduino UNO.” https://programarfacil.com/blog/arduino-blog/arduino-mega-2560/ (accessed Nov. 20, 2021).
[31] ACADEMIC, “Motor eléctrico sin escobillas.” https://es-academic.com/dic.nsf/eswiki/827787 (accessed Nov. 20, 2021).
[32] DYNAMO, “ESC 30A controlador para motor brushless - DynamoElectronics,” 2008. https://dynamoelectronics.com/tienda/esc-30a-controlador-para-motor-brushless/ (accessed Nov. 20, 2021).
[33] tdrobotica, “Batería de polímero de iones de litio 300mAh 7.4V 45-90C,” 2007. https://tienda.tdrobotica.co/baterias-y-cargadores-lipo/867-bateria-lipo-300mah-74v-45-90c.html (accessed Nov. 20, 2021).
[34] DYNAMO, “Cargador/balanceador batería Li-Ion 80W, 5A - DynamoElectronics,” 2007. https://dynamoelectronics.com/tienda/cargador-balanceador-bateria-li-ion-80w-5a/ (accessed Nov. 20, 2021).
[35] NAYLAMP MECHATRONICS, “ESP32-CAM con cámara OV2640 - ESP32 WiFi.” https://naylampmechatronics.com/espressif-esp/700-esp32-cam-con-camara-ov2640-esp32-wifi.html (accessed Nov. 20, 2021).
[36] MACTRONICA, “SERVOMOTOR MICRO SERVO SG90.” https://www.mactronica.com.co/servomotor-micro-servo-sg90 (accessed Nov. 20, 2021).
[37] F. Antonio and P. Jabib, “MANUAL DE BATERÍAS Y ACUMULADORES,” Universidad Pontificia Bolivariana, 2018.
[38] erlerobotics, “Baterias LiPo | Erle Robotics: Erle-copter.” https://erlerobotics.gitbooks.io/erle-robotics-erle-copter/content/es/safety/lipo.html (accessed Dec. 13, 2021).

[39] L. Felipe and G. Pimentel, “MONOPLAZA IMPLEMENTANDO MOTORES BRUSHLESS DC,” Universidad de San Carlos de Guatemala, 2021.
[40] “Amazon.com: RC Drone Motor sin escobillas, EMAX XA2212 820KV/980KV/1400KV 2-3S Motor sin escobillas para RC Drone Quadcopter Multi-Rotor Aircraft Accessroy Parts (820KV) : Juguetes y Juegos.” https://www.amazon.com/-/es/escobillas-Quadcopter-Multi-Rotor-Aircraft-Accessroy/dp/B07L1TX3C4?th=1 (accessed Nov. 01, 2021).
[41] S. Fitzgerald, “Uso de servomotores con Arduino | Taller de DroneBot,” 2013. https://dronebotworkshop.com/servo-motors-with-arduino/ (accessed Oct. 28, 2021).
[42] L. Llamas, “Controlar un servo con Arduino,” 2016. https://www.luisllamas.es/controlar-un-servo-con-arduino/ (accessed Oct. 28, 2021).
[43] R. Mamani, “Arma, Configura y Vuela tu Drone UAV desde cero, paso a paso | Udemy,” *Udemy*, 2018. https://www.udemy.com/course/arma-configura-y-vuela-un-drone-desde-cero-paso-a-paso/learn/lecture/10451146#overview (accessed Oct. 28, 2021).
[44] L. Llamas, “Detector de obstáculos con sensor infrarrojo y Arduino.” https://www.luisllamas.es/detectar-obstaculos-con-sensor-infrarrojo-y-arduino/ (accessed Nov. 16, 2021).
[45] Randomnerdtutorials, “Transmisión de video ESP32-CAM y reconocimiento facial con Arduino IDE | Tutoriales aleatorios de nerds,” 2019. https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/ (accessed Nov. 20, 2021).
[46] Randomnerdtutorials, “ESP32-CAM: Establecer punto de acceso (AP) usando Arduino IDE | Tutoriales aleatorios de nerds,” 2020. https://randomnerdtutorials.com/esp32-cam-access-point-ap-web-server/ (accessed Nov. 20, 2021).
[47] “Generador de perfil aerodinámico NACA de 4 dígitos (perfil aerodinámico NACA 2410).” http://airfoiltools.com/airfoil/naca4digit?MNaca4DigitForm%5Bcamber%5D=2&MNaca4DigitForm%5Bposition%5D=40&MNaca4DigitForm%5Bthick%5D=10&MNaca4DigitForm%5BnumPoints%5D=100&MNaca4DigitForm%5BcosSpace%5D=0&MNaca4DigitForm%5BcosSpace%5D=1&MNaca4DigitForm%5BcloseTe%5D=0&yt0=Plot (accessed Nov. 22, 2021).
[48] Hackerspace San Salvador, “GitHub - hackerspacesv / tutorial_protothreads: ejemplos básicos para usar Protothreads en Arduino,” 2017. https://github.com/hackerspacesv/tutorial_protothreads (accessed Nov. 07, 2021).
[49] D. Robots, “VL53L0X: Sensor de distancia que mide por la velocidad de la luz (Time-of-Fly) | Robots Didácticos,” 2019. http://robots-argentina.com.ar/didactica/vl53l0x-sensor-de-distancia-que-mide-por-la-velocidad-de-la-luz-time-of-fly/ (accessed Nov. 16, 2021).
[50] SenseFly, “senseFly eBee X - SF051000 - RMUS - Unmanned Solutions ™ - Ventas, capacitación y soporte de drones y robótica.” https://www.rmus.com/products/sensefly-ebee-x (accessed Dec. 14, 2021).

Página dejada en blanco intencionalmente.

Apéndice

En esta sección de ANEXOS y /o apéndices se agregan los códigos completos de cada uno de los sensores y componentes, se colocan sin formato, es decir, formato texto.

ANEXO 1

CÓDIGO 1 Cámara sin navegación

```
#include "esp_camera.h"         // Insertar librería del módulo y cámara
#include <WiFi.h>               // Librería de enrutamiento
#include "esp_timer.h"
#include "img_converters.h"
#include "Arduino.h"
#include "fb_gfx.h"
#include "soc/soc.h" //disable brownout problems
#include "soc/rtc_cntl_reg.h"  //disable brownout problems
#include "esp_http_server.h"

// Replace with your network credentials
const char* ssid     = "Ala_Zagi";                  // Se crea punto de acceso
const char* password = "123456789";                 // Clave de acceso

#define PART_BOUNDARY "123456789000000000000987654321"

// This project was tested with the AI Thinker Model, M5STACK PSRAM Model and M5STACK
WITHOUT PSRAM
#define CAMERA_MODEL_AI_THINKER
//#define CAMERA_MODEL_M5STACK_PSRAM
//#define CAMERA_MODEL_M5STACK_WITHOUT_PSRAM

// Not tested with this model
//#define CAMERA_MODEL_WROVER_KIT

#if defined(CAMERA_MODEL_WROVER_KIT)
  #define PWDN_GPIO_NUM    -1
  #define RESET_GPIO_NUM   -1
  #define XCLK_GPIO_NUM    21
  #define SIOD_GPIO_NUM    26
  #define SIOC_GPIO_NUM    27

  #define Y9_GPIO_NUM      35
  #define Y8_GPIO_NUM      34
  #define Y7_GPIO_NUM      39
  #define Y6_GPIO_NUM      36
  #define Y5_GPIO_NUM      19
  #define Y4_GPIO_NUM      18
  #define Y3_GPIO_NUM       5
```

```
  #define Y2_GPIO_NUM       4
  #define VSYNC_GPIO_NUM   25
  #define HREF_GPIO_NUM    23
  #define PCLK_GPIO_NUM    22

#elif defined(CAMERA_MODEL_M5STACK_PSRAM)
  #define PWDN_GPIO_NUM     -1
  #define RESET_GPIO_NUM    15
  #define XCLK_GPIO_NUM     27
  #define SIOD_GPIO_NUM     25
  #define SIOC_GPIO_NUM     23

  #define Y9_GPIO_NUM       19
  #define Y8_GPIO_NUM       36
  #define Y7_GPIO_NUM       18
  #define Y6_GPIO_NUM       39
  #define Y5_GPIO_NUM        5
  #define Y4_GPIO_NUM       34
  #define Y3_GPIO_NUM       35
  #define Y2_GPIO_NUM       32
  #define VSYNC_GPIO_NUM    22
  #define HREF_GPIO_NUM     26
  #define PCLK_GPIO_NUM     21

#elif defined(CAMERA_MODEL_M5STACK_WITHOUT_PSRAM)
  #define PWDN_GPIO_NUM     -1
  #define RESET_GPIO_NUM    15
  #define XCLK_GPIO_NUM     27
  #define SIOD_GPIO_NUM     25
  #define SIOC_GPIO_NUM     23

  #define Y9_GPIO_NUM       19
  #define Y8_GPIO_NUM       36
  #define Y7_GPIO_NUM       18
  #define Y6_GPIO_NUM       39
  #define Y5_GPIO_NUM        5
  #define Y4_GPIO_NUM       34
  #define Y3_GPIO_NUM       35
  #define Y2_GPIO_NUM       17
  #define VSYNC_GPIO_NUM    22
  #define HREF_GPIO_NUM     26
  #define PCLK_GPIO_NUM     21

#elif defined(CAMERA_MODEL_AI_THINKER)
  #define PWDN_GPIO_NUM     32
  #define RESET_GPIO_NUM    -1
  #define XCLK_GPIO_NUM      0
  #define SIOD_GPIO_NUM     26
  #define SIOC_GPIO_NUM     27

  #define Y9_GPIO_NUM       35
  #define Y8_GPIO_NUM       34
  #define Y7_GPIO_NUM       39
```

```
  #define Y6_GPIO_NUM       36
  #define Y5_GPIO_NUM       21
  #define Y4_GPIO_NUM       19
  #define Y3_GPIO_NUM       18
  #define Y2_GPIO_NUM        5
  #define VSYNC_GPIO_NUM    25
  #define HREF_GPIO_NUM     23
  #define PCLK_GPIO_NUM     22
#else
  #error "Camera model not selected"
#endif

static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-replace;boundary="
PART_BOUNDARY;
static const char* _STREAM_BOUNDARY = "\r\n--" PART_BOUNDARY "\r\n";
static const char* _STREAM_PART = "Content-Type: image/jpeg\r\nContent-Length: %u\r\n\r\n";

httpd_handle_t stream_httpd = NULL;

static esp_err_t stream_handler(httpd_req_t *req){
  camera_fb_t * fb = NULL;
  esp_err_t res = ESP_OK;
  size_t _jpg_buf_len = 0;
  uint8_t * _jpg_buf = NULL;
  char * part_buf[64];

  res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
  if(res != ESP_OK){
    return res;
  }

  while(true){
    fb = esp_camera_fb_get();
    if (!fb) {
      Serial.println("Camera capture failed");
      res = ESP_FAIL;
    } else {
      if(fb->width > 400){
        if(fb->format != PIXFORMAT_JPEG){
          bool jpeg_converted = frame2jpg(fb, 80, &_jpg_buf, &_jpg_buf_len);
          esp_camera_fb_return(fb);
          fb = NULL;
          if(!jpeg_converted){
            Serial.println("JPEG compression failed");
            res = ESP_FAIL;
          }
        } else {
          _jpg_buf_len = fb->len;
          _jpg_buf = fb->buf;
        }
      }
    }
    if(res == ESP_OK){
```

```
      size_t hlen = snprintf((char *)part_buf, 64, _STREAM_PART, _jpg_buf_len);
      res = httpd_resp_send_chunk(req, (const char *)part_buf, hlen);
    }
    if(res == ESP_OK){
      res = httpd_resp_send_chunk(req, (const char *)_jpg_buf, _jpg_buf_len);
    }
    if(res == ESP_OK){
      res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY, strlen(_STREAM_BOUNDARY));
    }
    if(fb){
      esp_camera_fb_return(fb);
      fb = NULL;
      _jpg_buf = NULL;
    } else if(_jpg_buf){
      free(_jpg_buf);
      _jpg_buf = NULL;
    }
    if(res != ESP_OK){
      break;
    }
    //Serial.printf("MJPG: %uB\n",(uint32_t)(_jpg_buf_len));
  }
  return res;
}

void startCameraServer(){
  httpd_config_t config = HTTPD_DEFAULT_CONFIG();
  config.server_port = 80;

  httpd_uri_t index_uri = {
    .uri       = "/",
    .method    = HTTP_GET,
    .handler   = stream_handler,
    .user_ctx  = NULL
  };

  //Serial.printf("Starting web server on port: '%d'\n", config.server_port);
  if (httpd_start(&stream_httpd, &config) == ESP_OK) {
    httpd_register_uri_handler(stream_httpd, &index_uri);
  }
}

void setup() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); //disable brownout detector

  Serial.begin(115200);
  Serial.setDebugOutput(false);

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
```

```
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;

  if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }

  // Camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }
  // Connect to Wi-Fi network with SSID and password
  Serial.print("Setting AP (Access Point)…");
  // Remove the password parameter, if you want the AP (Access Point) to be open
  WiFi.softAP(ssid, password);

  IPAddress IP = WiFi.softAPIP();
  Serial.print("Camera Stream Ready! Connect to the ESP32 AP and go to: http://");
  Serial.println(IP);

  // Start streaming web server
  startCameraServer();
}

void loop() {
  delay(1);
}
```

ANEXO 2

CÓDIGO 2 Cámara con navegación

```
#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
//            Ensure ESP32 Wrover Module or other board with PSRAM is selected
//            Partial images will be transmitted if image exceeds buffer size
//

// Select camera model
//#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
//#define CAMERA_MODEL_ESP_EYE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
//#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
//#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
//#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM

#include "camera_pins.h"

// Se asocian las diferentes redes wifi externas a las cuales la tarjeta ESP32 CAM se va a conectar

//const char* ssid = "Redmi";
//const char* password = "186f17e7bbe2";
//const char* ssid = "Magna";
//const char* password = "gabriela7";
const char* ssid = "Redmi Note 9S";
const char* password = "12345678";

void startCameraServer();

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
```

```
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;

  // if PSRAM IC present, init with UXGA resolution and higher JPEG quality
  //                     for larger pre-allocated frame buffer.
  if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }

#if defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }

  sensor_t * s = esp_camera_sensor_get();
  // initial sensors are flipped vertically and colors are a bit saturated
  if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1); // flip it back
    s->set_brightness(s, 1); // up the brightness just a bit
    s->set_saturation(s, -2); // lower the saturation
  }
  // drop down frame size for higher initial frame rate
  s->set_framesize(s, FRAMESIZE_QVGA);

#if defined(CAMERA_MODEL_M5STACK_WIDE) ||
defined(CAMERA_MODEL_M5STACK_ESP32CAM)
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);
#endif

  WiFi.begin(ssid, password);
```

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

  startCameraServer();

  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");
}

void loop() {
  // put your main code here, to run repeatedly:
  delay(10000);
}
```

ANEXO 3

Código final PT con condicionales.

```
#include <pt.h>            // Libreria Protothreads principal


// Secciones de desarrollo

struct pt hilo1;
struct pt hilo2;
struct pt hilo3;
struct pt hilo4;
struct pt hilo5;
struct pt hilo6;


// Se agregan las librerías correspondientes a los componentes

#include <Servo.h>          // Librería para motores
//#include <infrarrojo.h>     // Libreria Infrarrojo
#include <Wire.h>           // Librería de conexiones
#include <VL53L0X.h>        // Libreria Sensor Láser

Servo servo_11;             // Variables declaradas
VL53L0X sensor;
Servo aleron1;
Servo aleron2;
Servo Brush;                // Se nombra el motor

int pos = 0;                // Posición inicial del servomotor del LiDAR
const int led = 9;          // pin salida arduino (infrarrojo)
int VALOR;                  // Variable que recibe el dato
int led_estado =0;          // Variable de estado (infrarrojo)
int angulo =40;             // posición inicial motor brushless
int x;
int y;


void setup() {              // Inicialización de cada hilo o sección

// control de decisiones "Condicionales para los escenarios"

if ( x ==0 && y <=130 || y >=1300){    // caso 0  0
   angulo = angulo+10;                 // Se activa el motor a 55 rev
   aleron1.write(180);                 // alerones abajo
   aleron2.write(180);
}

if ( x ==0 && y >=130 || y <=1300){    // caso 0  1
  angulo = angulo+20;                  // Se acelera el motor 75 rev
  aleron1.write(0);                    // Posición abajo
```

```
  aleron2.write(180);            // Posición arriba
}

if (x ==1 && y <=130 || y >=1300){  // caso 1  0
  angulo = angulo-10;               // Se desacelera el motor 65 rev
  aleron1.write(180);              // Posición arriba
  aleron2.write(0);                // Posición abajo
}

if (x ==1 && y >=130 || y <=1300){      // caso 1  1
  angulo = angulo-15;                  // Se desacelera el motor 50 rev "se apaga el motor"
  aleron1.write(0);                   // alerones arriba
  aleron2.write(0);
}


  PT_INIT(&hilo1);
  PT_INIT(&hilo2);
  PT_INIT(&hilo3);
  PT_INIT(&hilo4);
  PT_INIT(&hilo5);
  PT_INIT(&hilo6);

}

void loop() {
  parpadeo(&hilo1);        // Ejemplo
  ServoLidar(&hilo2);
  SensorLidar(&hilo3);
  Infrarrojo(&hilo4);
  Alerones(&hilo5);
  Brushless(&hilo6);

}

void parpadeo(struct pt *pt) {
  PT_BEGIN(pt);
  // void setup() {
  static long t = 0;
  pinMode(13, OUTPUT);
  // }

  do {
  // void loop() {
    digitalWrite(13, HIGH);
    t = millis();
    PT_WAIT_WHILE(pt, (millis()-t)<1000);

    digitalWrite(13, LOW);
    t = millis();
    PT_WAIT_UNTIL(pt, (millis()-t)>=1000);
  // }
  } while(true);
```

```
  PT_END(pt);
}

void ServoLidar(struct pt *pt) {        // Código servomotor LiDAR
  PT_BEGIN(pt);
  // void setup() {
  static long t = 0;
  servo_11.attach(11, 500, 2500);
  // }

  do {
  // void loop() {

  // Recorrer el movimiento del servo desde 0° a 90°en pasos
  // de 1°
  for (pos = 0; pos <= 90; pos += 1) {
    // indicar al servo que vaya a la posición en la variable 'pos'
    servo_11.write(pos);
    // espera 15 ms para que el servo alcance la posición
    //delay(15); // espera 15 ms
    t = millis();
    PT_WAIT_UNTIL(pt, (millis()-t)>=15);
  }
  for (pos = 90; pos >= 0; pos -= 1) {
    servo_11.write(pos);
    //delay(15);
    t = millis();
    PT_WAIT_UNTIL(pt, (millis()-t)>=15);
  }

    t = millis();
    PT_WAIT_UNTIL(pt, (millis()-t)>=1000);
  // }
  } while(true);
  PT_END(pt);
}

void SensorLidar(struct pt *pt) {          // Inicio código Sensor LiDAR
  PT_BEGIN(pt);
  // void setup() {
  static long t = 0;
  Serial.begin(9600);
  Wire.begin();
  sensor.init();
  sensor.setTimeout(500);
  sensor.startContinuous();

  do {
  // void loop() {

//// codigo sensor
Serial.println("\n Distancia detectada por el Lidar en mm:" );
Serial.println(sensor.readRangeContinuousMillimeters());
```

```
y = Serial.println(sensor.readRangeContinuousMillimeters());
  if (sensor.timeoutOccurred()) { Serial.print(" TIMEOUT"); }

    t = millis();
    PT_WAIT_UNTIL(pt, (millis()-t)>=1000);
  // }
  } while(true);
  PT_END(pt);
}

void Infrarrojo(struct pt *pt) {    // Inicio código Sensor infrarrojo
  PT_BEGIN(pt);
  // void setup() {
  static long t = 0;
Serial.begin(9600);               // Velocidad del monitor serial
pinMode(led,INPUT);               // Led indicador

  do {
  // void loop() {

led_estado = digitalRead(led);        //lectura digital de pin
if(led_estado == HIGH)               // Etapa de comparación para activar el led según el estado del sensor
{
  //digitalWrite(led,HIGH);
  Serial.print("\n Infrarrojo no detecta ");     // Se imprime mensaje
  x = 0;
}
else
{
  Serial.print("\n Infrarrojo detecta ");       //Prende el led si se detecta obstáculo y se imprime mensaje
  x = 1;
}
 t = millis();
    PT_WAIT_UNTIL(pt, (millis()-t)>=100);
//delay(100);                       // Se genera un tiempo para leer y visualizar en el monitor serial

     t = millis();
    PT_WAIT_UNTIL(pt, (millis()-t)>=1000);
  // }
    } while(true);
  PT_END(pt);
}

void Alerones(struct pt *pt) {           // Código alerones
  PT_BEGIN(pt);
  // void setup() {
  static long t = 0;
  aleron1.attach(8);                   // pines arduino
  aleron2.attach(10);
  // }

  do {
  // void loop() {
```

```
  aleron1.write(0);              // Posición abajo
  aleron2.write(180);            // Posición arriba
  t = millis();
    PT_WAIT_WHILE(pt, (millis()-t)<2000);
  //delay(2000);                 // Una espera de 2 seg
  // Cambio de posición de los servos
  aleron1.write(180);            // Posición arriba
  aleron2.write(0);              // Posición abajo
  t = millis();
    PT_WAIT_WHILE(pt, (millis()-t)<2000);
  //delay(2000);

    t = millis();
    PT_WAIT_UNTIL(pt, (millis()-t)>=1000);
  // }
  } while(true);
  PT_END(pt);
}

void Brushless(struct pt *pt) {     // Código motor Brushless
  PT_BEGIN(pt);
  // void setup() {
  static long t = 0;
Brush.attach(4);                  // Se asigna puerto de salida del arduino
Serial.begin(9600);
Brush.write(angulo);
  // }

  do {
  // void loop() {
    if(Serial.available()>0){         // condiciones
  char num = Serial.read();
  if(num == '+'){
    angulo = angulo+5;            // Se aumentan las revoluciones de 5°
  }else if (num == '-'){
    angulo = angulo-5;            // Se disminuyen las revoluciones
  }
  Serial.println(angulo);
  Brush.write(angulo);
  }

    t = millis();
    PT_WAIT_UNTIL(pt, (millis()-t)>=1000);
  // }
  } while(true);
  PT_END(pt);
}
```

ANEXO 4

Código final del sensor LiDAR.

```
#include <Wire.h>
#include <VL53L0X.h>
VL53L0X sensor;

int n_Samples = 5; // número de muestras para promediar

//#define LONG_RANGE // Aumenta sensibilidad, +rango, -precision
//#define HIGH_SPEED // Mayor velocidad, menor precision
#define HIGH_ACCURACY // Alta precision, menor velocidad


void setup() {

 Serial.begin(9600);
 Wire.begin();

 sensor.init();
 sensor.setTimeout(500);

//###### Parametros Medida simple ##########
#if defined LONG_RANGE
// Limite de la tasa de retorno (por defecto 0.25 MCPS)
 sensor.setSignalRateLimit(0.1);
// Periodo de pulso laser (por defecto 14 y 10 PCLKs)
 sensor.setVcselPulsePeriod(VL53L0X::VcselPeriodPreRange, 18);
 sensor.setVcselPulsePeriod(VL53L0X::VcselPeriodFinalRange, 14);
#endif

#if defined HIGH_SPEED
// reduce tiempo estimado a 20 ms (por defecto 33 ms)
 sensor.setMeasurementTimingBudget(20000);
#elif defined HIGH_ACCURACY
// incrementa tiempo estimado a 200 ms
 sensor.setMeasurementTimingBudget(200000);
#endif
}


void loop() {

 float DISTANCIA = getDISTANCIA (n_Samples);

 if (sensor.timeoutOccurred()) {
 Serial.println(" TIME OUT");
 }else {
 if (DISTANCIA< 2) Serial.println("Fuera de Rango (d < 2 cm)");
 else if (DISTANCIA>220) Serial.println("Fuera de Rango (d > 2 m)");
```

```
 else {
 Serial.print(DISTANCIA, 1); // distancia en cm y 1 decimal
 Serial.println(" cm");
 }
 }
 delay (300);
}


float getDISTANCIA(int n) { // hacemos "n" mediciones

 float SUMA_n = 0;
 for (int i = 0; i < n; i++) {
 SUMA_n += sensor.readRangeSingleMillimeters();
 }
 return( SUMA_n /n /10); // Promedio en centimetros
}
```

ANEXO 5

Código diseñado, mediante la unión de la programación de cada componente implementando un sistema de control con condicionales.

CÓDIGO FINAL

```
// Inicio código, se incluyen las librerías correspondientes a los dispositivos

#include <Wire.h>
#include <VL53L0X.h>              // Libreria Sensor LiDAR
#include <Servo.h>                // Librería motores

// Se declaran las variables con respecto a las librerias

VL53L0X sensor;                   // LiDAR
Servo servo_11;                   // Servomotor del LiDAR
Servo aleron1;                    // Aleron derecho
Servo aleron2;                    // Aleron izquierdo
Servo Brush;                      // Motor brushless

// Se declaran las variables enteras y constantes

int n_Samples = 5;                // numero de muestras para promediar
const int sensorPin = 9;          // posición infrarrojo pin Arduino
int pos = 0;                      // Posición inicial servo LiDAR
int angulo=50;                    // posición inicial motor brushless


//#define LONG_RANGE              // Aumenta sensibilidad, +rango, -precisión
#define HIGH_SPEED                // Mayor velocidad, menor precisión
//#define HIGH_ACCURACY           // Alta precisión, menor velocidad


void setup() {

 Serial.begin(9600);              // Se inicia monitor serial
 Wire.begin();

 servo_11.attach(11, 500, 2500);  // pin Arduino
 sensor.init();
 sensor.setTimeout(500);
 aleron1.attach(8);               // pin Arduino
 aleron2.attach(10);              // pin Arduino

 pinMode(sensorPin , INPUT);      //definir pin como entrada

//###### Parametros Medida simple ##########

#if defined LONG_RANGE
// Limite de la tasa de retorno (por defecto 0.25 MCPS)
 sensor.setSignalRateLimit(0.1);
```

```
// Periodo de pulso laser (por defecto 14 y 10 PCLKs)
 sensor.setVcselPulsePeriod(VL53L0X::VcselPeriodPreRange, 18);
 sensor.setVcselPulsePeriod(VL53L0X::VcselPeriodFinalRange, 14);
#endif

#if defined HIGH_SPEED
// reduce tiempo estimado a 20 ms (por defecto 33 ms)
 sensor.setMeasurementTimingBudget(20000);
#elif defined HIGH_ACCURACY
// incrementa tiempo estimado a 200 ms
 sensor.setMeasurementTimingBudget(200000);
#endif

//////// BRUSH

Brush.attach(4);         // Se asigna puerto de salida del arduino
Brush.write(angulo);

}


void loop() {

 ////////////// LiDAR

 float DISTANCIA = getDISTANCIA (n_Samples);

 if (sensor.timeoutOccurred()) {
 Serial.println(" TIME OUT");
 }else {
 if (DISTANCIA< 2) Serial.println("Fuera de Rango (d < 2 cm)");
 else if (DISTANCIA>50) Serial.println("Fuera de Rango (d > 2 m)");     // original 220
 else {
 Serial.print(DISTANCIA, 1); // distancia en cm y 1 decimal
 Serial.println(" cm (Sensor LiDAR)");

 }
 }
 delay (100);

 ////////////// INFRARROJO

 int value = 0;
  value = digitalRead(sensorPin );  //lectura digital de pin

  if (value == LOW) {
     Serial.println("Detectado obstáculo por infrarrojo");
  }
  if (value == HIGH) {
     Serial.println("NO Detecta obstáculo ");
  }
  delay(100);
```

```
//  ////////////// SERVO LiDAR
//
  // Recorrer el movimiento del servo desde 0° a 80°en pasos
  // de 1°
  for (pos = 0; pos <= 80; pos += 1) {
    // indicar al servo que vaya a la posición en la variable 'pos'
    servo_11.write(pos);
    // espera 15 ms para que el servo alcance la posición
    delay(15); // espera 15 ms
  }
  for (pos = 80; pos >= 0; pos -= 1) {
    servo_11.write(pos);
    delay(15);
  }

  if(Serial.available()>0){         // condiciones
  char num = Serial.read();
  if(num == '+'){
    angulo = angulo+5;       // Se aumentan las revoluciones de 5°
  }else if (num == '-'){
    angulo = angulo-5;       // Se disminuyen las revoluciones
  }
  Serial.println(angulo);
  Brush.write(angulo);
  }

//  /////////////// ALGORITMO DE CONTROL POR MEDIO DE CONDICIONALES
//  TABLA DE VERDAD (Escenarios)

  if (value == HIGH && DISTANCIA >50 ){     //Condición 0  0
    angulo = 50;           // Se activa el motor a 50 rev
    Serial.println(angulo);
    Brush.write(angulo);
    aleron1.write(180);         // alerones abajo
    aleron2.write(180);
}
////
  if (value == HIGH && DISTANCIA >2 && DISTANCIA <50 ){      //Condición 0  1
    angulo = 65;           // Se activa el motor a 50 rev; acelera el motor
    Serial.println(angulo);
    Brush.write(angulo);
    aleron1.write(180);         // alerones contrarios
    aleron2.write(0);
}

  if (value == LOW && DISTANCIA >50 ){      //Condición 1  0
    angulo = 55;           // Se activa el motor a 50 rev disminuye revoluciones
    Serial.println(angulo);
    Brush.write(angulo);
    aleron1.write(0);         // alerones contrarios
    aleron2.write(180);
}
//
```

```
  if (value == LOW && DISTANCIA >2 && DISTANCIA <50  ){      //Condición 1  1
   angulo = 30;           // Se activa el motor a 55 rev
   Serial.println(angulo);
   Brush.write(angulo);
   aleron1.write(0);         // alerones arriba
   aleron2.write(0);
}


//  ////////////////// ALERONES: Se le asigna la posición inicial a los servos (0°)
//  aleron1.write(0);    // Posición arriba
//  aleron2.write(180);  // Posición abajo
//  delay(2000);       // Una espera de 2 seg
//  // Cambio de posición de los servos
//  aleron1.write(180);   // Posición abajo
//  aleron2.write(0);     // Posición arriba
//  delay(1000);


////////// BRUSH

  if(Serial.available()>0){         // condiciones
  char num = Serial.read();
  if(num == '+'){
    angulo = angulo+5;       // Se aumentan las revoluciones de 5°
  }else if (num == '-'){
    angulo = angulo-5;       // Se disminuyen las revoluciones
  }
  Serial.println(angulo);
  Brush.write(angulo);
  }
}


float getDISTANCIA(int n) { // hacemos "n" mediciones

 float SUMA_n = 0;
 for (int i = 0; i < n; i++) {
 SUMA_n += sensor.readRangeSingleMillimeters();
 }
 return( SUMA_n /n /10); // Promedio en centimetros
}
```

Página dejada en blanco intencionalmente.