



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA DE MECATRÓNICA

**IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL Y PLANIFICACIÓN DE
VUELO AUTOMÁTICO PARA EL DRON DJI TELLO**

Trabajo de titulación previo a la obtención del

Título de Ingenieros en Mecatrónica

AUTORES:

JOSSELYN ESTEFANIA MOREIRA RIOS

FÉLIX LEONARDO JARRÍN GORDÓN

TUTOR:

JAIME ROLANDO HEREDIA VELASTEGUÍ

Quito – Ecuador

2023

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, **Félix Leonardo Jarrín Gordón** con documento de identificación N° 1751447366 y **Josselyn Estefania Moreira Rios** con documento de identificación N° 1725409179 manifestamos que:

Somos los autores y responsables del presente trabajo; y autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 13 de marzo del año 2023.

Atentamente,

Josselyn Estefania Moreira Rios

1725409179

Félix Leonardo Jarrín Gordón

1751447366

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO
DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, **Félix Leonardo Jarrín Gordón** con documento de identificación N° 1751447366 y **Josselyn Estefania Moreira Rios** con documento de identificación N° 1725409179, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos los autores de un Dispositivo tecnológico: “Implementación de un sistema de control y planificación de vuelo automático para el dron DJI Tello”, el cual ha sido desarrollado para optar por el título de: Ingenieros en Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.
Quito, 13 de marzo del año 2023.

Atentamente,

Josselyn Estefania Moreira Rios

1725409179

Félix Leonardo Jarrín Gordón

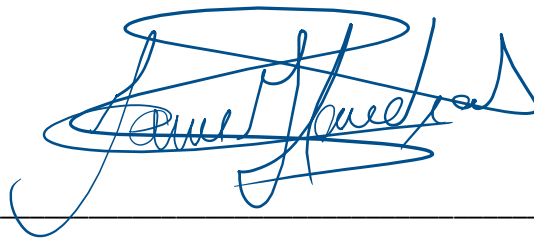
1751447366

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **Jaime Rolando Heredia Velasteguí** con documento de identificación N° 1721400768, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL Y PLANIFICACIÓN DE VUELO AUTOMÁTICO PARA EL DRON DJI TELLO, realizado por **Félix Leonardo Jarrín Gordón** con documento de identificación N° 1751447366 y **Josselyn Estefania Moreira Rios** con documento de identificación N° 1725409179, obteniendo como resultado final el trabajo de titulación bajo la opción de un Dispositivo tecnológico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 13 de marzo del año 2023

Atentamente,



Ing. Jaime Rolando Heredia Velasteguí, MSc.

1721400768

AGRADECIMIENTO

En primer lugar, quiero agradecer a mi abuelita Clemencia, aunque ya no está conmigo ahora, quisiera agradecerle por todo el amor, el apoyo y la sabiduría que me brindó a lo largo de los años. Estoy muy agradecido por todos los consejos y experiencias que me compartió a lo largo de su vida, las cuales me han ayudado a convertirme en la persona que soy.

Quiero agradecer a mis padres, por el apoyo incondicional que me han dado durante mi proceso de formación de mi vida profesional, animándome a perseverar, ser constante y a creer en mí, siempre recordándome que no hay límites para el éxito.

Además, agradecer a todos los mis profesores que me han ayudado durante el proceso y desarrollo de este trabajo de investigación. En particular, a mi profesor guía, quien me supo orientar para alcanzar el objetivo de esta tesis.

Félix Leonardo Jarrín Gordón

AGRADECIMIENTO

Quiero agradecer muy especialmente a mi madre Diana, por tu amor incondicional y por todo el apoyo que me diste durante mi carrera. Estoy muy agradecida por todo lo que has hecho por mí para ayudarme a alcanzar esta meta.

A mi mejor amiga Belén, agradezco sinceramente todo tu apoyo durante mi tesis. Desde el inicio, te has portado como una verdadera amiga, estando siempre ahí para escuchar, ofrecer consejos y motivarme. Me has dado la oportunidad de encontrar el equilibrio entre el trabajo y la vida, me has recordado que los resultados no vienen de la noche a la mañana, y me has dado la fuerza para seguir adelante cuando la tarea parecía imposible.

Quiero dar las gracias a mi compañero de tesis Félix, por todo el trabajo duro y el apoyo incondicional que me ha ofrecido a lo largo de este trabajo. De no haber sido por su ayuda, no habría podido superar con éxito este proyecto. Estoy muy agradecida por haber estado allí para mí cuando lo necesitaba, y por aportar su conocimiento y opinión al proyecto. Gracias de nuevo por tu ayuda.

Por último, a mi gato, Mijo, quiero agradecerte por toda la compañía que me brindaste durante el tiempo que estuviste conmigo, por cuidarme cuando me quedaba hasta altas horas de la madrugada y siempre demostrarme tu amor incondicional.

Josselyn Estefania Moreira Rios

ÍNDICE

| | |
|--|------------|
| RESUMEN..... | I |
| ABSTRACT..... | II |
| PROBLEMA DE ESTUDIO..... | III |
| JUSTIFICACIÓN..... | IV |
| INTRODUCCIÓN..... | V |
| OBJETIVOS..... | VI |
| Objetivo general..... | VI |
| Objetivos específicos..... | VI |
| METODOLOGÍA..... | VII |
| Fase de investigación..... | VII |
| Fase de programación..... | VII |
| Fase de diseño y construcción..... | VII |
| Fase de pruebas..... | VII |
| CAPÍTULO 1..... | 1 |
| 1. MARCO TEÓRICO..... | 1 |
| 1.1. Dron DJI Tello..... | 1 |
| 1.2. Características técnicas del dron..... | 1 |
| 1.3. Tipos de control de vuelo..... | 2 |
| 1.3.1. Control de vuelo mediante control remoto, o rutas de vuelo..... | 3 |

| | |
|---|-----------|
| 1.3.2. Control de vuelo autónomo o automático..... | 3 |
| 1.3.3. Control de vuelo por aplicación móvil. | 4 |
| 1.4. Hélice del dron | 5 |
| 1.4.1. Ángulo de ataque de la hélice..... | 6 |
| 1.4.2. Perfil aerodinámico..... | 6 |
| 1.4.3. Longitud y paso de la hélice | 7 |
| 1.5. Transmisión de videos sobre redes de datos | 7 |
| 1.6. Protocolos de comunicación mediante redes inalámbricas | 8 |
| 1.7. Tele operación | 9 |
| 1.8. Redes neuronales..... | 9 |
| 1.9. Realidad aumentada | 10 |
| 1.10. Software implementado..... | 10 |
| 1.10.1. Python..... | 11 |
| 1.10.2. Google Colab..... | 11 |
| 1.10.3. YoloV5 | 11 |
| CAPÍTULO 2 | 13 |
| 2. ANÁLISIS DE ALTERNATIVAS | 13 |
| 2.1. Diferentes tipos de planificadores de ruta utilizando drones. | 13 |
| 2.1.1. Alternativa 1: Planificación automática de rutas para un dron no tripulado. | 13 |
| 2.1.2. Alternativa 2: Planificación automática de rutas para un dron no tripulado con dinámica de vuelo restringida..... | 13 |

| | |
|---|----|
| 2.1.3. Alternativa 3: Drones como servicio: un análisis basado en simulación para la toma de decisiones en drones. | 14 |
| 2.1.4. Alternativa 4: Planificación de vuelo de drones para operaciones urbanas seguras mediante control remoto. | 14 |
| 2.2. Identificación de necesidades. | 14 |
| 2.2.1. Jerarquización de las necesidades. | 15 |
| 2.2.2. Encuestas para establecer la importancia relativa de las necesidades. | 16 |
| 2.2.3. Comparación de las alternativas según las necesidades. | 16 |
| 2.2.4 Resultados de las alternativas de trabajo. | 17 |
| 2.3 IDEs de programación | 18 |
| 2.3.1. Encuestas para establecer la importancia relativa de las necesidades que buscan los usuarios en los IDEs de programación. | 18 |
| 2.3.2. PyCharm | 19 |
| 2.3.3. Jupyter Lab. | 19 |
| 2.3.4. Spyder | 19 |
| 2.3.4. Comparación de los IDEs según las necesidades | 20 |
| 2.3.5. Resultados de las alternativas de IDEs. | 21 |
| 2.4 Sistema de implementación. | 21 |
| 2.4.1. Encuestas para establecer la importancia relativa de las necesidades que buscan los usuarios en los Software. | 22 |
| 2.4.2. YoloV5. | 22 |
| 2.4.3. TensorFlow. | 23 |

| | |
|---|-----------|
| 2.4.4. Haar Cascade. | 23 |
| 2.4.4. Comparación de los Sistemas de implementación según las necesidades..... | 24 |
| 2.4.5. Resultados de las alternativas de softwares de implementación. | 25 |
| CAPÍTULO 3 | 27 |
| 3. DISEÑO E IMPLEMENTACIÓN | 27 |
| 3.1. Diseño de la hélice en Solid Works. | 27 |
| 3.1.1. Aproximación fotográfica..... | 27 |
| 3.1.2. Aproximación por medición. | 28 |
| 3.1.3. Diseño de la hélice..... | 28 |
| 3.1.4. Diseño del soporte de la hélice en SolidWorks. | 29 |
| 3.1.5. Implementación del área de simulación en SolidWorks. | 30 |
| 3.2. Control del dron mediante phyton..... | 31 |
| 3.2.1. Control mediante el computador. | 31 |
| 3.2.1.1. Librería DJI Tello. | 31 |
| 3.3. Realidad aumentada. | 31 |
| 3.3.1. Marcadores aruco. | 32 |
| 3.3.1.1. Generador online de marcadores aruco..... | 32 |
| 3.4. Detector de objetos..... | 33 |
| 3.4.1. Hardware y software utilizados para el entrenamiento de la red neuronal..... | 33 |
| 3.4.2. YoloV5. | 33 |
| 3.4.2.1. Crear el Dataset..... | 34 |

| | |
|--|-----------|
| 3.4.2.1.1. Creación de imágenes..... | 35 |
| 3.4.2.1.2. Creación de etiquetas. | 36 |
| 3.4.2.2. Batch size..... | 36 |
| 3.4.2.3. Épocas..... | 36 |
| 3.4.2.4. Inferencia. | 37 |
| 3.4.3. Detección del dron mediante cuadrículas..... | 37 |
| 3.4.3.1. Detección de dron mediante cuadrículas columna 1..... | 38 |
| 3.4.3.2. Detección de dron mediante cuadrículas columna 2..... | 39 |
| 3.4.3.3. Detección de dron mediante cuadrículas columna 3..... | 40 |
| 3.4.3.4. Detección de dron mediante cuadrículas columna 4..... | 41 |
| 3.4.3.5. Vista general de la trayectoria del dron dentro de las cuadrículas..... | 42 |
| 3.5. Sistema eléctrico. | 43 |
| 3.5.1. Diagrama de flujo del funcionamiento del programa de la estación. | 44 |
| CAPÍTULO 4 | 45 |
| 4. ANÁLISIS DE RESULTADOS..... | 45 |
| 4.1. Análisis de los resultados obtenidos mediante la simulación de la hélice en el programa SolidWorks..... | 45 |
| 4.1.1. Ángulo de incidencia 20° | 45 |
| 4.1.2. Ángulo de incidencia 25° | 47 |
| 4.1.3. Ángulo de incidencia 30° | 49 |
| 4.1.4. Ángulo de incidencia 35° | 52 |
| 4.1.5. Ángulo de incidencia 40° | 54 |

| | |
|--|-----------|
| 4.1.6. Ángulo de incidencia 45° | 56 |
| 4.1.7. Análisis de fuerza en la simulación de la hélice. | 59 |
| 4.1.8. Análisis de velocidad en la simulación de la hélice. | 59 |
| 4.1.9 Análisis de la masa de la hélice. | 60 |
| 4.1.10. Análisis de las simulaciones en la hélice. | 60 |
| 4.2. Pruebas en la red neuronal de YoloV5. | 62 |
| 4.2.1. Batch size. | 62 |
| 4.2.1.1. Pruebas con un batch size de 16. | 62 |
| 4.2.1.2. Pruebas con un batch size de 8. | 64 |
| 4.2.3. Hardware de implementación. | 65 |
| 4.2.4. Épocas. | 65 |
| 4.3. Pruebas del dron | 66 |
| 4.3.1. Pruebas de la batería del dron en el área de trabajo asignada. | 66 |
| 5. CONCLUSIONES Y RECOMENDACIONES..... | 68 |
| 5.1. Conclusiones | 68 |
| 5.2. Recomendaciones..... | 69 |
| REFERENCIAS BIBLIOGRÁFICAS..... | 70 |
| ANEXOS..... | 74 |
| INDICE DE FIGURAS | |
| Figura 1. Dron DJI Tello..... | 1 |

| | |
|--|----|
| Figura 2. Aplicación móvil para control de dron DJI Tello..... | 5 |
| Figura 3. Hélice del dron DJI Tello. | 6 |
| Figura 4. Determinación del ángulo de incidencia. | 27 |
| Figura 5. Determinación del ángulo de incidencia. | 28 |
| Figura 6. Diseño de la hélice en SolidWorks..... | 29 |
| Figura 7. Diseño de la hélice en SolidWorks..... | 29 |
| Figura 8. Diseño del soporte de la hélice en SolidWorks | 30 |
| Figura 9. Implementación del área de simulación en SolidWorks | 30 |
| Figura 10. Generador de marcador aruco online. | 32 |
| Figura 11. Ejemplo de marcador aruco..... | 32 |
| Figura 12. Parámetros importantes al entrenar con YoloV5..... | 34 |
| Figura 13. Archivos necesarios para la creación del dataset. | 35 |
| Figura 14. Imágenes tomadas del dron | 35 |
| Figura 15. Etiquetado del dron en la región en la que se encuentra. | 36 |
| Figura 16. Cámara detectando el cuadrante donde se encuentra el dron. | 37 |
| Figura 17. Referencia de cuadrícula generada en Python..... | 38 |
| Figura 18. Detección del dron mediante cuadrículas..... | 39 |
| Figura 19. Detección del dron mediante cuadrículas..... | 40 |
| Figura 20. Detección del dron mediante cuadrículas..... | 41 |
| Figura 21. Detección del dron mediante cuadrículas..... | 42 |
| Figura 22. Detección del dron mediante cuadrículas..... | 43 |

| | |
|--|----|
| Figura 23. Diagrama eléctrico utilizado..... | 44 |
| Figura 24. Diagrama de flujo de la estación. | 44 |
| Figura 25. Hélice original del dron. | 45 |
| Figura 26. Simulación hélices con ángulo de 20°..... | 46 |
| Figura 27. Fuerza producida por la hélice. | 46 |
| Figura 28. Velocidad producida por la hélice. | 47 |
| Figura 29. Hélice impresa a un ángulo de 25° | 47 |
| Figura 30. Simulación hélices con ángulo de 25°..... | 48 |
| Figura 31. Fuerza producida por la hélice. | 48 |
| Figura 32. Velocidad producida por la hélice..... | 49 |
| Figura 33. Hélice impresa a un ángulo de 30°..... | 50 |
| Figura 34. Simulación hélices con ángulo de 30°..... | 50 |
| Figura 35. Fuerza producida por la hélice. | 51 |
| Figura 36. Velocidad producida por la hélice..... | 51 |
| Figura 37. Hélice impresa a un ángulo de 35° | 52 |
| Figura 38. Simulación hélices con ángulo de 35°..... | 52 |
| Figura 39. Fuerza producida por la hélice. | 53 |
| Figura 40. Velocidad producida por la hélice..... | 53 |
| Figura 41. Hélice impresa a un ángulo de 40° | 54 |
| Figura 42. Simulación hélices con ángulo de 40°..... | 55 |
| Figura 43. Fuerza producida por la hélice. | 55 |

| | |
|--|----|
| Figura 44. Velocidad producida por la hélice. | 56 |
| Figura 45. Hélice impresa a un ángulo de 45° | 57 |
| Figura 46. Simulación hélices con ángulo de 40° | 57 |
| Figura 47. Fuerza producida por la hélice. | 58 |
| Figura 48. Velocidad producida por la hélice. | 58 |
| Figura 49. Factores de la red neuronal | 62 |
| Figura 50. Entrenamiento con un batch size de 16 | 63 |
| Figura 51. Resultados del entrenamiento con un batch size de 16. | 63 |
| Figura 52. Entrenamiento con un batch size de 8. | 64 |
| Figura 53. Resultados del entrenamiento con un batch size de 8 | 64 |
| Figura 54. Resultados del entrenamiento en Google Colab..... | 65 |
| Figura 55. Pruebas del dron con la batería por debajo del 70-60% de carga..... | 66 |
| Figura 56. Pruebas del dron con la batería 100% cargada | 67 |

INDICE DE TABLAS

| | |
|---|----|
| Tabla 1. Identificación de necesidades, Autores..... | 15 |
| Tabla 2. Grados de importancia respecto a las necesidades, Autores..... | 15 |
| Tabla 3. Necesidades y su importancia relativa, Autores. | 16 |
| Tabla 4. Análisis y calificación de las alternativas, Autores. | 17 |
| Tabla 5. Características de los IDEs y su importancia relativa, Autores. | 18 |
| Tabla 6. Análisis y calificación de los IDEs de programación, Autores. | 20 |
| Tabla 7. Características de los softwares y su importancia relativa, Autores..... | 22 |

| | |
|---|----|
| Tabla 8. Análisis y calificación de los sistemas de implementación, Autores. | 25 |
| Tabla 9. Análisis del software para la hélice de 20°. | 47 |
| Tabla 10. Análisis del software para la hélice de 25°. | 49 |
| Tabla 11. Análisis del software para la hélice de 35°. | 51 |
| Tabla 12. Análisis del software para la hélice de 35°. | 54 |
| Tabla 13. Análisis del software para la hélice de 40°. | 56 |
| Tabla 14. Análisis del software para la hélice de 45°. | 59 |
| Tabla 15. Análisis de la fuerza en la simulación de la hélice. | 59 |
| Tabla 16. Análisis de la velocidad en la simulación de la hélice. | 59 |
| Tabla 17. Análisis de la masa de la hélice con respecto al ángulo. | 60 |
| Tabla 18. Masa en la simulación de la hélice | 60 |
| Tabla 19. Velocidad en la simulación de la hélice..... | 61 |
| Tabla 20. Fuerza en la simulación de la hélice | 61 |

RESUMEN

El presente proyecto tiene como objetivo la implementación de un equipo para monitorear las estaciones de trabajo, mejorando los tiempos de producción, como también los errores que puedan ocurrir durante el proceso de producción, ya que este equipo tendrá la supervisión del técnico asignado.

En el marco teórico, se describe las características del dron que se va a utilizar, también se da un breve detalle de proyectos con similitud al que se realiza. Por último, se explica de manera sucinta los programas que se utilizan para alcanzar los objetivos planteados.

Se realiza cambios en el ángulo de sustentación de la hélice del dron, mediante ingeniería inversa y con la ayuda del programa de modelado SolidWorks para poder generar varias simulaciones de la misma y determinar el flujo del aire en relación a la velocidad y la fuerza que genera.

La implementación del proyecto se divide en dos etapas, la primera es el control mediante el computador, en donde consta el entrenamiento que se realizó para poder detectar al dron, y que el mismo pueda detectar las fallas. La segunda etapa es la parte eléctrica, en donde se explica los componentes utilizados y la forma en la que estos funcionan junto con el computador y el dron.

Por último, en las conclusiones se detallan los resultados obtenidos, y se especifican varias recomendaciones para poder tener un mejor desempeño del dron, y las mejoras que se podrían realizar al mismo.

ABSTRACT

The objective of this project is the implementation of an equipment that monitors these workstations, thus improving production times, as well as the errors that may occur during production, because it will always be in constant review by the monitoring equipment.

The development of this paper has as an opening the theoretical framework, which explains the characteristics of the drone to be used, also gives a brief detail of projects with similarity to the one we do. Finally, the programs by means of which the proposed objectives were achieved are explained in a quick way.

Changes in the angle of lift of the propeller of the drone are made by reverse engineering and with the help of the SolidWorks modeling program in order to generate several simulations of it and determine the air flow in relation to the speed and force it generates.

The implementation of the project is divided into two stages, the first one is the computer control, which consists of the training that was done to detect the drone, and that it can detect failures. The second stage is the electrical part, where the components used and how they work together with the computer and the drone are explained.

Finally, the conclusions are determined based on the results obtained, and several recommendations are specified in order to have a better performance of the drone, and the improvements that could be made to it.

PROBLEMA DE ESTUDIO

Un equipo de monitoreo continuo es un aparato sumamente importante en la industria, haciendo que la carga laboral humana se vea reducida significativamente, esto es necesario porque para una empresa no es rentable la contratación de varios trabajadores que puedan estar monitoreando cada operación que se realice, en especial de aquellas que funcionan las veinte y cuatro horas del día, hoy en día existen varios casos en los que esto ya se ha logrado automatizar por medio de software, los cuales han sido de gran ayuda para poder reducir la cantidad de carga laboral humana, el problema con este tipo de software radica en que no se puede interactuar con el equipo en vivo, en otras ocasiones logran tener un precio demasiado elevado, en ciertos casos no logran presentar todas las alertas del proceso o lo hacen demasiado tarde debido a que no hace el censado de las señales en el momento preciso en el que se envían los errores.

JUSTIFICACIÓN

La solución que se da a este problema es la implementación de un equipo que monitoree dichas estaciones de trabajo, mejorando así los tiempos de producción, como también los errores que puedan ocurrir durante la misma, debido a que siempre estará en constante revisión por parte del equipo de monitoreo. Existen varias opciones para realizar un equipo de monitoreo, pero el más asequible es el uso de un dron, este al ser volador, tiene facilidad de movimiento por toda el área de trabajo designada, así como el de desplazamiento en los diferentes ejes de coordenadas. Con una buena optimización de la hélice, se podría llegar a conseguir varias mejoras en la aerodinámica del dron, corrigiendo los diferentes fallos que se den en sus desplazamientos, también mejoraría su autonomía y capacidad de carga, para que el monitoreo se lleve a cabo de una manera óptima.

INTRODUCCIÓN

El presente proyecto de titulación, tiene como objetivo implementar un sistema de planificación de vuelo automático para el dron DJI Tello, para comunicar al usuario si la maquina está presentando fallos.

Existen varias opciones para realizar un equipo de monitoreo, pero el más asequible es el uso de un dron, este al ser volador, tiene facilidad de movimiento por toda el área de trabajo designada, así como el de desplazamiento en los diferentes ejes de coordenadas. Con una buena optimización de la hélice, se podría llegar a conseguir varias mejoras en la aerodinámica del dron, corrigiendo los diferentes fallos que se den en sus desplazamientos, también mejoraría su autonomía y capacidad de carga, para que el monitoreo se lleve a cabo de una manera óptima.

Dado que un equipo de monitoreo continuo es un aparato sumamente importante en la industria, haciendo que la carga laboral humana se vea reducida significativamente, esto es necesario porque para una empresa no es rentable la contratación de varios trabajadores que puedan estar monitoreando cada operación que se realice.

Es relevante este proyecto porque favorecerá los tiempos de producción de las empresas, además menorará los errores en las maquinas, ya que avisará con antelación para que la persona encargada pueda solucionarlo en el menor tiempo posible. Para su desarrollo, se tomará en cuenta el espacio de trabajo, ya que el dron primero debe hacer un estudio del lugar en el que se encuentra y así poder acercarse a visualizar los posibles fallos que se ve en cada estación de trabajo, por medio de la realidad aumentada. La ventaja de realizar este proyecto, es que, al ser un dron con cámara incorporada se puede interactuar en vivo, asimismo, el dron es de un precio accesible para todas las personas que deseen adquirirlo. Dentro del proyecto se exhiben los soportes teóricos, diseño y análisis de la hélice y entrenamiento - programación del dron.

OBJETIVOS

Objetivo general

Implementar un sistema de control y planificación de vuelo automático para el dron DJI Tello.

Objetivos específicos

- Diseñar el algoritmo para la planificación de ruta y detección de fallos utilizando el dron.
- Realizar el análisis aerodinámico del dron mediante la revisión de su hélice para obtener mejoras en el dron.
- Diseñar un algoritmo que permita generar una realidad aumentada empleando la cámara del dron para la detección de fallos.
- Analizar los resultados del algoritmo implementado mediante pruebas de funcionamiento para validar la planificación de ruta y la detección de fallos.

METODOLOGÍA

Para una mejor comprensión del proyecto se lo puede dividir en cuatro fases de trabajo, las cuales estarán detalladas a continuación:

Fase de investigación.

En esta fase, se llevará a cabo la investigación sobre el dron DJI Tello, también se realizará una comparación con otros drones capaces de realizar un plan de ruta en su vuelo, para así poder tener una comprensión profunda del tema y poder producir información nueva y valiosa para el proyecto.

Fase de programación.

Se realizará la programación del dron, por medio de Python, el cual sea capaz de producir un código fuente confiable y eficiente que implemente las funcionalidades especificadas de manera correcta.

Fase de diseño y construcción.

Se construirá una pequeña estación de trabajo del dron, la cual será la base de despegue del mismo, también un pequeño simulador de fallo de la máquina para verificar su funcionamiento. Se producirá un proyecto que cumpla con los requisitos especificados y sea funcional, seguro y eficiente.

Fase de pruebas.

Por último, se desarrollará diseños mediante SolidWorks, en el cual se realizarán varios cambios en el ángulo de la hélice, para corroborar cual sería el ideal en el vuelo y así, asegurarse de que el producto funcione de manera adecuada y satisfaga las necesidades del usuario.

CAPÍTULO 1

1. MARCO TEÓRICO

En este capítulo se detallará las características principales del dron DJI Tello y su hélice, de igual forma se dará un breve resumen de las maneras básicas en las que se puede operar a un dron, por último, se explicará de los programas a utilizar.

1.1. Dron DJI Tello

DJI Tello es un dron de bajo costo desarrollado por la compañía DJI en colaboración con Intel. Es un dron asequible y fácil de usar que ofrece características avanzadas como una cámara HD, vuelo estable y controles intuitivos. Ideal para principiantes y entusiastas de los drones [4].



Figura 1. Dron DJI Tello.

1.2. Características técnicas del dron

El dron Tello DJI, El DJI Tello es un dron de bajo costo y fácil de usar diseñado para principiantes y entusiastas. Algunas de sus características incluyen:

- Tamaño compacto y ligero: El DJI Tello es un dron ligero, siendo así fácil de llevar y maniobrar.
- Cámara HD: El dron viene equipado con una cámara HD de 5 megapíxeles que permite capturar imágenes y videos de alta calidad.

- Vuelo estable: El DJI Tello está equipado con tecnología de estabilización de imagen avanzada, lo que permite un vuelo estable y suave.
- Modo sin cabeza: Con este modo, el dron volará en la dirección que selecciones, independientemente de la dirección en la que esté apuntando [5].
- Autonomía de vuelo: El DJI Tello puede volar hasta 13 minutos con una sola carga de batería [3].
- Controles intuitivos: El DJI Tello puede ser controlado con un smartphone o un control remoto opcional, lo que lo hace fácil de usar sin necesidad de que la persona tenga experiencia con drones.
- Compatible con programación: El DJI Tello es compatible con la programación, lo que permite a los usuarios personalizar y automatizar misiones de vuelo [5].

1.3. Tipos de control de vuelo

El control de vuelo es el proceso mediante el cual un piloto o un sistema automático regula la dirección, velocidad y altura de un dron. Este proceso incluye la recepción y procesamiento de señales desde los sensores del dron, así como la transmisión de señales hacia los motores y actuadores del dron.

El control de vuelo es un aspecto crítico en la operación de los drones, ya que asegura que el dron se mueva de manera segura y estable, realice misiones específicas y evite colisiones con objetos u obstáculos [6].

Hay varios tipos de control de vuelo, incluyendo control remoto tradicional, control por aplicación móvil, control por voz, control automático y control por sensores. Cada tipo ofrece diferentes niveles de conveniencia, automatización y precisión en la operación de los drones.

1.3.1. Control de vuelo mediante control remoto, o rutas de vuelo

El control de vuelo mediante control remoto es en el que el piloto utiliza un control remoto físico para controlar el dron. En este tipo de control, el control remoto envía señales a un sistema de recepción en el dron, que luego procesa estas señales y las utiliza para regular la dirección, velocidad y altura del dron.

El control remoto tradicional es comúnmente utilizado para drones de mayor tamaño y alcance, ya que ofrece un mayor grado de precisión y control manual en la operación del dron. Los controles remotos pueden incluir sticks de dirección, gatillos, botones y pantallas LCD que muestran información sobre el estado del dron [7].

Aunque el control de vuelo remoto es menos convencional que el control por aplicación móvil, aún es una opción popular para los pilotos de drones, debido a que tiene un mayor grado de flexibilidad en la operación del dron [6].

Las aeronaves no tripuladas tienen incorporado un sistema de control de vuelo que les realiza maniobras y desplazamientos en el aire. Estos movimientos son controlados desde tierra por una persona o el vuelo es realizado mediante un plan de vuelo determinado previamente al despegue; Aunque, con este tipo de vuelo los drones tienden a mantener estas rutas preestablecidas, aunque existan perturbaciones en el ambiente [7].

1.3.2. Control de vuelo autónomo o automático.

El control de vuelo automático es en el que el dron es capaz de volar y realizar misiones de forma autónoma. En este tipo de control, el dron utiliza un sistema de navegación integrado que incluye sensores, cámaras, GPS y otros equipos para planificar y ejecutar misiones de manera autónoma.

El control automático es común en drones con aplicaciones comerciales y militares, ya que permite a los drones realizar misiones repetitivas y peligrosas sin la necesidad de un piloto humano. Por ejemplo, los drones pueden ser utilizados para inspeccionar instalaciones industriales, realizar mapeos geográficos y monitorear áreas naturales [7].

Aunque el control de vuelo automático ofrece una mayor eficacia en el manejo de los drones, también presenta algunos desafíos. Por ejemplo, los drones pueden requerir una gran cantidad de información para realizar misiones autónomas, y pueden ser más propensos a errores y fallas en comparación con el control humano.

1.3.3. Control de vuelo por aplicación móvil.

El control de vuelo por aplicación móvil es un tipo de control de vuelo en el que el piloto utiliza una aplicación en un dispositivo móvil para controlar el dron. En este tipo de control, la aplicación envía señales a un sistema de recepción en el dron, que luego procesa estas señales y las utiliza para regular la dirección, velocidad y altura del dron.

El control de vuelo por aplicación móvil es una opción popular para los drones de tamaño y alcance limitados, ya que ofrece un grado moderado de precisión y control manual en la operación del dron. Las aplicaciones móviles también pueden incluir características adicionales, como vistas en primera persona (FPV), planificación de rutas y grabación de vídeo. Aunque el control de vuelo por aplicación móvil es más conveniente que el control remoto tradicional, puede ser menos preciso debido a la menor cantidad de controles físicos disponibles. Además, la conexión inalámbrica entre el dispositivo móvil y el dron también puede ser intermitente, lo que puede afectar la estabilidad y la eficiencia en la operación del dron [8].



Figura 2. Aplicación móvil para control de dron DJI Tello.

1.4. Hélice del dron

Las hélices son un componente esencial en la mayoría de los drones. Son las partes móviles que generan el movimiento y el empuje del dron, lo que permite que el dron se eleve y se mueva en el aire.

Las hélices están compuestas por un eje central y unas aspas o palas. En la mayoría de los drones, hay al menos dos pares de hélices, uno en la parte delantera y otro en la parte trasera. Estas hélices pueden ser controladas de forma individual o en grupo para regular la velocidad y la dirección del dron.

Las hélices también pueden ser de diferentes tamaños y formas, y se pueden elegir en función del tipo de dron y la aplicación prevista. Por ejemplo, las hélices más grandes proporcionan más empuje, lo que es ideal para los drones más pesados, mientras que las hélices más pequeñas permiten una mayor maniobrabilidad y velocidad, lo que es adecuado para los drones más ligeros [9].



Figura 3. Hélice del dron DJI Tello.

1.4.1. Ángulo de ataque de la hélice

El ángulo de ataque de una hélice en un dron es el ángulo formado entre la dirección del flujo de aire y la dirección perpendicular al plano de las palas de la hélice.

El ángulo de ataque es un factor crítico en el desempeño de la hélice de un dron. Un ángulo de ataque demasiado grande puede causar pérdida de eficiencia y un aumento en la resistencia al avance, mientras que un ángulo de ataque demasiado pequeño puede resultar en una falta de empuje suficiente para mantener el vuelo estable. Es importante ajustar el ángulo de ataque de la hélice para obtener un equilibrio óptimo entre eficiencia y empuje [8].

1.4.2. Perfil aerodinámico

El perfil aerodinámico es la forma de la superficie de un objeto que se mueve a través del aire, y es un aspecto importante en la construcción y el diseño de los drones.

Un buen perfil aerodinámico es crucial para garantizar un buen rendimiento y estabilidad del dron. Por ejemplo, un perfil aerodinámico bien diseñado permite que el dron mantenga una estabilidad adecuada en el aire y evite vibraciones y otros problemas de estabilidad.

El perfil aerodinámico también afecta a la velocidad y la eficiencia del dron. Por ejemplo, un perfil aerodinámico más estrecho y en forma de flecha ayuda a reducir la resistencia y mejorar

la velocidad, mientras que un perfil aerodinámico más ancho ayuda a proporcionar una mayor estabilidad y control en el aire [9].

una placa que se encuentra inclinada hacia arriba que se encuentra en una corriente de aire es capaz de crear cierta fuerza para su sustentación. Un perfil aerodinámico no solo puede producir su propia sustentación, ya que está diseñado de una manera en la que, cuando es puesto en una corriente de aire, sea apto para aprovechar de una buena manera todas las fuerzas que se originan debido a los cambios de velocidad y presión [9].

1.4.3. Longitud y paso de la hélice

La longitud y el paso son dos características importantes de las hélices de los drones.

La longitud de la hélice se refiere a la medida desde un extremo de la pala hasta el otro, y puede variar dependiendo del tamaño y el modelo del dron. Las hélices más largas suelen proporcionar más empuje, pero también requieren más energía y pueden ser menos eficientes.

El paso se refiere a la distancia que la hélice se mueve a través del aire en una revolución completa. Un paso más grande puede proporcionar más empuje, pero también puede requerir más energía y ser menos eficiente. Por otro lado, un paso más pequeño puede mejorar la eficiencia, pero puede no proporcionar suficiente empuje para el dron.

La selección adecuada de la longitud y el paso de las hélices es importante para lograr un buen rendimiento y una buena estabilidad del dron. Estos valores pueden variar según el tamaño, el peso y el uso previsto del dron, y se deben elegir cuidadosamente en función de los requisitos específicos del dron [10].

1.5. Transmisión de videos sobre redes de datos

La transmisión de video sobre redes de datos se refiere a la transferencia de contenido de video a través de una red de computadoras. Esto puede incluir la distribución de videos en tiempo

real (como en el caso de una videoconferencia en línea) o la transmisión de video pregrabados (como en el caso de un servicio de transmisión de video en línea). A menudo se realiza utilizando protocolos específicos y formatos de codificación de video para garantizar una calidad óptima de transmisión y visualización.

La transmisión de video a través del transporte de señales por medio de redes de datos se ha convertido en una solución estándar tanto para estudiantes, trabajadores y el público en general, ya que el uso que se da solamente ha venido aumentando a lo largo de los años. Existen tres tipos de transmisión de vídeo:

- La descarga tradicional, donde se descarga el archivo del vídeo a reproducir.
- La descarga progresiva o pseudo transmisión donde el reproductor va mostrando el contenido a medida que este recibe los fragmentos de vídeo.
- El streaming es una transmisión de datos en tiempo real, normalmente se transmite audio o video o los dos al mismo tiempo [11].

1.6. Protocolos de comunicación mediante redes inalámbricas

Para que los diferentes dispositivos electrónicos puedan comunicarse entre ellos, es necesario la implementación de un mismo protocolo que, sin importar la marca, sistema operativo, arquitectura de fabricación, etc., puedan comunicarse entre sí fácilmente; pero esto no es tan sencillo ya que por la competencia que hay ciertos equipos tienen dificultados al comunicarse con otros que no sean específicamente de la misma marca que ofrece el fabricante.

Lo que establece las reglas para la transmisión de datos entre diferentes equipos electrónicos es lo que se conoce como protocolo, estos son asignados mediante capas, haciendo que cada uno de estos cumpla una función específica [12].

1.7. Tele operación

La tele operación se define como a la capacidad de manejar un sistema distancia a través de una red de comunicaciones. En general, la tele operación se utiliza para controlar robots, vehículos no tripulados o otros sistemas que se encuentran en una ubicación remota o peligrosa. La tele operación se logra mediante la transmisión de señales de control desde un controlador remoto hasta el sistema controlado, lo que permite a un operador realizar tareas a distancia. Estas señales de control pueden ser eléctricas o electromagnéticas, como una conexión a Internet o un enlace inalámbrico.

La tele operación se utiliza en una amplia variedad de aplicaciones, incluyendo la exploración remota de planetas, la manipulación de objetos en entornos peligrosos y la operación de vehículos submarinos. Además, la tele operación también se utiliza en la industria para realizar tareas que requieren habilidades especializadas [13].

1.8. Redes neuronales.

Las redes neuronales son un de modelo computacional que se inspira en la estructura y el comportamiento de las redes de neuronas en el cerebro humano. Las redes neuronales consisten en una serie de nodos conectados que se utilizan para procesar y transmitir información.

Dentro de la inteligencia artificial se usa una red neuronal, que sirve para realizar tareas como la clasificación, la regresión, la detección de patrones y la traducción automática. Cada nodo en una red neuronal representa una unidad de procesamiento que aplica una función matemática a la información de entrada, y los nodos están conectados entre sí a través de enlaces con pesos asignados. Los datos son enviados a través de la red a medida que los nodos procesan y transmiten la información a los nodos siguientes.

Las redes neuronales se entrenan utilizando grandes conjuntos de datos y técnicas de optimización como el descenso del gradiente para ajustar los pesos de los enlaces y mejorar la

precisión de la red. Una vez entrenadas, las redes neuronales pueden ser utilizadas para realizar predicciones y tomar decisiones automáticas sobre nuevos datos [31].

1.9. Realidad aumentada

La realidad aumentada (AR) es una tecnología que combina información virtual con el mundo real en tiempo real. La información virtual se superpone al mundo real mediante un dispositivo como un smartphone o una tableta, creando una experiencia de visualización combinada.

La realidad aumentada permite a los usuarios ver información adicional y gráficos en su entorno, como etiquetas virtuales en objetos reales o animaciones en el mundo real. Esta tecnología se utiliza en una amplia variedad de aplicaciones, incluyendo juegos, educación, marketing y asistencia técnica.

La realidad aumentada también puede ser utilizada en combinación con otras tecnologías, como la realidad virtual, para crear experiencias de realidad mixta que combinan elementos de la realidad aumentada y la realidad virtual [14].

La realidad aumentada no se ve limitada simplemente al uso del sentido de la vista ya que esta puede potenciar los diferentes sentidos como son: el olfato, tacto y audición; esta se puede implementar por reemplazo sensorial para expandir o intercambiar los sentidos de los usuarios que la utilicen, como, por ejemplo: aumentar la vista de los usuarios invidentes o de los usuarios con visión disminuida por medio del uso de señales auditivas, o aumentar el oído a usuarios sordos por medio del uso de señales visuales [15].

1.10. Software implementado

A continuación, se dará un breve resumen sobre cómo funciona cada uno de los sistemas que se implementaron para poder realizar este proyecto.

1.10.1. Python

Python es un lenguaje de programación de alto nivel interpretado, es decir, que no se necesita de un compilador para poder correr las aplicaciones programadas en el mismo, sino que se ejecuten de manera directa por medio del computador, utilizando el programa interpretador, por lo que no es necesario llevarlo a un lenguaje de máquina. Este lenguaje de programación se utiliza para desarrollar varios tipos de aplicaciones, ya que es sumamente sencillo de leer y escribir debido a su alta similitud con el lenguaje humano [17].

1.10.2. Google Colab

Google Colab fue diseñado para permitir a los usuarios crear y ejecutar códigos en línea sin necesidad de instalar ningún software en sus computadoras. Esto significa que los usuarios pueden usar la herramienta desde cualquier lugar con una conexión a internet. Esta herramienta es gratuita y fácil de usar.

Google Colab también permite a los usuarios compartir sus códigos con otros usuarios. Esto significa que los usuarios pueden trabajar en proyectos de código juntos sin tener que estar en la misma ubicación [20].

1.10.3. YoloV5

YoloV5 es una herramienta de detección de objetos de aprendizaje profundo que ofrece una mejor precisión y velocidad de detección de objetos que sus predecesores. Fue creado por el equipo de investigación de Facebook AI, y ha sido desarrollado para ser una herramienta versátil para detectar objetos en imágenes y videos [18].

YoloV5 es una herramienta basada en la arquitectura de red neuronal Convolutiva (CNN) y está diseñada para realizar detección de objetos de forma rápida y precisa. Utiliza una arquitectura de red neuronal llamada EfficientDet, que es una mezcla de varias técnicas de aprendizaje profundo, como el aprendizaje por refuerzo, el aumento de datos y la

regularización. Estas técnicas permiten que la red neuronal sea más precisa al detectar objetos, y también mejoran su rendimiento en términos de velocidad de procesamiento [19].

Además, YoloV5 es capaz de detectar una variedad de objetos, desde objetos pequeños como una manzana hasta objetos más grandes como un coche. Esto se logra mediante el uso de una red neuronal más profunda, lo que permite que la red sea más precisa al detectar objetos., lo que significa que la herramienta es útil para tareas como la clasificación de imágenes [18].

CAPÍTULO 2

2. ANÁLISIS DE ALTERNATIVAS

En este capítulo se dará una breve descripción de diferentes tipos de planificadores de ruta de vuelo en drones.

2.1. Diferentes tipos de planificadores de ruta utilizando drones.

Actualmente existen varios drones que son capaces de realizar la ruta de vuelo automáticamente, pero generalmente los costos son muy elevados, por ende, no son accesibles para todas las personas que los necesiten.

2.1.1. Alternativa 1: Planificación automática de rutas para un dron no tripulado.

En esta investigación se presenta un resumen general sobre lo que son los drones, la tecnología que se utiliza en ellos, y en que propósitos pueden servir actualmente, se destaca también la importancia del video en tiempo real, la transmisión de imágenes, y finalmente se valida más de 40 usos en los que el dron puede ayudar a la sociedad tanto en exploración, ingeniería, proyectos, planificación, logística, etc. [1].

2.1.2. Alternativa 2: Planificación automática de rutas para un dron no tripulado con dinámica de vuelo restringida.

En este artículo se resuelve de una manera breve como se puede realizar la tarea de planificación de ruta para un vehículo aéreo no tripulado multirotor (multicóptero), se lo hace mediante restricciones de la geometría de la ruta basado en el modelo de la dinámica de vuelo, todo esto se hace tanto teórica como empíricamente [2].

2.1.3. Alternativa 3: Drones como servicio: un análisis basado en simulación para la toma de decisiones en drones.

En este artículo, se estudia las simulaciones en drones y como las diferentes estrategias ayudan a la toma de decisiones para que los clientes y proveedores cuenten con este servicio. Ya que ayudan a maximizar los ingresos del proveedor y la satisfacción de los clientes.

Las diferentes combinaciones de factores que influyen en el comportamiento de los drones (por ejemplo, velocidad, distribución de clientes, criterios para juzgar las solicitudes de los clientes, duración del servicio, efecto de la batería y si los drones pueden cambiar su cliente objetivo sobre la marcha) [3].

2.1.4. Alternativa 4: Planificación de vuelo de drones para operaciones urbanas seguras mediante control remoto.

En este documento se describe los requisitos de una herramienta de planificación de vuelo para operaciones urbanas seguras, que se puede utilizar para diseñar operaciones teniendo en cuenta las restricciones y limitaciones de vuelo. Este sistema está diseñado para trabajar en coordinación con un sistema de gestión de tráfico no tripulado encargado de distribuir los recursos de espacio aéreo de muy bajo nivel disponibles entre diferentes operaciones, autorizarlas, y de monitorear el cumplimiento de los vuelos reales con las autorizaciones de vuelo [4].

2.2. Identificación de necesidades.

En este apartado se determinan las necesidades básicas que busca un consumidor al momento de escoger un dron para manejarlo. En la tabla a continuación se podrá observar cada una de las necesidades junto con su interpretación.

Tabla 1. Identificación de necesidades, Autores.

| Necesidades | Interpretación |
|--------------------------------------|---|
| Que sea económico y funcional | El dispositivo debe tener un precio bajo en comparación con otros productos similares en el mercado y ofrece una relación precio-calidad aceptable. |
| Fácil de transportar | Debe ser un dispositivo ligero, compacto y fácil de manejar, lo que lo hace conveniente para llevarlo de un lugar a otro. |
| De fácil manejo | Para manejar el dron debe tener funciones accesibles y fáciles de usar, lo que lo hace apto para una amplia gama de usuarios, incluyendo personas sin experiencia previa en tecnología. |
| Fácil comprensión de la programación | Se debe asegurar que el dispositivo contenga una interfaz de programación visual que permita a los usuarios crear y modificar el código sin escribirlo directamente. |
| Equipo con bajo consumo de energía | El dispositivo debe funcionar durante un período prolongado de tiempo sin necesidad de ser recargado o conectado a una fuente de alimentación eléctrica. |
| Cámara con buena definición | El dispositivo debe contar con una cámara la cual permita ver al usuario con claridad lo que el dron está visualizando por medio de la misma |

2.2.1. Jerarquización de las necesidades.

A continuación, se muestra la jerarquización de las necesidades del dron, las cuales se ponderan como 6 la necesidad menos importante y 1 la necesidad más importante.

Tabla 2. Grados de importancia respecto a las necesidades, Autores.

| Grado de Importancia | Necesidades |
|-----------------------------|--------------------------------------|
| 1 | Que sea económico y funcional |
| 2 | Fácil comprensión de la programación |
| 3 | Cámara con buena definición |
| 4 | De fácil manejo |
| 5 | Fácil de transportar |
| 6 | Equipo con bajo consumo de energía |

2.2.2. Encuestas para establecer la importancia relativa de las necesidades.

Se realizó unas preguntas entre personas cercanas a nosotros que tenían claro el objetivo y funcionamiento del proyecto para así, poder dar un valor relativo a las necesidades indicadas anteriormente, ponderadas como 1 como calificación baja y 5 como calificación alta.

Tabla 3. Necesidades y su importancia relativa.

| Necesidades | Encuestados | | |
|--------------------------------------|--------------------|---|---|
| Que sea económico y funcional | 5 | 4 | 5 |
| Fácil comprensión de la programación | 3 | 5 | 4 |
| Cámara con buena definición | 4 | 4 | 4 |
| De fácil manejo | 3 | 4 | 3 |
| Fácil de transportar | 3 | 4 | 2 |
| Equipo con bajo consumo de energía | 3 | 4 | 4 |

Con los resultados obtenidos en se concluye que lo más necesario es que sea económico y funcional, en la que todos estamos de acuerdo, ya que el equipo debe funcionar totalmente, logrando realizar su planificación de vuelo.

2.2.3. Comparación de las alternativas según las necesidades.

En base a los resultados de la encuesta de necesidades, se realizó una tabla comparativa, para diferenciar cuál de ellas es la que cumple con los puntos más importantes, y así, poder escoger la mejor.

En la tabla 4 se muestra la calificación de las alternativas de la planificación de vuelo del dron, ponderados con parámetros del 1 al 5, donde 1 es el valor más bajo de la calificación y 5 es el valor más alto.

Tabla 4. Análisis y calificación de las alternativas, Autores.

| ÍTEM | CRITERIO | Alternativa 1 | Alternativa 2 | Alternativa 3 | Alternativa 4 | REAL |
|---------------------------|--------------------------------------|---------------|---------------|---------------|---------------|------|
| | | Calificación | Calificación | Calificación | Calificación | |
| 1 | Que sea económico y funcional | 5 | 5 | 4 | 5 | 5 |
| 2 | Fácil comprensión de la programación | 5 | 5 | 3 | 5 | 5 |
| 3 | Cámara con buena definición | 4 | 4 | 5 | 3 | 5 |
| 4 | De fácil manejo | 5 | 5 | 3 | 3 | 5 |
| 5 | Fácil de transportar | 5 | 5 | 4 | 4 | 5 |
| 6 | Equipo con bajo consumo de energía | 4 | 5 | 3 | 4 | 5 |
| Sumatorio | | 28 | 29 | 22 | 24 | 30 |
| Índice porcentual | | 0,93 | 0,97 | 0,73 | 0,80 | 1 |
| Orden de selección | | 93% | 97% | 73% | 80% | 100% |

2.2.4 Resultados de las alternativas de trabajo.

En base a las necesidades anteriormente analizadas, se obtuvieron los siguientes resultados:

- La alternativa 1, cumple con la mayor puntuación en la mayoría de las necesidades, con excepción de ser un equipo con bajo consumo de energía, ya que al ser un dron que esta siempre en movimiento deberá mantenerse todo el tiempo cargándose.
- La alternativa 2, cumple con una buena puntuación en cada una de las necesidades, menos con la cámara de buena definición, pero no es problema para realizar el proyecto.

- La alternativa 3, tiene varias desventajas como que no es fácil de comprender la programación ni de manejar, por lo cual no es una buena alternativa para realizar el proyecto.
- Por último, la alternativa 4, no tiene buena puntuación en ninguna de las necesidades, por lo que se descarta.

La mejor opción para realizar este proyecto es la alternativa 2, ya que cumple con todas las necesidades antes mencionadas.

2.3 IDEs de programación

En este apartado se indicarán los IDE (Entorno Integrado de desarrollo) que son los programas de software que proporciona un entorno integrado y completo para desarrollar y probar código fuente. A continuación, se detallará brevemente el entorno utilizado (PyCharm), y otros dos con los que se podría realizar el mismo proyecto [17].

2.3.1. Encuestas para establecer la importancia relativa de las necesidades que buscan los usuarios en los IDEs de programación.

Se realizó una pequeña encuesta entre personas cercanas a nosotros, las cuales comprenden cómo funcionan los lenguajes de programación y que es lo que buscan ellos al momento de elegir uno, de esta manera se obtuvieron las características, y fueron calificadas como 1 la más baja y 5 la más alta.

Tabla 5. Características de los IDEs y su importancia relativa, Autores.

| Características | Encuestados | | |
|--|--------------------|---|---|
| Código avanzado con destacado sintáctico | 5 | 4 | 5 |
| Entorno integrado y completo | 4 | 5 | 4 |
| Análisis de datos | 4 | 4 | 4 |
| Facilidad de instalación de librerías | 5 | 5 | 4 |
| Manejo online en página web | 3 | 3 | 2 |
| Desarrollo de aplicaciones | 4 | 4 | 5 |

Con los resultados se determinó que lo más importante es que tenga código avanzado con destacado sintáctico y que tenga facilidad de instalación de librerías.

2.3.2. PyCharm

Sé utilizó esta IDE ya que incluye un editor de código avanzado con destacado sintáctico, un depurador, un sistema de control de versiones integrado, un gestor de proyectos, y una interfaz de programación visual, entre otras características. Además, también ofrece una amplia gama de plugins y extensiones disponibles para descargar e instalar para mejorar aún más sus funciones [18].

2.3.3. Jupyter Lab.

Jupyter Lab es un entorno de desarrollo web que se enfoca en el análisis de datos y la exploración de datos interactivos. Jupyter Lab permite a los usuarios crear y compartir documentos que contienen código, gráficos, visualizaciones y texto explicativo.

Se descartó esta IDE debido a que, aunque también es útil para desarrolladores en Python, es más adecuado para el análisis de datos y la exploración de datos interactivos [19].

2.3.4. Spyder

Spyder es un IDE más ligero y enfocado en la ciencia de datos y el análisis de datos. Ofrece un editor de código con destacado sintáctico, un depurador, un gestor de consolas, un gestor de variables y una interfaz de programación visual, entre otras características.

Se descartó esta opción debido a que, PyCharm es una herramienta más potente y completa para el desarrollo de aplicaciones de Python, mientras que Spyder es más adecuado para la ciencia de datos y el análisis de datos [20].

2.3.4. Comparación de los IDEs según las necesidades

En base a los resultados de la encuesta de necesidades en los IDEs de programación, se realizó una tabla comparativa, para diferenciar cuál de ellas es la que cumple con los puntos más importantes, y así, poder escoger la mejor.

En la tabla 6 se muestra la calificación de las alternativas de la planificación de vuelo del dron, ponderados con parámetros del 1 al 5, donde 1 es el valor más bajo de la calificación y 5 es el valor más alto.

Tabla 6. Análisis y calificación de los IDEs de programación, Autores.

| ÍTEM | CRITERIO | PyCharm | Jupyter Lab | Spyder | REAL |
|---------------------------|--|--------------|--------------|--------------|--------------|
| | | Calificación | Calificación | Calificación | Calificación |
| 1 | Código avanzado con destacado sintáctico | 5 | 4 | 4 | 5 |
| 2 | Entorno integrado y completo | 5 | 4 | 4 | 5 |
| 3 | Análisis de datos | 4 | 5 | 5 | 5 |
| 4 | Facilidad de instalación de librerías | 5 | 4 | 3 | 5 |
| 5 | Manejo online en página web | 4 | 5 | 3 | 5 |
| 6 | Desarrollo de aplicaciones | 4 | 4 | 4 | 5 |
| Sumatorio | | 27 | 26 | 23 | 30 |
| Índice porcentual | | 0,90 | 0,87 | 0,77 | 1 |
| Orden de selección | | 90% | 87% | 77% | 100% |

2.3.5. Resultados de las alternativas de IDEs.

En base a las características de los IDEs anteriormente analizadas, se obtuvieron los siguientes resultados:

- **Spyder:** Spyder es un IDE de código abierto para programación en Python que se enfoca en la ciencia de datos y el análisis numérico. Es un IDE intuitivo y fácil de usar que incluye características como un editor de código, un depurador, un explorador de variables y una consola interactiva. Spyder es una buena opción para científicos de datos y analistas que desean trabajar con Python.
- **PyCharm:** PyCharm es un IDE comercial desarrollado por JetBrains que se enfoca en el desarrollo de aplicaciones en Python. Es un IDE muy completo que incluye una amplia gama de características, como un editor de código avanzado, un depurador, herramientas de prueba y control de versiones. PyCharm es una buena opción para desarrolladores de aplicaciones que buscan un IDE completo y potente para su trabajo con Python.
- **Jupyter:** Jupyter es una plataforma web de código abierto que permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto explicativo. Jupyter es popular en la comunidad científica y de análisis de datos debido a su facilidad de uso y capacidad para combinar código, texto y visualizaciones en un solo documento. Jupyter es una buena opción para científicos de datos y analistas que desean explorar y compartir sus análisis de datos de manera efectiva.

2.4 Sistema de implementación.

En este apartado se indicarán los sistemas que se pueden utilizar para implementar una solución específica al problema del proyecto. El sistema de implementación se utiliza para llevar a cabo

tareas específicas, como la integración de diferentes sistemas, la automatización de procesos empresariales o la creación de aplicaciones personalizadas.

2.4.1. Encuestas para establecer la importancia relativa de las necesidades que buscan los usuarios en los Software.

Se realizó una pequeña encuesta entre personas cercanas a nosotros, las cuales comprenden cómo funcionan los softwares o sistemas que se piensan utilizar y que es lo que buscan ellos al momento de entrenar la inteligencia artificial, de esta manera se obtuvieron las características, y fueron calificadas como 1 la más baja y 5 la más alta.

Tabla 7. Características de los softwares y su importancia relativa, Autores.

| Características | Encuestados | | |
|---|-------------------------------------|---|---|
| | Detección de objetos en tiempo real | 5 | 5 |
| Aprendizaje automático | 5 | 5 | 4 |
| Técnicas de detección de objetos | 4 | 5 | 4 |
| Modelos de aprendizaje automáticos | 4 | 4 | 4 |
| Amplia variedad de bibliotecas para el desarrollo | 3 | 3 | 2 |
| Detección de objetos lenta y precisa | 4 | 3 | 5 |

Con los resultados obtenidos en la encuesta se pudo observar que la necesidad más importante es que tenga detección de objetos en tiempo real, lo que es muy útil para el desarrollo de nuestro proyecto.

2.4.2. YoloV5.

YOLOv5 es un sistema de detección de objetos en imágenes y videos desarrollado por el equipo de Joseph Redmon, autor del sistema YOLO original. La versión 5 es la más reciente y ha sido optimizada para ofrecer mejores resultados de detección de objetos en comparación con versiones anteriores.

YOLOv5 utiliza una red neuronal para realizar la detección de objetos en tiempo real. La entrada de la red es una imagen o un frame de video y la salida son las coordenadas de las cajas delimitadoras y las clases correspondientes de los objetos detectados en la imagen o video.

Además de ser eficiente en términos de tiempo de procesamiento, YOLOv5 también ofrece una buena precisión en la detección de objetos. Esto lo hace ideal para una variedad de aplicaciones, como la seguridad, la robótica y la inteligencia artificial [28].

2.4.3. TensorFlow.

TensorFlow es una plataforma de código abierto para el desarrollo de modelos de aprendizaje automático y deep learning. Fue desarrollado por Google y ha sido adoptado ampliamente en la industria y la academia.

TensorFlow permite a los desarrolladores y investigadores crear y entrenar modelos de aprendizaje automático, así como también implementar y desplegar modelos entrenados en una amplia variedad de plataformas, incluyendo dispositivos móviles, servidores y la nube.

Una de las principales características de TensorFlow es su capacidad para ejecutar cálculos en un grafo computacional, lo que permite una alta eficiencia en términos de rendimiento y escalabilidad. También proporciona una amplia documentación y herramientas de desarrollo para facilitar el desarrollo de modelos de aprendizaje automático y deep learning [29].

2.4.4. Haar Cascade.

Haar Cascade es un algoritmo de detección de objetos desarrollado por Viola y Jones en 2001. Es uno de los primeros algoritmos de detección de objetos eficientes en términos de tiempo de procesamiento y es ampliamente utilizado en el campo de la visión por computadora.

Haar Cascade funciona analizando las características de una imagen y comparándolas con una serie de patrones previamente entrenados, conocidos como cascadas de Haar. Estas cascadas

se basan en la detección de diferencias de intensidad en la imagen y pueden ser utilizadas para detectar objetos específicos, como rostros humanos, ojos, etc.

El algoritmo Haar Cascade es muy eficiente en términos de tiempo de procesamiento y es ampliamente utilizado en aplicaciones que requieren una detección rápida de objetos en tiempo real, como la seguridad, la robótica y la inteligencia artificial. Sin embargo, su precisión puede ser limitada en comparación con otros algoritmos más recientes, como YOLO o R-CNN [30].

2.4.4. Comparación de los Sistemas de implementación según las necesidades

En base a los resultados de la encuesta de necesidades en los sistemas o softwares de implementación, se realizó una tabla comparativa, para diferenciar cuál de ellas es la que cumple con los puntos más importantes, y así, poder escoger la mejor.

En la tabla 8 se muestra la calificación de las alternativas de la planificación de vuelo del dron, ponderados con parámetros del 1 al 5, donde 1 es el valor más bajo de la calificación y 5 es el valor más alto.

Tabla 8. Análisis y calificación de los sistemas de implementación, Autores.

| ÍTEM | CRITERIO | YoloV5 | Tensor Flow | Haar Cascade | REAL |
|---------------------------|---|--------------|--------------|--------------|--------------|
| | | Calificación | Calificación | Calificación | Calificación |
| 1 | Detección de objetos en tiempo real | 5 | 4 | 4 | 5 |
| 2 | Aprendizaje automático | 5 | 3 | 3 | 5 |
| 3 | Técnicas de detección de objetos | 5 | 5 | 5 | 5 |
| 4 | Modelos de aprendizaje automáticos | 5 | 4 | 3 | 5 |
| 5 | Amplia variedad de bibliotecas para el desarrollo | 5 | 3 | 3 | 5 |
| 6 | Detección de objetos lenta y precisa | 5 | 3 | 4 | 5 |
| Sumatorio | | 30 | 22 | 22 | 30 |
| Índice porcentual | | 1 | 0,73 | 0,73 | 1 |
| Orden de selección | | 100% | 73% | 73% | 100% |

2.4.5. Resultados de las alternativas de softwares de implementación.

En base anteriormente a las características de los sistemas analizados anteriormente, se obtuvieron los siguientes resultados:

- **YOLOv5:** YOLOv5 es una técnica de detección de objetos en tiempo real que utiliza una red neuronal profunda para clasificar y localizar objetos en imágenes y videos. Es una de las técnicas de detección de objetos más rápidas y precisas disponibles y se utiliza ampliamente en aplicaciones de inteligencia artificial y visión por computadora.

- Haar Cascade: Haar Cascade es una técnica de detección de objetos basada en características que utiliza una combinación de patrones simples llamados "cascadas" para detectar objetos en imágenes. Es una técnica de detección de objetos efectiva, pero menos precisa y rápida que YOLOv5.
- TensorFlow: TensorFlow es una plataforma de aprendizaje automático y visión por computadora desarrollada por Google. Es una de las plataformas más populares para el desarrollo de aplicaciones de inteligencia artificial y se utiliza ampliamente en investigación y producción. TensorFlow proporciona una amplia gama de herramientas y bibliotecas para el desarrollo de modelos de aprendizaje automático, incluyendo técnicas de detección de objetos como YOLOv5.

CAPÍTULO 3

3. DISEÑO E IMPLEMENTACIÓN

Este capítulo abarca la simulación, en la que se ve como cambiaría el vuelo del dron mediante los distintos ángulos de incidencia de la hélice, asimismo se evidencia estos resultados mediante la impresión de las hélices y probando cada una de ellas en el dron.

3.1. *Diseño de la hélice en Solid Works.*

Se detallan todos los pasos para obtener un diseño virtual de la hélice del dron, y con la información obtenida del análisis del ángulo original de la hélice, se puede realizar las pruebas en el cambio de este, mediante el análisis de la herramienta “Simulación de Fluidos” que tenemos en el software de diseño Solid Works, y a su vez mediante la impresión en 3D de está para realizar pruebas físicas en el dron.

3.1.1. *Aproximación fotográfica.*

De acuerdo con la inspección realizada a una imagen de forma lateral tomada de la hélice, la cual se muestra en la figura 4, el ángulo de la hélice forma una inclinación de 108.86° , que restándole los 90° de la línea paralela, obtenemos nuestro ángulo de referencia de 18.86° .

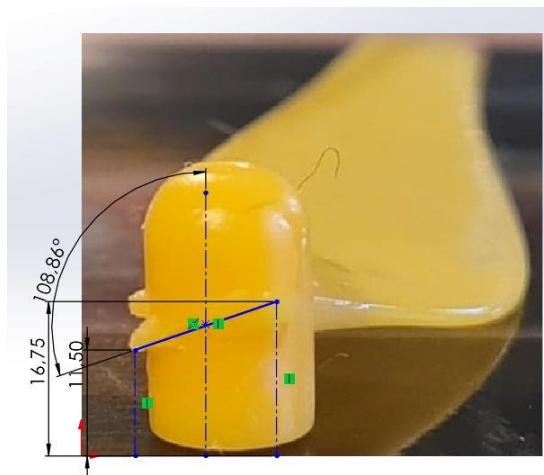


Figura 4. Determinación del ángulo de incidencia.

3.1.2. Aproximación por medición.

De acuerdo con la inspección realizada de la hélice mediante mediciones con el calibrador digital, se pudo generar un diseño con las medidas obtenidas, la cual se muestra en la figura 5, el ángulo de la hélice forma una inclinación de 108.65° , que restándole los 90° de la línea paralela, obtenemos nuestro ángulo de referencia de 18.65° .

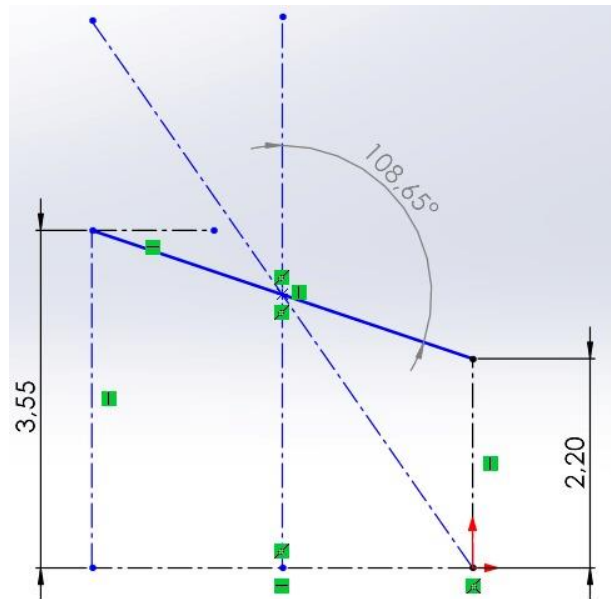


Figura 5. Determinación del ángulo de incidencia.

3.1.3. Diseño de la hélice.

En el programa de diseño se implementó el método de la ingeniería inversa, en este caso tomamos las mediciones de la hélice que viene en el dron y se realizaron las respectivas líneas de trazos para poder tener un diseño muy acercado al original para nuestras posteriores pruebas, como se lo ve en la figura 6.

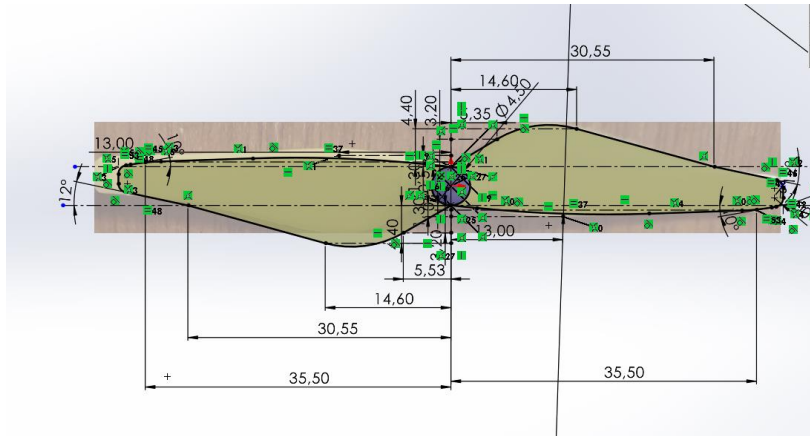


Figura 6. Diseño de la hélice en SolidWorks

Una vez logrado el croquis inicial, se detallan todos los aspectos para lograr la semejanza con el diseño original, como se indica en la figura 7, teniendo en cuenta un pequeño cambio en la parte superior de la hélice para que este no genere inconvenientes al momento de llevarlo al software de impresión y por ende a la impresora 3D cuando se requiera tener en físico la hélice.

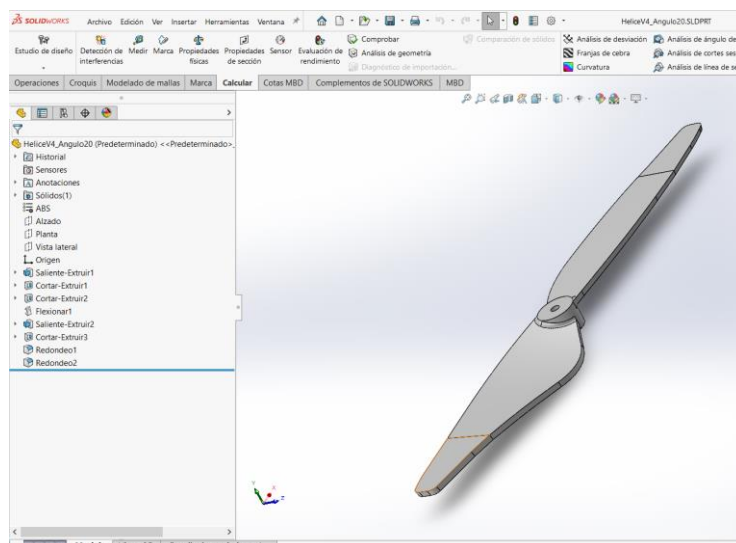


Figura 7. Diseño de la hélice en SolidWorks

3.1.4. Diseño del soporte de la hélice en SolidWorks.

Para poder ejecutar todos los estudios en la hélice se creó un prototipo semejante al motor del dron, esto con el fin de simular la colocación de la hélice sobre un eje rotativo, como se lo puede observar en la figura 8.

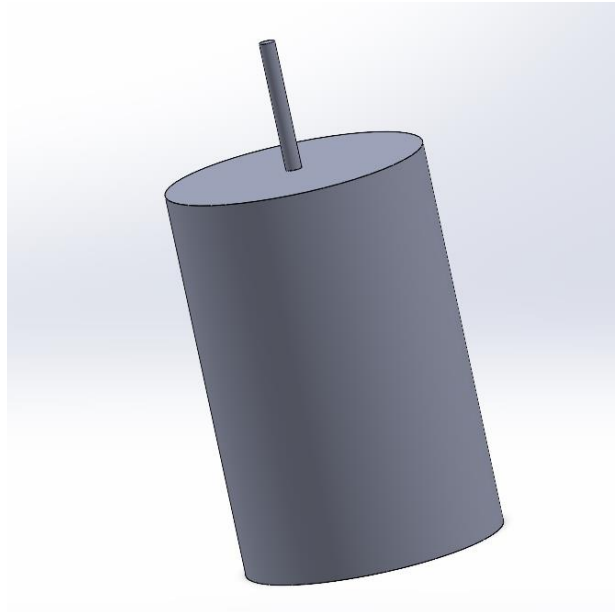


Figura 8. Diseño del soporte de la hélice en SolidWorks

3.1.5. Implementación del área de simulación en SolidWorks.

En el programa de simulación se requiere de un túnel de viento imaginario que se colocó alrededor de la hélice el cual nos ayudará para generar la trayectoria del paso del aire a través del mismo, esto con las variables a tomar en cuenta, que son determinadas automáticamente por el programa ya que este tiene implementado al aire como un fluido estándar dentro de sus características, como se indica en la figura 9.

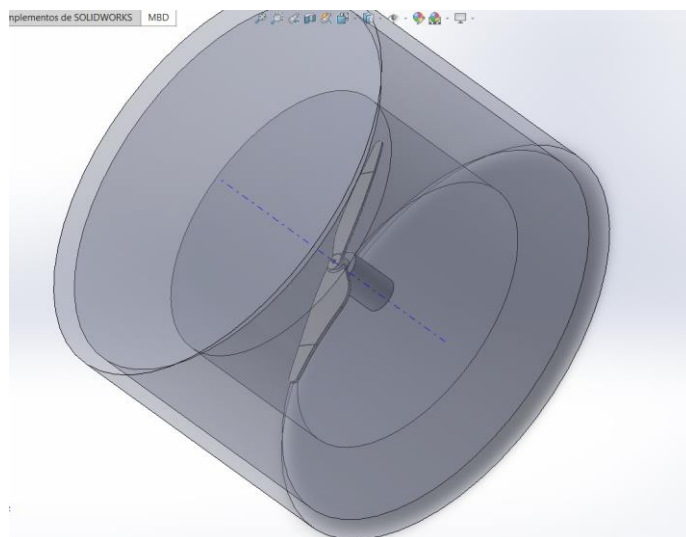


Figura 9. Implementación del área de simulación en SolidWorks

3.2. Control del dron mediante phyton

A continuación, se explicará cómo se logró el control del dron en el proyecto.

3.2.1. Control mediante el computador.

Se realizó el control a través de comandos de programación con Python, mediante los cuales se puede realizar que el dron sea capaz de tomar fotos, grabar vídeos y se pueda conducir manualmente mediante el teclado del computador.

3.2.1.1. Librería DJI Tello.

Por medio de la librería del dron DJI Tello nos ayudamos en varias cosas como el poder dar soporte al dron para grabar videos, también es utilizada para manejar enjambre de drones, entre otras cosas que son necesarias para poder manejar el dron desde la computadora.

Dentro del proyecto se utilizó la librería correspondiente del dron en Python para poder comunicarnos con este a través del ordenador, ejecutando instrucciones detalladas en el dron, así como la obtención de la batería, que despegue, que se mueva para arriba o para abajo e indicar las imágenes que el dron ha capturado.

3.3. Realidad aumentada.

La realidad aumentada se la realizó con la ayuda de la cámara del dron, esta capta un código QR, el cual contiene una imagen. El objetivo de esto es que, al momento de que una máquina arroje el error por medio del código, el dron se acerque a ella y nos pueda indicar si es que está en error, al contrario, mientras la máquina no muestre ningún error solo arroja el código de que todo está bien y el dron seguirá su camino en búsqueda de algún error para poder mandar al usuario.

3.3.1. Marcadores aruco.

El sistema Aruco, es un grupo de marcadores y algoritmos que permite su identificación y detección. Un marcador Aruco, es un cuadro sintético compuesto por una matriz binaria interna, la cual ayuda a identificar el objetivo. Dentro del proyecto se utilizó este marcador para que las maquinas puedan arrojar el error, y que este sea mandado al usuario para su pronta reparación.

3.3.1.1. Generador online de marcadores aruco.

El sistema de marcador se lo realizó mediante una página online, en donde se colocó las dimensiones del marcador que necesitábamos y lo generó automáticamente.

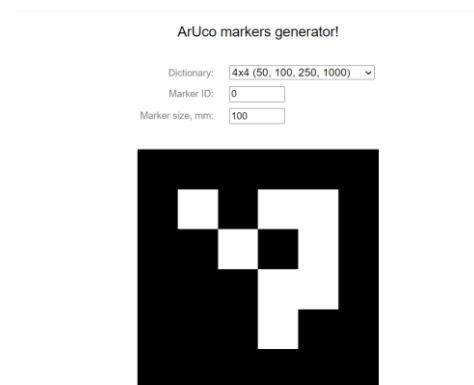


Figura 10. Generador de marcador aruco online.

En la figura 11 se detalla cómo se ve el marcador y que imagen se colocaría para que lo muestre en la cámara del dron mediante la generación de una realidad aumentada.



Figura 11. Ejemplo de marcador aruco

3.4. Detector de objetos.

La detección de objetos permite a los sistemas informáticos ver su entorno y detectar objetos o imágenes en fotogramas, estos forman parte de la estructura de los videos.

Se tomaron varias imágenes del dron desde distintos lugares dentro del espacio de trabajo, a partir de esto, la cámara será capaz de detectar en que posición se encuentra el dron. Para poder lograrlo se realizó distintas etapas de procesos que se detallaran a continuación:

3.4.1. Hardware y software utilizados para el entrenamiento de la red neuronal.

En este punto, el computador ya tiene la información suficiente para enseñarle lo que queremos que detecte, se eligió un marco de aprendizaje para facilitar las cosas, el marco de aprendizaje son estructuras preestablecidas que ayudan a ahorrar código que está escrito, de esta manera evitamos errores. Se realizó el entrenamiento por medio de Google Colab y mediante un computador que contaba con una buena tarjeta de video instalada, dentro de la página web, se escoge la opción de realizarlo por medio de YoloV5; se verifica que el acelerador de hardware este en GPU para su mayor rendimiento. En el computador se debe tener instalado los cuda Cores en caso de tener una tarjeta gráfica nvidia y a su vez que estos sean compatibles con la versión de Yolo que se trabaje.

3.4.2. YoloV5.

Se utilizó YoloV5, que es una familia de modelos y arquitecturas para la detección de objetos, su implementación en este caso es la de detectar al dron en un área de trabajo designada, esto se lo hará mediante una cámara colocada en la parte superior de dicha área de trabajo para que abarque toda la cobertura del vuelo del dron a través de la sala designada; se deben de cumplir los requerimientos que el programa necesita, como se lo indica en la figura 12 , los cuales se explicados a continuación.

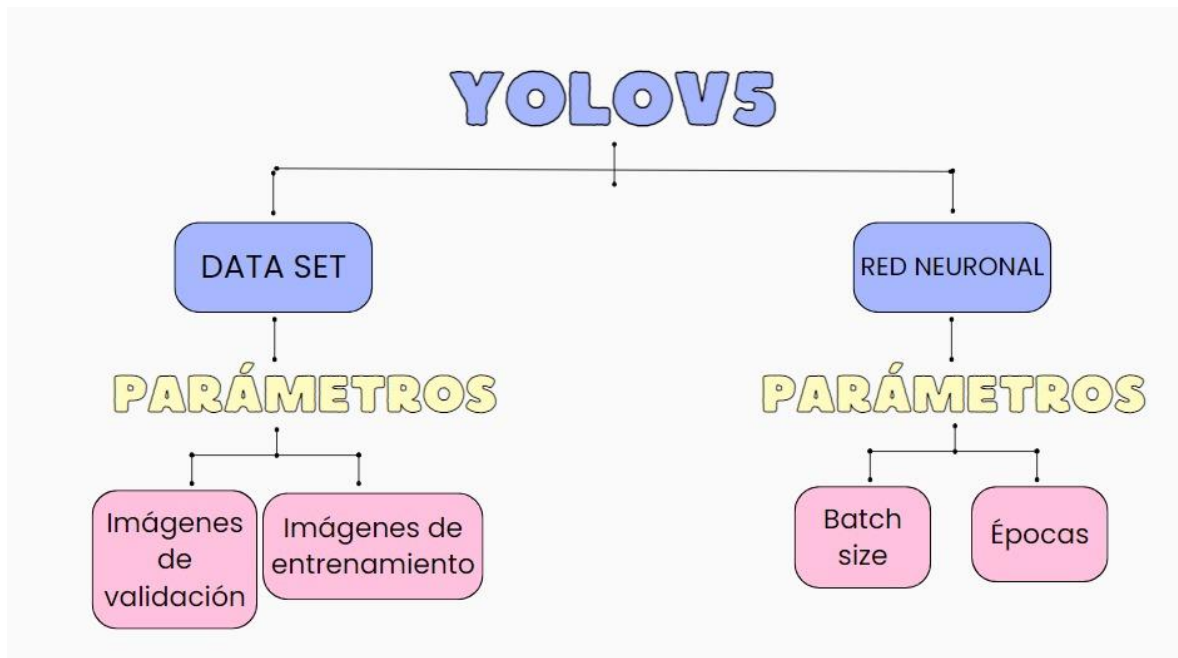


Figura 12. Parámetros importantes al entrenar con YoloV5.

3.4.2.1. Crear el Dataset.

En esta etapa, se reúne toda la información que la cámara debe detectar, es decir almacenar el conjunto de imágenes del objeto que vamos a detectar, en nuestro caso varias imágenes del dron. Las imágenes fueron tomadas dentro del área de trabajo del dron, también desde varios lugares de trabajo y a distintas distancias, como se indica en la figura 13. A demás se realizó un etiquetado de cada una de las fotografías, donde se señaló con un rectángulo la región en la que se encontraba el dron, de esta manera se obtuvo almacenada toda la información que requiere la cámara para poder diferenciar el dron.



Figura 13. Archivos necesarios para la creación del dataset.

3.4.2.1.1. Creación de imágenes.

Se realizó la toma de alrededor de 200 imágenes del dron más otras imágenes descargadas de internet, para mejorar la efectividad del entrenamiento de la inteligencia artificial ya que existirá una mayor retroalimentación en la base de datos a la que acceda, y de esta manera sea capaz de detectar al dron en toda el área del lugar de trabajo, como se ve en la figura 14.



Figura 14. Imágenes tomadas del dron

3.4.2.1.2. Creación de etiquetas.

Se utilizó el programa de etiquetado llamado Make Sense, por medio de este se etiquetó varias imágenes dentro de un rectángulo para definir el lugar en el que se encontraba el objeto y que el computador sea capaz de detectarlo con una mejor precisión, este procedimiento se lo realizó en dos partes una de entrenamiento y otra de validación con las imágenes tomadas previamente, como se lo indica en la figura 15.



Figura 15. Etiquetado del dron en la región en la que se encuentra.

3.4.2.2. Batch size.

Las imágenes generadas en la creación del dataset se toman por lotes para que se analicen de manera simultaneas varias a la vez, el programa nos permite realizar esta toma de lotes desde 4, 16, 20, 32, 40, 64, 80, 96 y 128.

3.4.2.3. Épocas.

Se realizó ciclos de entrenamiento al computador por medio de las imágenes, entre mayor es el número de imágenes mayor entrenamiento va a tener el computador, y así ayudar para que la información sea más precisa al momento de detectar el dron.

3.4.2.4. Inferencia.

Se puso en práctica lo que el programa aprendió a detectar como un dron luego del entrenamiento mediante YoloV5, en este punto el computador está listo para detectar el dron dentro del área de trabajo designada como se indica en la figura 16.

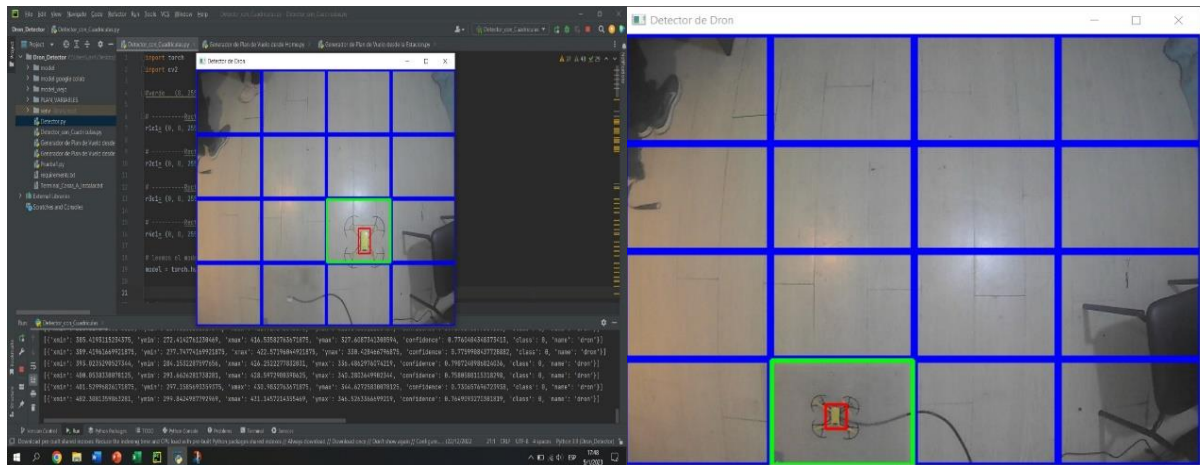


Figura 16. Cámara detectando el cuadrante donde se encuentra el dron.

3.4.3. Detección del dron mediante cuadrículas.

Se realizó el entrenamiento de la detección mediante cuadrículas, como se puede observar en la Figura 17 la cual es una referencia de la cuadrícula generada en Python en la que se detectará al dron para saber en qué posición se encuentra para la generación del plan de vuelo y como ayuda visual para el operador de que el dron se moviliza por el espacio adecuado para llegar a la planta que tenga algún error. Las columnas se dividen en 4; para la columna 1 va de 0 a 160 pixeles, para la columna 2 va de 163 a 320 pixeles, para la columna 3 va de 323 a 480 pixeles, y para la columna 4 va de 483 a 640 pixeles, estas tienen una separación de 3 pixeles por la anchura de la línea del rectángulo que se genera. En cambio, para las filas de los rectángulos se tiene: el rectángulo 1 va de 1 a 120 pixeles. El rectángulo 2 va de 123 a 240 pixeles. El rectángulo 3 va de 243 a 360 pixeles. El rectángulo 4 va de 363 a 479 pixeles.

| | | C1 | C2 | C3 | C4 | | | | |
|----|-----|----|-----|-----|-----|-----|-----|-----|-----|
| | | 0 | 160 | 163 | 320 | 323 | 480 | 483 | 640 |
| R1 | 1 | | | | | | | | |
| | 120 | | | | | | | | |
| R2 | 123 | | | | | | | | |
| | 240 | | | | | | | | |
| R3 | 243 | | | | | | | | |
| | 360 | | | | | | | | |
| R4 | 363 | | | | | | | | |
| | 479 | | | | | | | | |

Figura 17. Referencia de cuadrícula generada en Python.

3.4.3.1. Detección de dron mediante cuadrículas columna 1.

Se realizó el entrenamiento de la detección del dron mediante cuadrículas, como se puede observar en la Figura 18 la cual es una ayuda visual de cómo se programó la detección que hace el programa de inteligencia virtual y como este interactúa con las diferentes posiciones del dron dentro de las cuadrículas de la columna 1, el cual está dividido entre un rango de 0 a 150 pixeles; para el 1er rectángulo este va de 0 a 140 pixeles, indicado con color rojo. El segundo rectángulo va desde 80 a 260 pixeles, indicado con color morado. El tercer rectángulo va desde 200 a 360 pixeles, indicado con color amarillo. Por último, el rectángulo 4 va desde 320 a 480 pixeles, indicado con color verde.

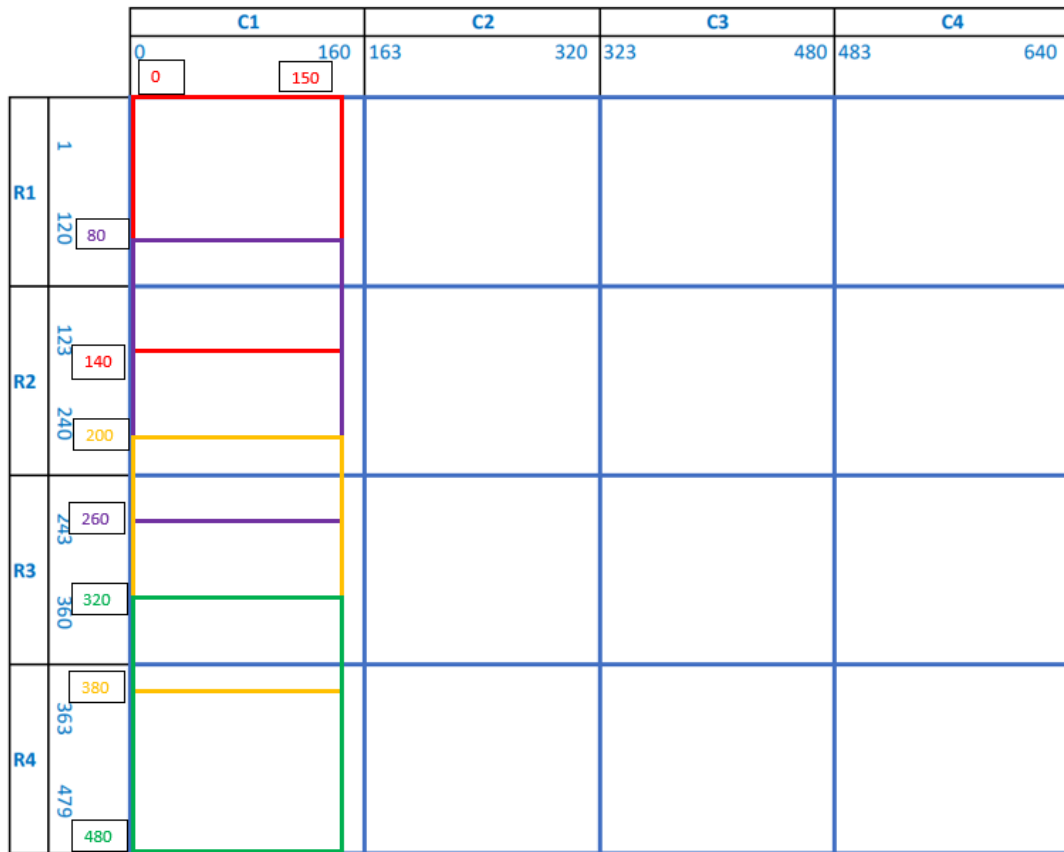


Figura 18. Detección del dron mediante cuadrículas.

3.4.3.2. Detección de dron mediante cuadrículas columna 2.

Se continuó con el entrenamiento de la detección del dron mediante cuadrículas, como se puede observar en la Figura 19 la cual es una ayuda visual de cómo se programó la detección que hace el programa de inteligencia virtual y como este interactúa con las diferentes posiciones del dron dentro de las cuadrículas de la columna 2, el cual está dividido entre un rango de 100 a 320 pixeles; para el 1er rectángulo este va de 0 a 140 pixeles, indicado con color verde. El segundo rectángulo va desde 80 a 260 pixeles, indicado con color amarillo. El tercer rectángulo va desde 200 a 360 pixeles, indicado con color morado. Por último, el rectángulo 4 va desde 320 a 480 pixeles, indicado con color rojo.

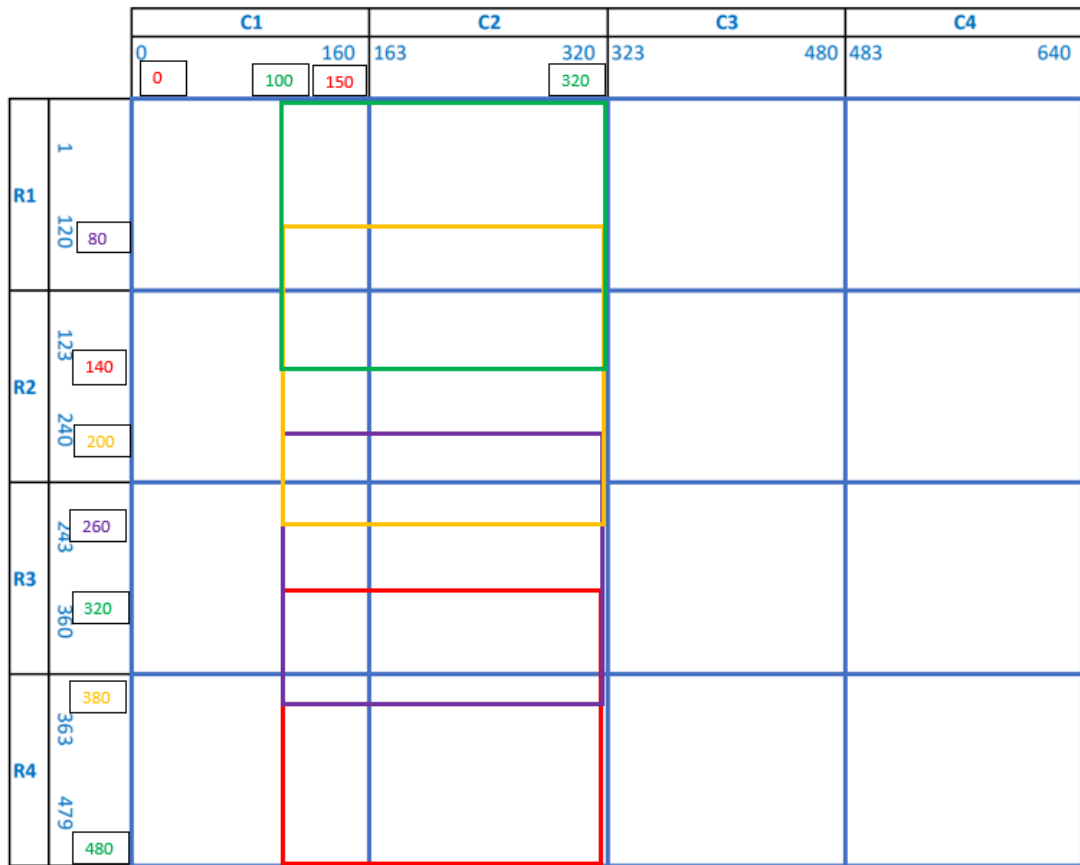


Figura 19. Detección del dron mediante cuadrículas

3.4.3.3. Detección de dron mediante cuadrículas columna 3.

En el entrenamiento de la detección del dron mediante cuadrículas, como se puede observar en la Figura 20 la cual es una ayuda visual de cómo se programó la detección que hace el programa de inteligencia virtual y como este interactúa con las diferentes posiciones del dron dentro de las cuadrículas de la columna 3, el cual está dividido entre un rango de 321 a 520 píxeles; para el 1er rectángulo este va de 0 a 140 píxeles, indicado con color verde. El segundo rectángulo va desde 80 a 260 píxeles, indicado con color amarillo. El tercer rectángulo va desde 200 a 360 píxeles, indicado con color morado. Por último, el rectángulo 4 va desde 320 a 480 píxeles, indicado con color rojo.

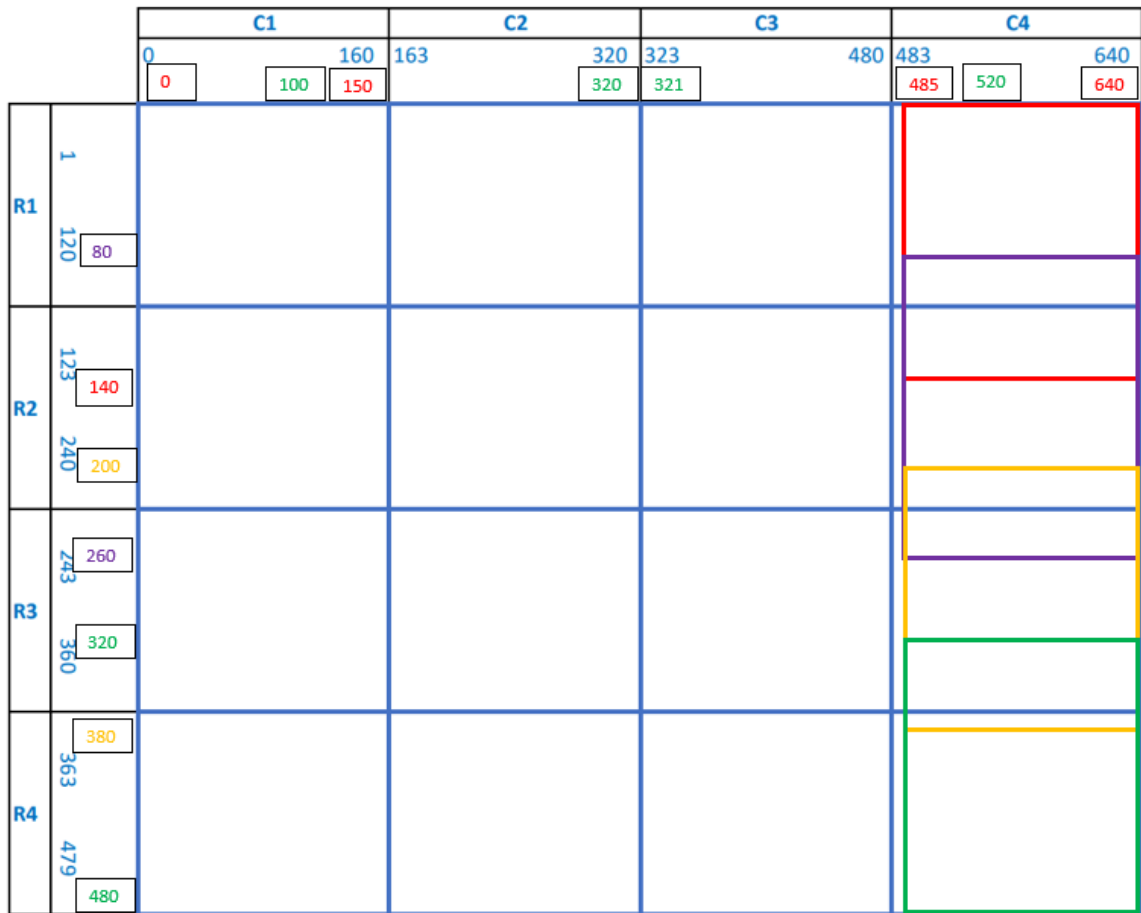


Figura 21. Detección del dron mediante cuadrículas

3.4.3.5. Vista general de la trayectoria del dron dentro de las cuadrículas.

Se realizó una vista de cómo interpreta la trayectoria del dron, como se puede observar en la figura 22, que, al momento de generar la detección del dron con inteligencia artificial, cambiando el color de las cuadrículas en las que está el dron de color azul a color verde si está en la trayectoria correcta o roja de estar fuera del rango de la misma. Como se indica en el anexo 1, el dron ya puede ser detectado por nuestro programa, conociendo la posición de este en las cuadrículas correspondientes. Una vez teniendo al dron en espacios específicos, se puede crear un plan de vuelo que vaya de acuerdo al vuelo que este esté haciendo, como se indica en el anexo 2 para el caso de la trayectoria desde home hacia la estación y para cuando este regrese a home se genera otro plan de vuelo especificado en el anexo 3.

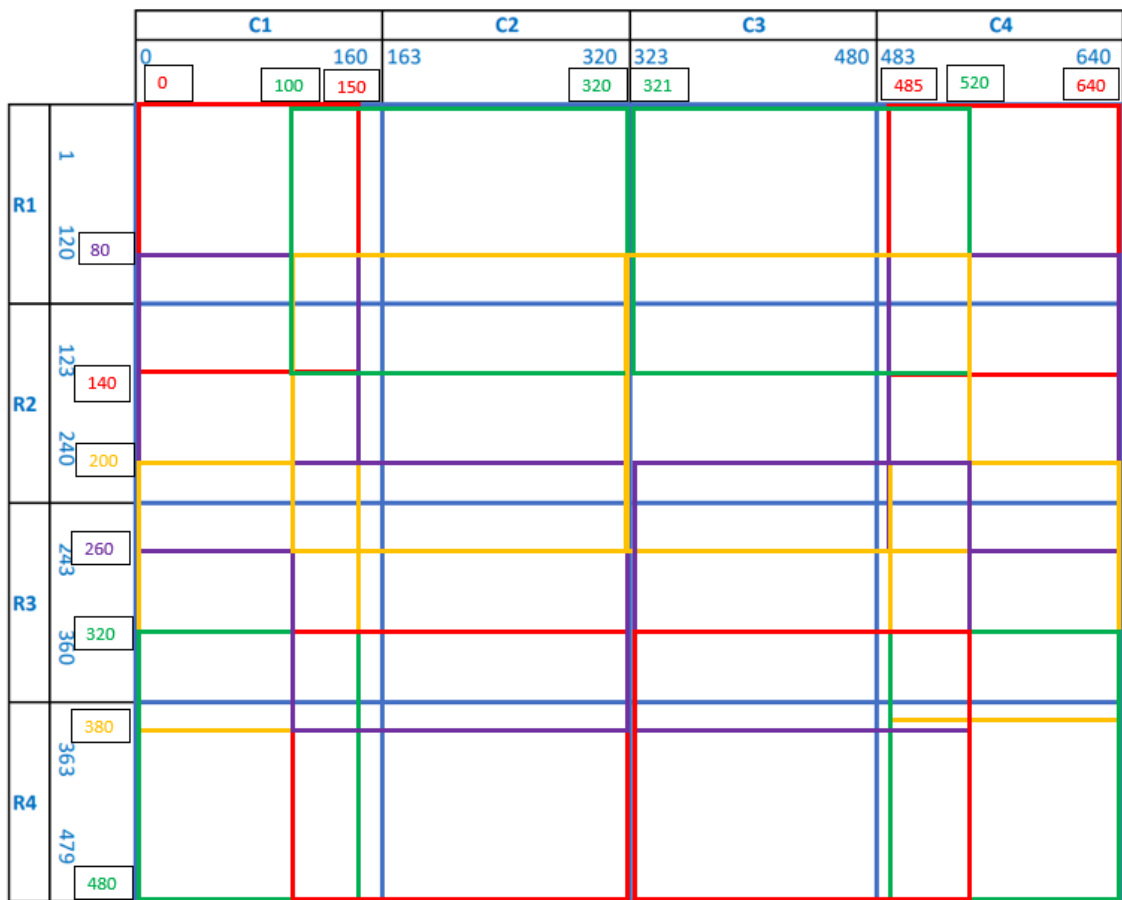


Figura 22. Detección del dron mediante cuadrículas.

3.5. Sistema eléctrico.

Se realizó un sistema eléctrico por medio de un final de carrera y dos sensores. El uso del final de carrera es para que, una vez que el dron llegue a su puesto inicial de trabajo, este lo detecte y envíe una señal, por medio de la cual se enviará un mensaje para que el dron salga de la estación, se procede a pulsar un botón y el dron saldrá para realizar su recorrido en su espacio de trabajo, indicado en la figura 23.

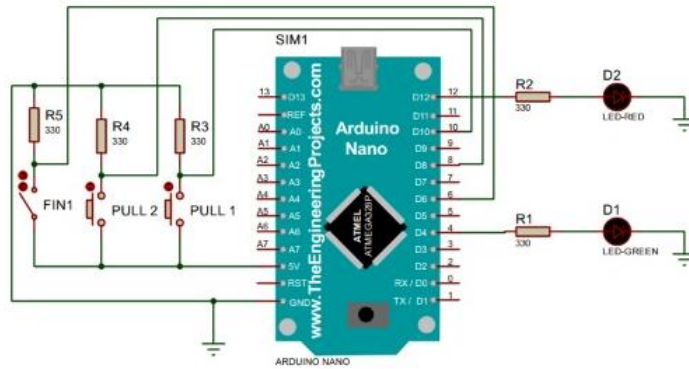


Figura 23. Diagrama eléctrico utilizado.

3.5.1. Diagrama de flujo del funcionamiento del programa de la estación.

A continuación, en la figura 24, se muestra el diagrama que explica el funcionamiento del sistema de la estación, que cuenta de 3 botones de accionamiento, siendo el primero un botón que alerta un error en la estación, un segundo botón que indica que la estación ya a sido reparada, y un tercero que es un final de carrera utilizado debajo de una plancha para saber si el dron se encuentra en la base de salida designada, y si cumple con este requerimiento y la estación provoca un llamado de error, el dron ejecutará el plan de vuelo hacia la estación.

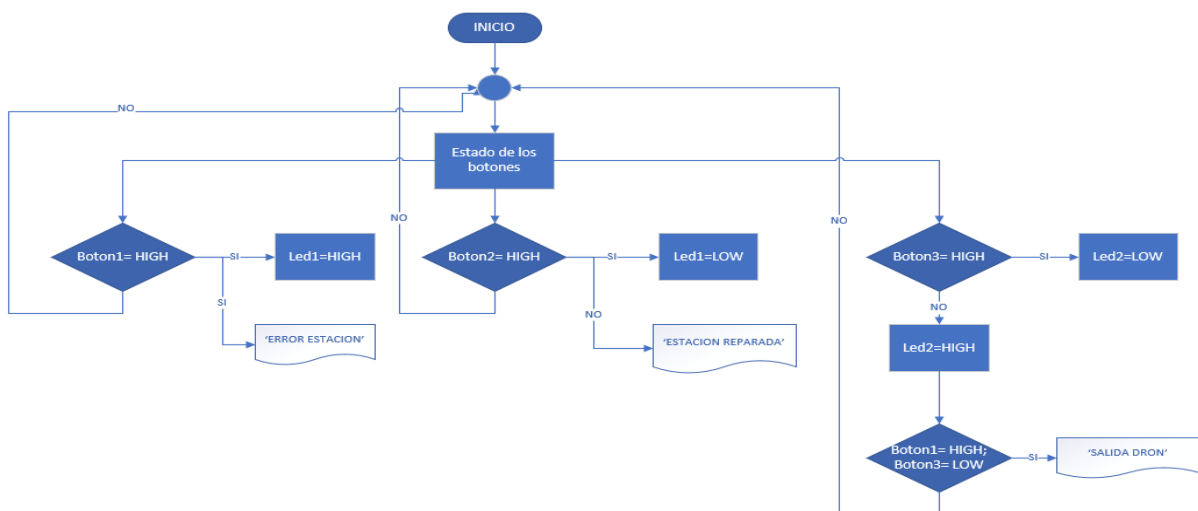


Figura 24. Diagrama de flujo de la estación.

CAPÍTULO 4

4. ANÁLISIS DE RESULTADOS

En este capítulo se presentarán las pruebas realizadas del dron en un área de trabajo, además de los resultados obtenidos mediante la simulación de la hélice en el programa de SolidWorks.

4.1. Análisis de los resultados obtenidos mediante la simulación de la hélice en el programa SolidWorks

Realizando modificaciones en el ángulo de incidencia de la hélice, tanto para su impresión en 3D, así como para su respectiva simulación, se logra determinar la masa, fuerza y velocidad que podría alcanzar la hélice en cada uno de los casos mostrados a continuación.

4.1.1. Ángulo de incidencia 20°

La primera prueba realizada fue con un ángulo de 20°, que es el ángulo más aproximado al ángulo original de la hélice, en donde se puede observar en la figura 25, cabe recalcar que estas tienen dos sentidos, para lograr el vuelo uno en sentido horario y otro en antihorario.



Figura 25. Hélice original del dron.

Como se puede ver en la figura 26 la relación de la hélice con el viento y su velocidad es buena, ya que mueve el aire con una velocidad mayor a la de inicio, y no dejando que este se disperse a través de esta, lo que nos indicaría que podría impulsar al dron a volar sin errores.

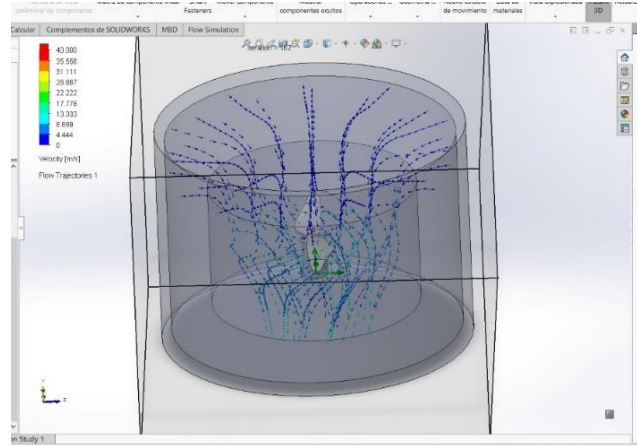


Figura 26. Simulación hélices con ángulo de 20°.

Una vez ejecutado el software de simulación obtenemos la gráfica de iteraciones en relación a la fuerza que produce la hélice al girar con respecto al viento como se ve en la figura 27. Esta tiene ciertos momentos en la iteración en la que se genera fuerzas muy bajas, casi llegando a cero.

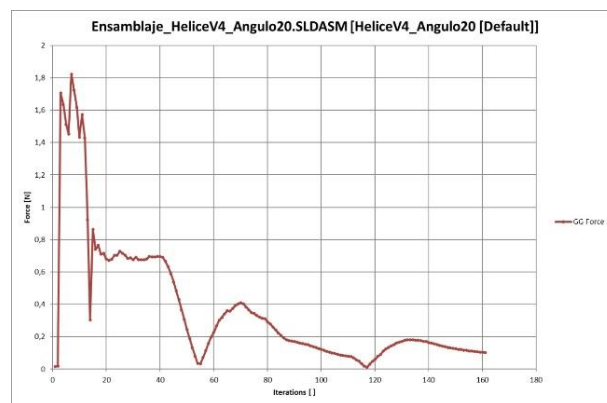


Figura 27. Fuerza producida por la hélice.

Además, nos indica la gráfica de iteraciones en relación a la velocidad que produce la hélice al girar con respecto al viento como se ve en la figura 28. Esta se estabiliza a partir de la iteración 130 aproximadamente.

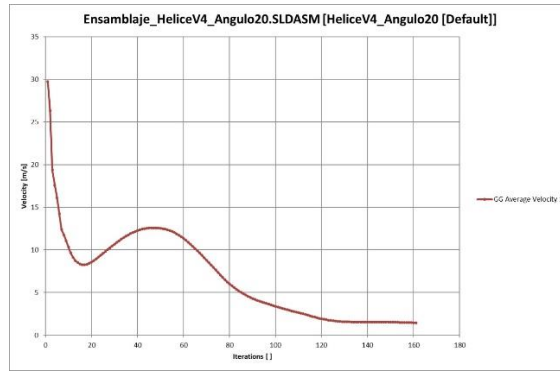


Figura 28. Velocidad producida por la hélice.

El software de simulación nos genera una tabla ponderando todos los cálculos efectuados dentro de la simulación, como se muestra en la tabla 9, para su posterior comparación en los cambios que se van a efectuar.

Tabla 9. Análisis del software para la hélice de 20°.

| Nombre del objetivo | Unidad | Valor | Valor promedio | Valor mínimo | Valor máximo |
|------------------------|--------|-------------|----------------|--------------|--------------|
| Velocidad Media | [m/s] | 1,44761041 | 1,562296806 | 1,44761041 | 1,853916243 |
| Fuerza | [N] | 0,101608105 | 0,140018437 | 0,079786437 | 0,181919298 |
| Masa | [g] | 0.47 | ---- | ---- | ---- |

4.1.2. Ángulo de incidencia 25°

La segunda prueba realizada fue con un ángulo de 25°, esta tuvo que ser impresa en 3D debido a la facilidad de fabricación que esta máquina nos permite, como se puede observar en la figura 29.



Figura 29. Hélice impresa a un ángulo de 25°

Como se puede ver en la Figura 30 la relación de la hélice con el viento y su velocidad es buena, ya que mueve el aire con una velocidad mayor a la de inicio, y no dejando que este se disperse a través de esta, lo que nos indicaría que podría impulsar al dron a volar sin errores.

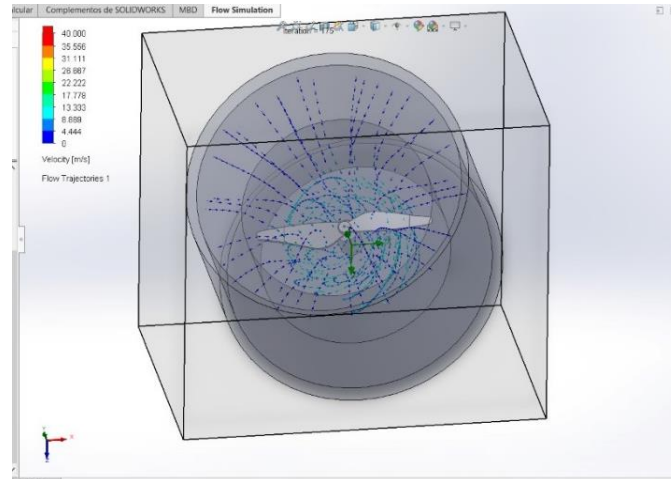


Figura 30. Simulación hélices con ángulo de 25°.

Una vez ejecutado el software de simulación obtenemos la gráfica de iteraciones en relación a la fuerza que produce la hélice al girar con respecto al viento como se ve en la figura 31. Esta tiene ciertos momentos en la iteración en la que se genera fuerzas muy bajas, casi llegando a cero.

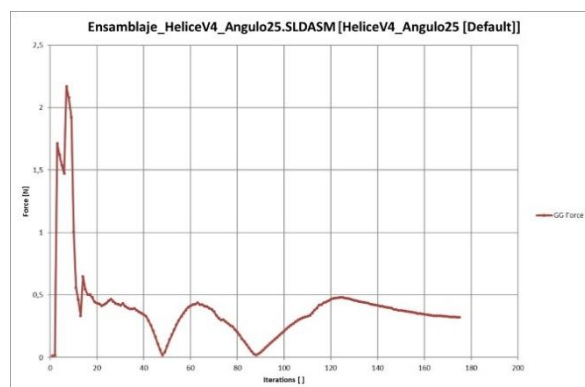


Figura 31. Fuerza producida por la hélice.

Además, nos indica la gráfica de iteraciones en relación a la velocidad que produce la hélice al girar con respecto al viento como se ve en la figura 32. Esta se estabiliza a partir de la iteración 130 aproximadamente.

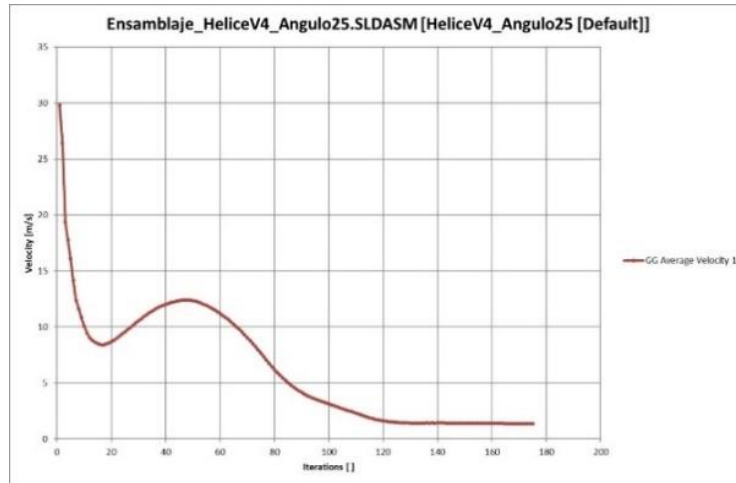


Figura 32. Velocidad producida por la hélice.

El software de simulación nos genera una tabla ponderando todos los cálculos efectuados dentro de la simulación, como se muestra en la tabla 10, para su posterior comparación en los cambios que se van a efectuar.

Tabla 10. Análisis del software para la hélice de 25°.

| Nombre del objetivo | Unidad | Valor | Valor promedio | Valor mínimo | Valor máximo |
|------------------------|--------|-------------|----------------|--------------|--------------|
| Velocidad Media | [m/s] | 1,354763106 | 1,394844619 | 1,354763106 | 1,420954091 |
| Fuerza | [N] | 0,321851473 | 0,367738837 | 0,321851473 | 0,439847775 |
| Masa | [g] | 0.51 | ---- | ---- | ---- |

4.1.3. Ángulo de incidencia 30°

La tercera prueba realizada fue con un ángulo de 30°, esta tuvo que ser impresa en 3D debido a la facilidad de fabricación que esta máquina nos permite, como se puede observar en la Figura 33.



Figura 33. Hélice impresa a un ángulo de 30°.

Como se puede ver en la Figura 34 la relación de la hélice con el viento y su velocidad es buena, ya que mueve el aire con una velocidad mayor a la de inicio, y no dejando que este se disperse a través de esta, lo que nos indicaría que podría impulsar al dron a volar sin errores.

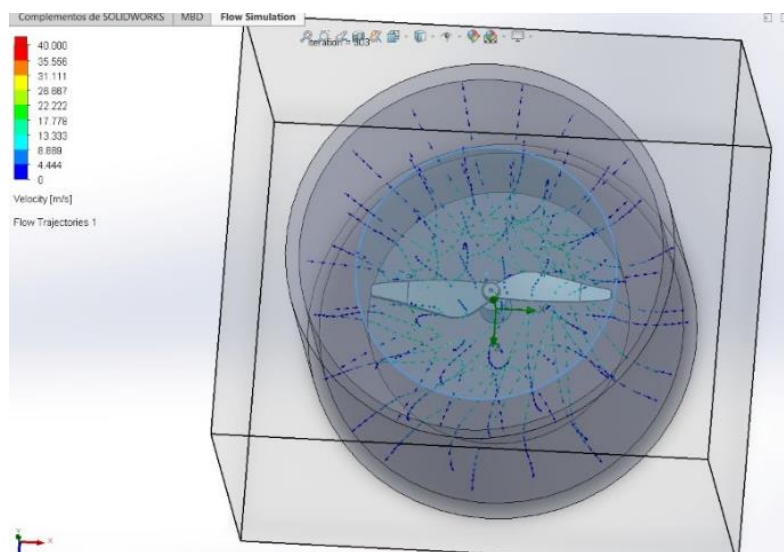


Figura 34. Simulación hélices con ángulo de 30°.

Una vez ejecutado el software de simulación obtenemos la gráfica de iteraciones en relación a la fuerza que produce la hélice al girar con respecto al viento como se ve en la figura 35. Esta tiene ciertos momentos en la iteración en la que se genera fuerzas muy bajas, pero con respecto a las gráficas anteriores, no llega a valores de cero.

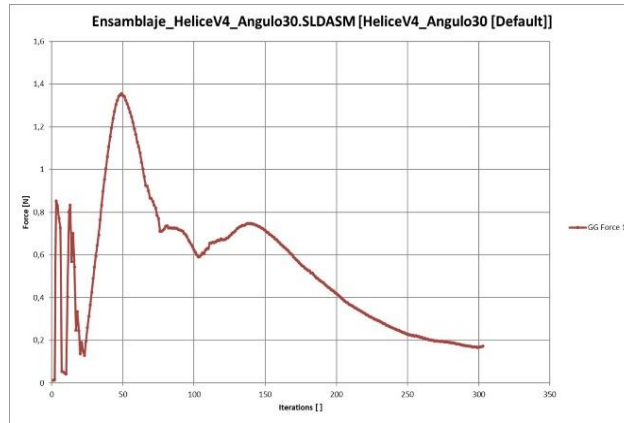


Figura 35. Fuerza producida por la hélice.

Además, nos indica la gráfica de iteraciones en relación a la velocidad que produce la hélice al girar con respecto al viento como se ve en la figura 36. Esta se estabiliza a partir de la iteración 110 aproximadamente.

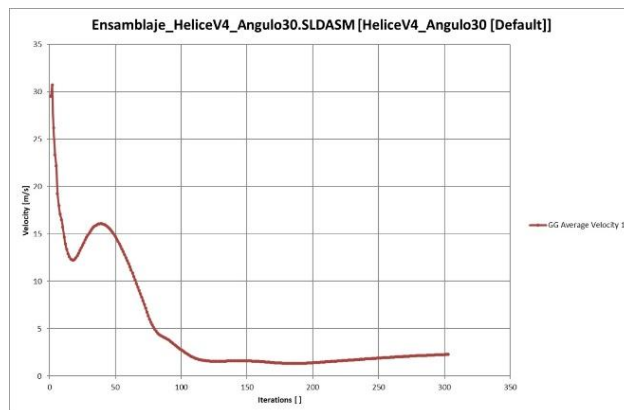


Figura 36. Velocidad producida por la hélice.

El software de simulación nos genera una tabla ponderando todos los cálculos efectuados dentro de la simulación, como se muestra en la tabla 11, para su posterior comparación en los cambios que se van a efectuar.

Tabla 11. Análisis del software para la hélice de 35°.

| Nombre del objetivo | Unidad | Valor | Valor promedio | Valor mínimo | Valor máximo |
|------------------------|--------|-------------|----------------|--------------|--------------|
| Velocidad Media | [m/s] | 2,296453638 | 2,171334764 | 2,026425418 | 2,296453638 |
| Fuerza | [N] | 0,17139411 | 0,184675431 | 0,166962873 | 0,206244012 |
| Masa | [g] | 0.55 | ---- | ---- | ---- |

4.1.4. Ángulo de incidencia 35°

La cuarta prueba realizada fue con un ángulo de 35°, está tuvo que ser impresa en 3D debido a la facilidad de fabricación que esta máquina nos permite, como se puede observar en la figura 37.



Figura 37. Hélice impresa a un ángulo de 35°

Como se puede ver en la Figura 38 la relación de la hélice con el viento y su velocidad es buena, ya que mueve el aire con una velocidad mayor a la de inicio, y no dejando que este se disperse a través de esta, lo que nos indicaría que podría impulsar al dron a volar sin errores.

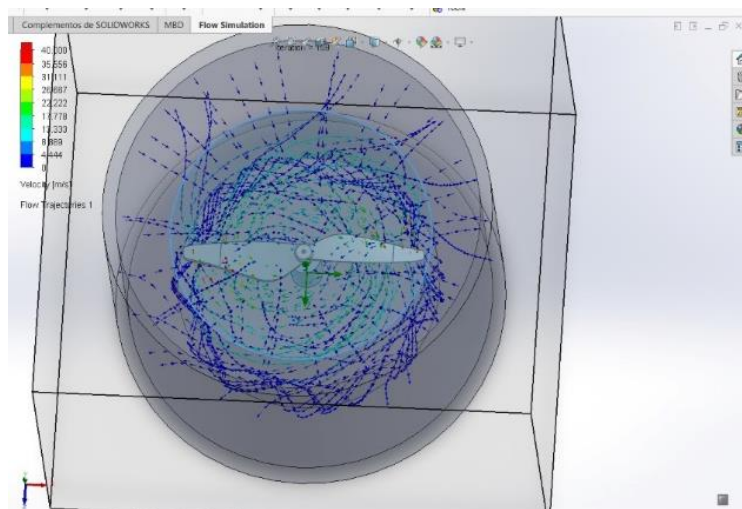


Figura 38. Simulación hélices con ángulo de 35°.

Una vez ejecutado el software de simulación obtenemos la gráfica de iteraciones en relación a la fuerza que produce la hélice al girar con respecto al viento como se ve en la figura 39. Esta tiene ciertos momentos en la iteración en la que se genera fuerzas bajas, pero con respecto a las gráficas anteriores, no llega a valores de cero y es más estable.

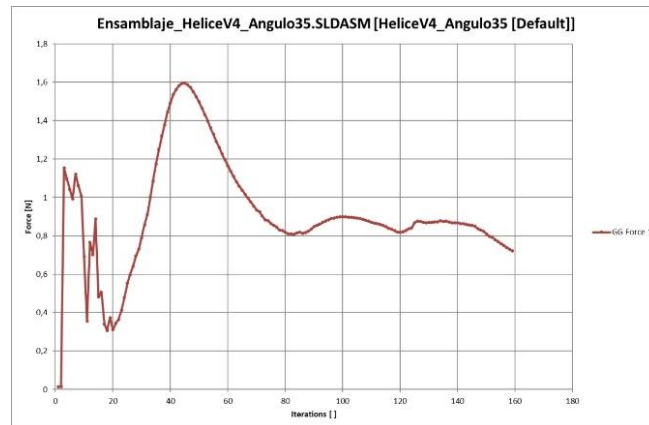


Figura 39. Fuerza producida por la hélice.

Además, nos indica la gráfica de iteraciones en relación a la velocidad que produce la hélice al girar con respecto al viento como se ve en la figura 40. Esta se estabiliza a partir de la iteración 120 aproximadamente.

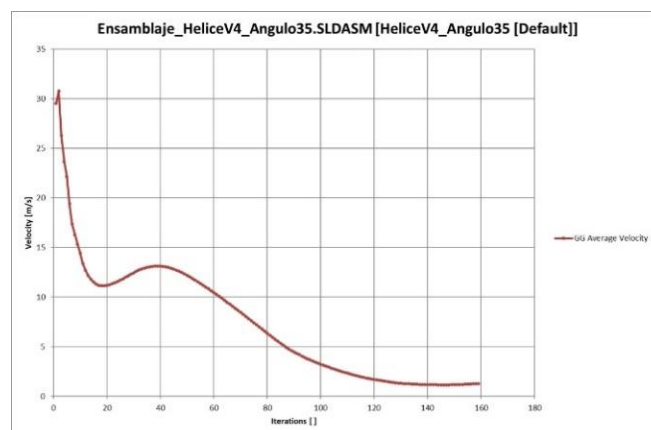


Figura 40. Velocidad producida por la hélice.

El software de simulación nos genera una tabla ponderando todos los cálculos efectuados dentro de la simulación, como se muestra en la tabla 12, para su posterior comparación en los cambios que se van a efectuar.

Tabla 12. Análisis del software para la hélice de 35°.

| Nombre del objetivo | Unidad | Valor | Valor promedio | Valor mínimo | Valor máximo |
|---------------------|--------|-------------|----------------|--------------|--------------|
| Velocidad Media | [m/s] | 1,280039503 | 1,289587953 | 1,155227525 | 1,737583786 |
| Fuerza | [N] | 0,721250336 | 0,834250031 | 0,721250336 | 0,877999728 |
| Masa | [g] | 0.59 | ---- | ---- | ---- |

4.1.5. Ángulo de incidencia 40°

La quinta prueba realizada fue con un ángulo de 40°, esta tuvo que ser impresa en 3D debido a la facilidad de fabricación que esta máquina nos permite, como se puede observar en la figura 41.



Figura 41. Hélice impresa a un ángulo de 40°

Como se puede ver en la figura 42 la relación de la hélice con el viento y su velocidad es muy brusca, ya que mueve el aire con una velocidad mayor a la de inicio, pero esto hace que no tenga una buena disertación a través de esta, lo que nos indicaría que podría impulsar al dron con errores de vuelo.

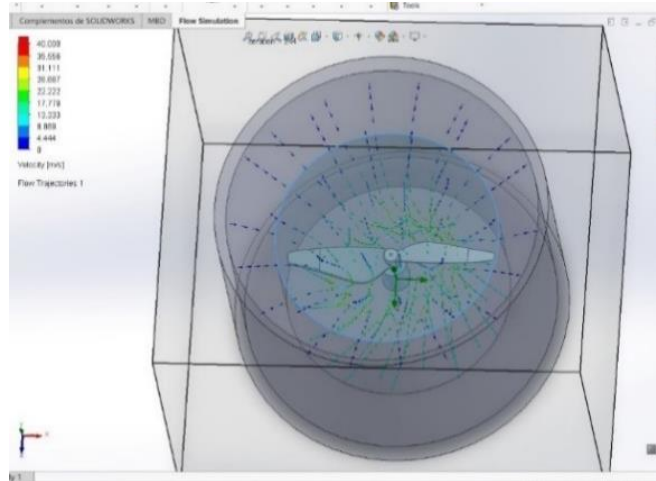


Figura 42. Simulación hélices con ángulo de 40°.

Una vez ejecutado el software de simulación obtenemos la gráfica de iteraciones en relación a la fuerza que produce la hélice al girar con respecto al viento como se ve en la figura 43. Esta tiene ciertos momentos en la iteración en la que se genera fuerzas bajas, pero con respecto a las gráficas anteriores, no llega a valores de cero y es más estable.

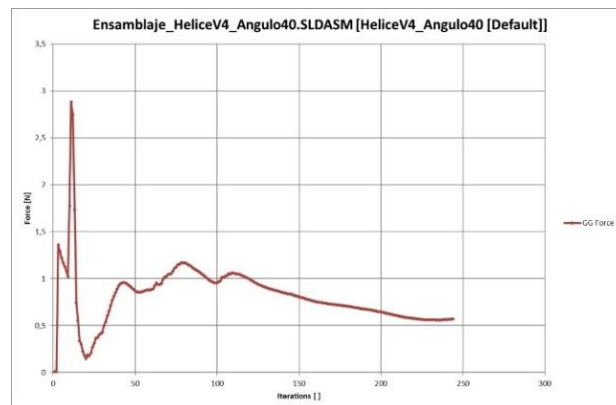


Figura 43. Fuerza producida por la hélice.

Además, nos indica la gráfica de iteraciones en relación a la velocidad que produce la hélice al girar con respecto al viento como se ve en la figura 44. Esta se estabiliza a partir de la iteración 110 aproximadamente con un incremento al final.

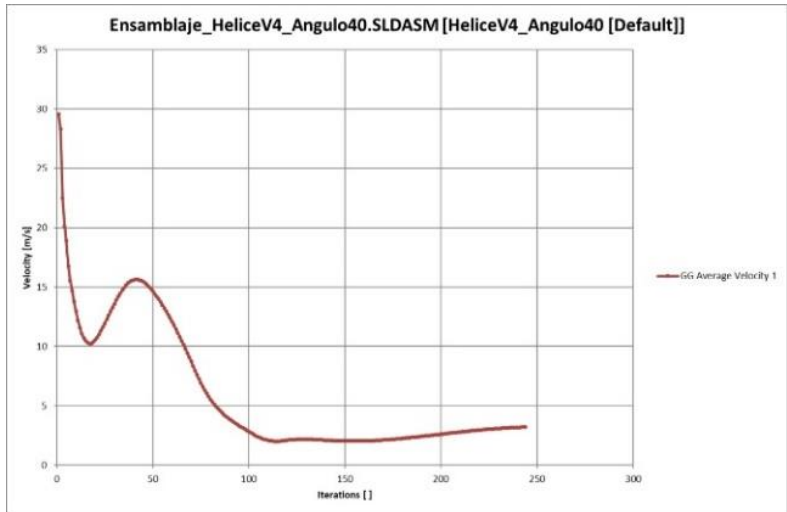


Figura 44. Velocidad producida por la hélice.

El software de simulación nos genera una tabla ponderando todos los cálculos efectuados dentro de la simulación, como se muestra en la tabla 13, para su posterior comparación en los cambios que se van a efectuar.

Tabla 13. Análisis del software para la hélice de 40°.

| Nombre del objetivo | Unidad | Valor | Valor promedio | Valor mínimo | Valor máximo |
|------------------------|--------|-------------|----------------|--------------|--------------|
| Velocidad Media | [m/s] | 3,201224475 | 2,987732572 | 2,680797508 | 3,201224475 |
| Fuerza | [N] | 0,571109201 | 0,579443994 | 0,559394102 | 0,628501732 |
| Masa | [g] | 0.65 | ---- | ---- | ---- |

4.1.6. Ángulo de incidencia 45°

El último análisis de la hélice se lo realizó con un ángulo de 45°, está tuvo que ser impresa en 3D debido a la facilidad de fabricación que esta máquina nos permite, como se puede observar en la figura 45.



Figura 45. Hélice impresa a un ángulo de 45°

Como se puede ver en la figura 46 la relación de la hélice con el viento y su velocidad es muy brusca, ya que mueve el aire con una velocidad mayor a la de inicio, pero esto hace que no tenga una buena disertación a través de esta, lo que nos indicaría que podría impulsar al dron con errores de vuelo.

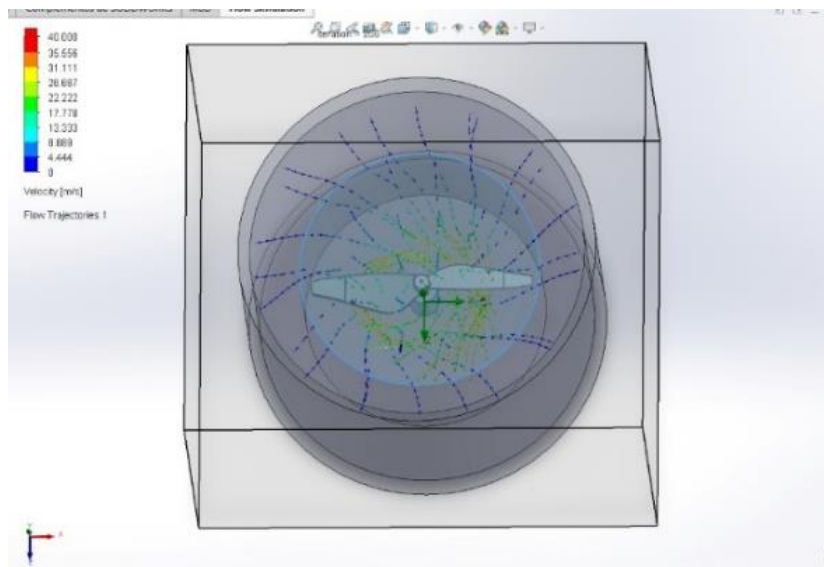


Figura 46. Simulación hélices con ángulo de 40°.

Una vez ejecutado el software de simulación obtenemos la gráfica de iteraciones en relación a la fuerza que produce la hélice al girar con respecto al viento como se ve en la figura 47. Esta

tiene ciertos momentos en la iteración en la que se genera fuerzas muy bajas, pero con respecto a las gráficas anteriores, es más estable.

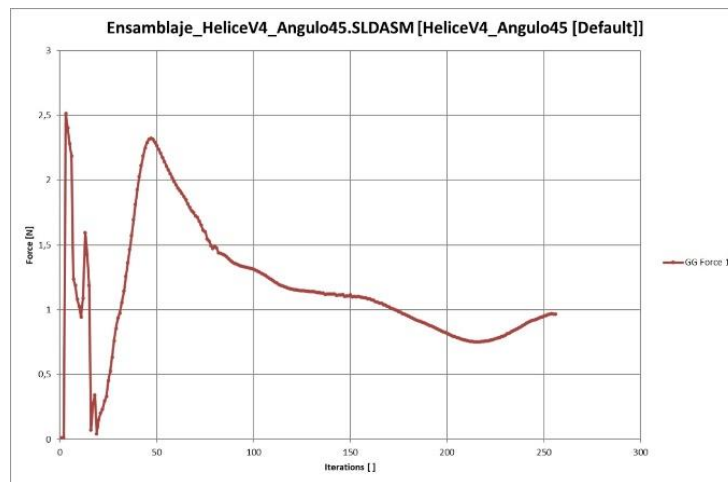


Figura 47. Fuerza producida por la hélice.

Además, nos indica la gráfica de iteraciones en relación a la velocidad que produce la hélice al girar con respecto al viento como se ve en la figura 48. Esta no se estabiliza y cuenta con un incremento al final de las iteraciones.

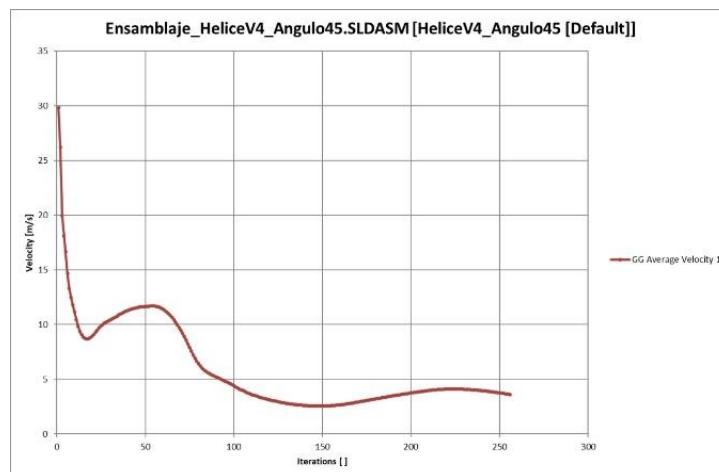


Figura 48. Velocidad producida por la hélice.

El software de simulación nos genera una tabla ponderando todos los cálculos efectuados dentro de la simulación, como se muestra en la tabla 14, para su posterior comparación en los cambios que se van a efectuar.

Tabla 14. Análisis del software para la hélice de 45°.

| Nombre del objetivo | Unidad | Valor | Valor promedio | Valor mínimo | Valor máximo |
|---------------------|--------|-------------|----------------|--------------|--------------|
| Velocidad Media | [m/s] | 3,612948416 | 3,95944762 | 3,612948416 | 4,111420829 |
| Fuerza | [N] | 0,967928845 | 0,855829906 | 0,752518886 | 0,968821329 |
| Masa | [g] | 0.73 | ---- | ---- | ---- |

4.1.7. Análisis de fuerza en la simulación de la hélice.

La simulación en el programa nos entregó varios resultados de fuerzas para cada ángulo que se estudió, los cuales se detallaran en la tabla 15, para su correspondiente comparativa.

Tabla 15. Análisis de la fuerza en la simulación de la hélice.

| Ángulo de la Hélice | Valor Promedio [N] |
|---------------------|--------------------|
| Ángulo de 20° | 0,140018437 |
| Ángulo de 25° | 0,367738837 |
| Ángulo de 30° | 0,184675431 |
| Ángulo de 35° | 0,834250031 |
| Ángulo de 40° | 0,579443994 |
| Ángulo de 45° | 0,855829906 |

4.1.8. Análisis de velocidad en la simulación de la hélice.

La simulación en el programa nos entregó varios resultados de velocidades para cada ángulo que se estudió, los cuales se detallaran en la tabla 16, para su correspondiente comparativa.

Tabla 16. Análisis de la velocidad en la simulación de la hélice.

| Ángulo de la Hélice | Valor Promedio [m/s] |
|---------------------|----------------------|
| Ángulo de 20° | 1,562296806 |
| Ángulo de 25° | 1,394844619 |
| Ángulo de 30° | 2,171334764 |
| Ángulo de 35° | 1,289587953 |
| Ángulo de 40° | 2,987732572 |
| Ángulo de 45° | 3,95944762 |

4.1.9 Análisis de la masa de la hélice.

El programa utilizado tiene un apartado para determinar las propiedades físicas de las piezas, por lo que al variar el ángulo las mediciones en la masa cambiaban, los cuales se detallaran en la tabla 17, para su correspondiente comparativa.

Tabla 17. Análisis de la masa de la hélice con respecto al ángulo.

| Ángulo de la Hélice | Valor [g] |
|----------------------------|------------------|
| Ángulo de 20° | 0.47 |
| Ángulo de 25° | 0.51 |
| Ángulo de 30° | 0.55 |
| Ángulo de 35° | 0.59 |
| Ángulo de 40° | 0.65 |
| Ángulo de 45° | 0.73 |

4.1.10. Análisis de las simulaciones en la hélice.

Con las tablas obtenidas de cada ángulo, se ordenaron los datos que obtuvieron los mejores resultados, esto teniendo en cuenta el peso de la hélice que se obtiene de cada ángulo, como se muestra en la tabla 18.

Tabla 18. Masa en la simulación de la hélice

| Ángulo de la Hélice | Valor [g] |
|----------------------------|------------------|
| Ángulo de 20° | 0.47 |
| Ángulo de 25° | 0.51 |
| Ángulo de 30° | 0.55 |
| Ángulo de 35° | 0.59 |
| Ángulo de 40° | 0.65 |
| Ángulo de 45° | 0.73 |

De igual manera, se ordenaron los datos que obtuvieron los mejores resultados, esta vez teniendo en cuenta el factor de la velocidad de la hélice que se obtiene en cada ángulo estudiado, explicada en la tabla 19.

Tabla 19. Velocidad en la simulación de la hélice

| Ángulo de la Hélice | Valor [m/s] |
|---------------------|-------------|
| Ángulo de 45° | 3,9594476 |
| Ángulo de 40° | 2,9877326 |
| Ángulo de 30° | 2,1713348 |
| Ángulo de 20° | 1,5622968 |
| Ángulo de 25° | 1,3948446 |
| Ángulo de 35° | 1,289588 |

Por último, se ordenaron los datos que obtuvieron los mejores resultados, esta vez teniendo en cuenta la fuerza de la hélice que se obtiene en cada ángulo estudiado, explicada en la tabla 20.

Tabla 20. Fuerza en la simulación de la hélice

| Ángulo de la Hélice | Valor [N] |
|---------------------|-----------|
| Ángulo de 45° | 0,8558299 |
| Ángulo de 35° | 0,83425 |
| Ángulo de 40° | 0,579444 |
| Ángulo de 25° | 0,3677388 |
| Ángulo de 30° | 0,1846754 |
| Ángulo de 20° | 0,1400184 |

Con estos datos obtuvimos que los mejores ángulos a trabajar son los de 25° y 30° y 35°, tanto con respecto a peso, velocidad y fuerza según los datos de las tablas y mediante el estudio que las gráficas de la simulación mostraban en cada etapa de las pruebas se obtiene que el mejor ángulo de incidencia de la hélice es entre 30° y 35°, siendo el primero algo más veloz, pero con

menor estabilidad que el segundo. En el anexo 4 se adjuntó el plano de la hélice con ángulo de 35°, que fue el mas adecuado según el estudio realizado en la hélice.

4.2. Pruebas en la red neuronal de YoloV5.

Al entrenar la red neuronal se debe tener en cuenta varios factores que afectarán a la eficiencia de esta, como lo indica la figura 49; para lo cual se realizaron varias pruebas en el entrenamiento de esta, detalladas a continuación.

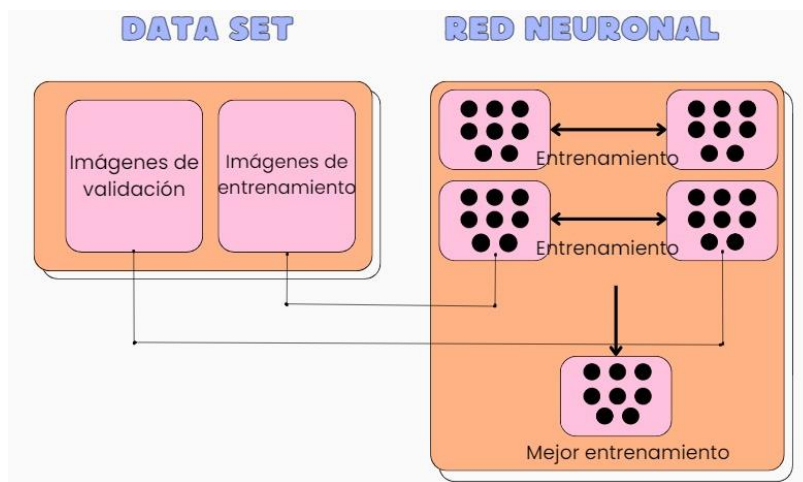


Figura 49. Factores de la red neuronal

4.2.1. Batch size.

Los lotes de las imágenes que tomará nuestra red neuronal es de suma importancia ya que puede generar ciertas discrepancias al este ser menor o mayor.

4.2.1.1. Pruebas con un batch size de 16.

En un entrenamiento de 16 lotes, como se ve en la figura 50, se puede obtener buenos resultados, pero existen ciertas imágenes que no detecta de manera correcta nuestro programa, lo que podría generar posibles fallos a futuro.

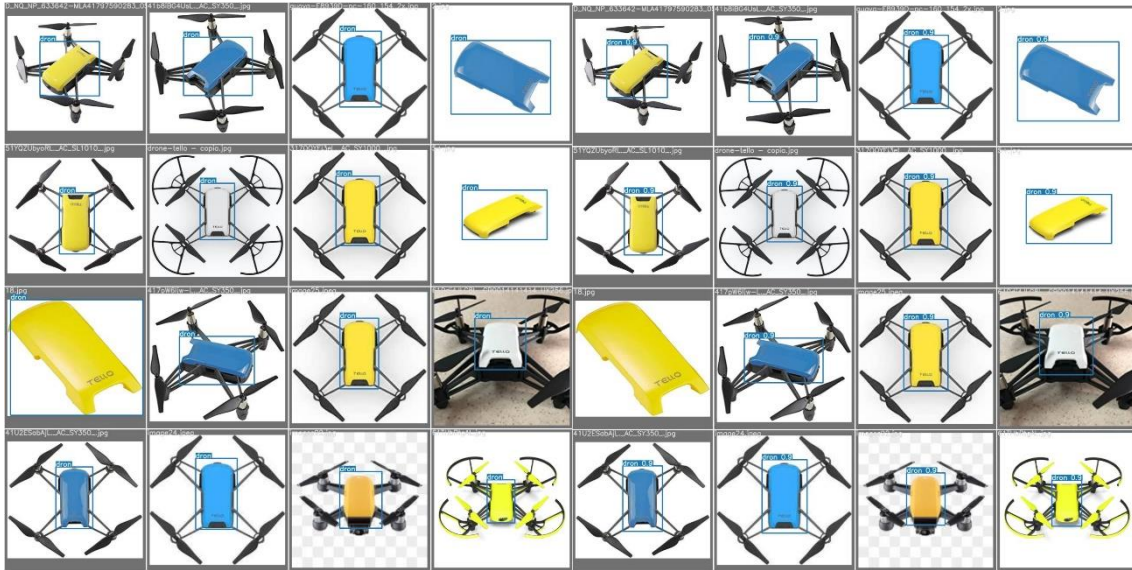


Figura 50. Entrenamiento con un batch size de 16

Como se indica en la figura 51, los resultados son bastante buenos, aunque se puede observar ciertos picos de bajada en la precisión que este genera, así como picos de subida en las pérdidas del objetivo, lo que generaría ciertos inconvenientes en detectar precisamente nuestro dron al ponerlo en el área de pruebas.

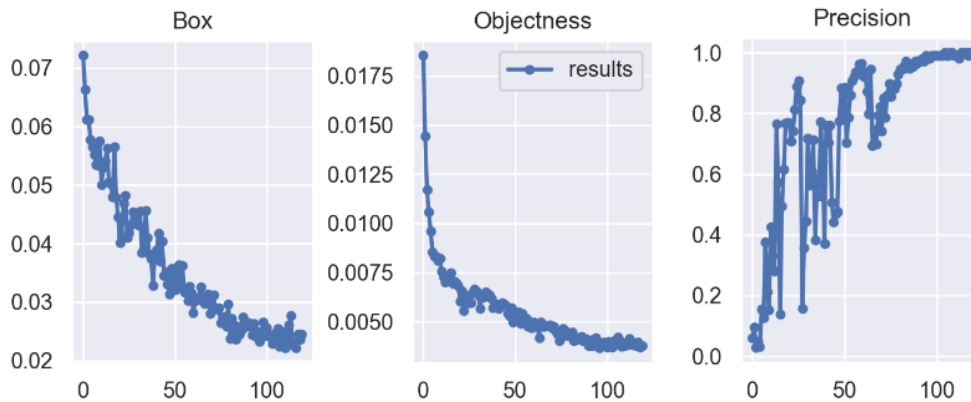


Figura 51. Resultados del entrenamiento con un batch size de 16.

4.2.1.2. Pruebas con un batch size de 8.

En un entrenamiento de 8 lotes, como se ve en la figura 52, se puede obtener mejores resultados, ya que las imágenes que no detectaba anteriormente como dron, ya lo hacen, siendo una significativa mejora para el detector que se está entrenando.

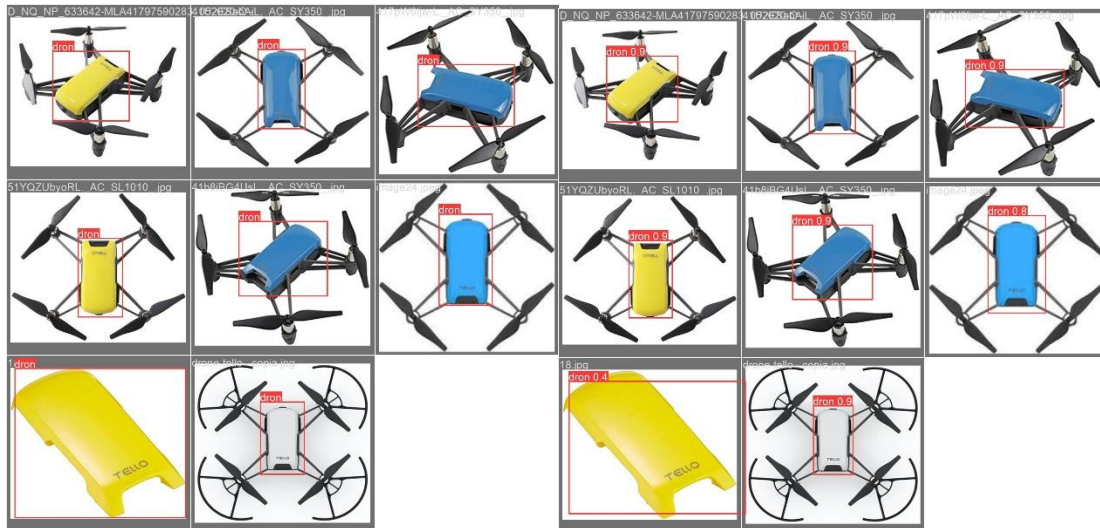


Figura 52. Entrenamiento con un batch size de 8.

Como se indica en la figura 53, los resultados son bastante buenos, aunque se puede observar ciertos picos de bajada en la precisión que este genera, así como picos de subida en las pérdidas del objetivo, pero no son tan bruscos como se venían dando en el anterior entrenamiento.

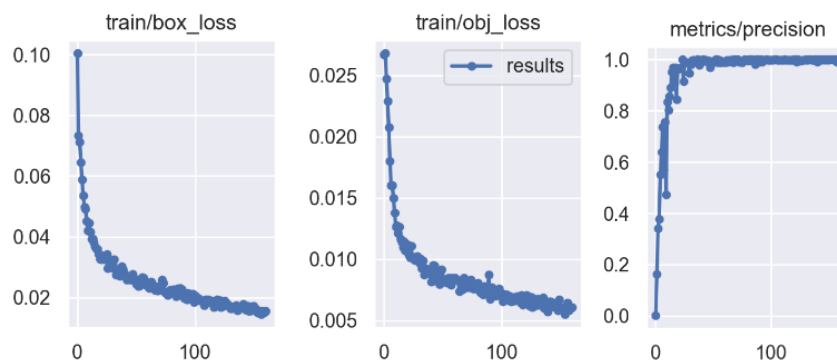


Figura 53. Resultados del entrenamiento con un batch size de 8

4.2.3. Hardware de implementación.

Una vez obtenido la parametrización adecuada para el entrenamiento se implementó una ayuda en el procesamiento de la información mediante Google Colab, ya que esta cuenta con máquinas que tienen una capacidad de procesamiento superior al que ya se vino realizando las pruebas, encontrando así ciertas diferencias en utilizar una computadora de casa, a la máquina en la que se ejecutó el entrenamiento en la página web, siendo esta última superior a la que ya se venía trabajando, lo que generó ciertas mejoras en los resultados del entrenamiento como se puede observar en la figura 54.

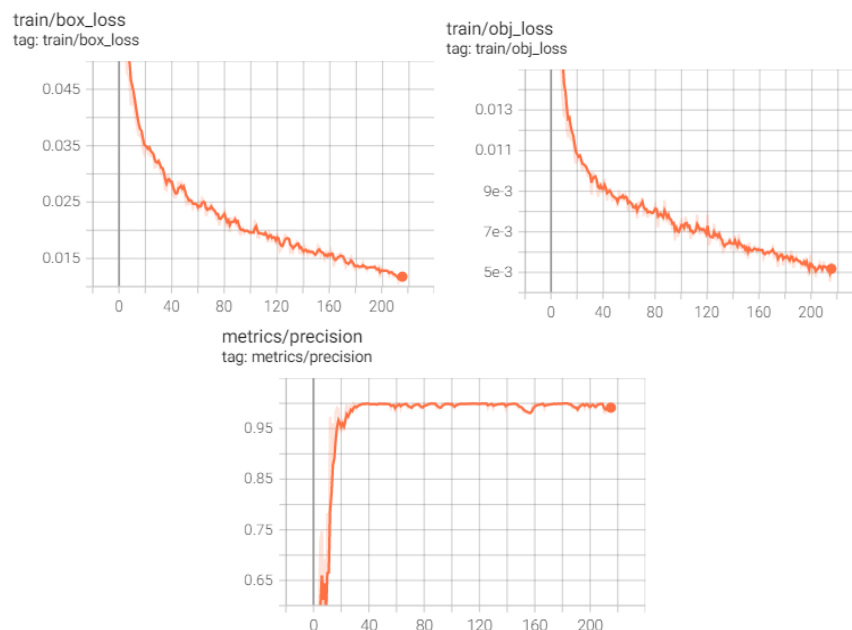


Figura 54. Resultados del entrenamiento en Google Colab.

4.2.4. Épocas.

La cantidad de veces que repita nuestra red neuronal las pruebas, genera mejores resultados en esta, debido a que, mientras más datos tiene para entrenarse, esta indicará una mejor detección de objetos, en este caso el dron que vamos a utilizar, aunque tiene cierta limitante ya que, si el programa después de 100 épocas no genera un aprendizaje significativo, se dejará de entrenar de manera automática para así evitar un overfitting del modelo entrenado.

4.3. Pruebas del dron

Una vez entrenada la red neuronal se puede realizar las correspondientes pruebas para saber el rendimiento del dron con el plan de vuelo en ejecución, para lo cual se tuvo en cuenta el rendimiento de este con respecto a la batería.

4.3.1. Pruebas de la batería del dron en el área de trabajo asignada

Al realizar numerosas pruebas de vuelo con el dron, desde home hasta el punto donde se designó la estación se evidenció ciertas fallas de vuelo cuando la batería se encontraba por debajo del 70-60% de su carga aproximadamente, como se muestra en la figura 55.

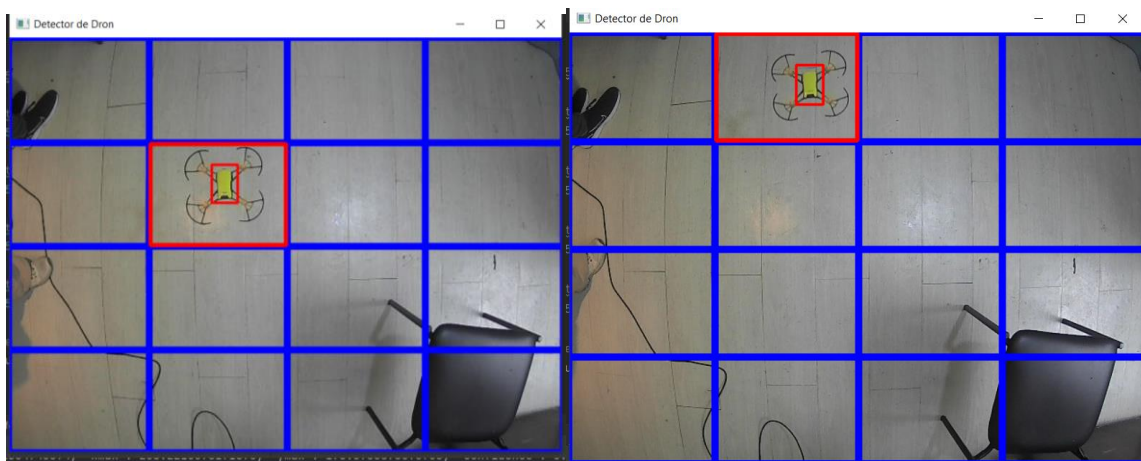


Figura 55. Pruebas del dron con la batería por debajo del 70-60% de carga

Cuando la batería estaba totalmente cargada, es decir al 100% de su capacidad, la prueba que se realizaba en una primera instancia se efectuaba de manera correcta y efectiva, como se muestra en la figura 56.

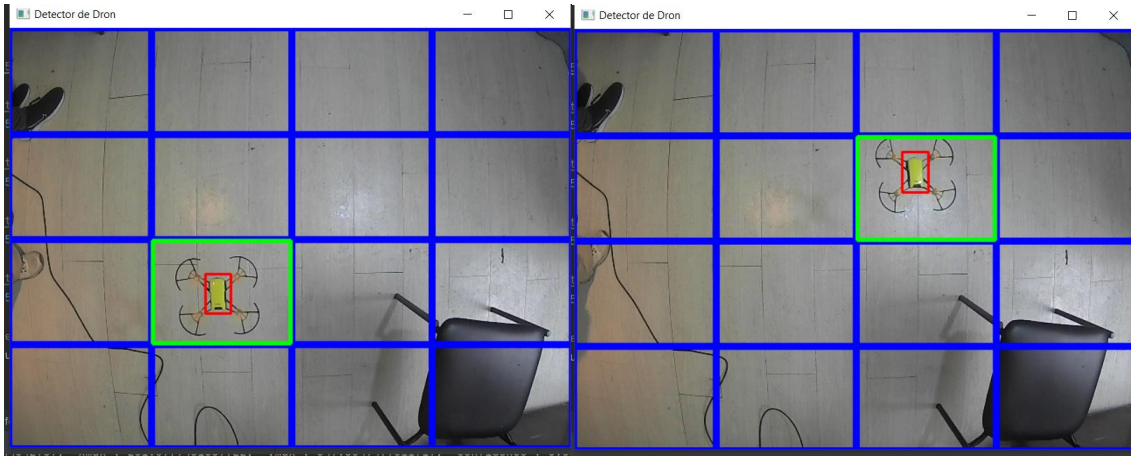


Figura 56. Pruebas del dron con la batería 100% cargada

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Se diseñó un programa que consta de un conjunto de cuadrículas con cuatro columnas y cuatro filas en las que el dron se logre posicionar, el programa logra identificarlo mediante el uso del detector de objetos entrenado con la herramienta de Yolo, la misma que se retroalimentó de imágenes del dron en el área de vuelo designado y que, con las pruebas realizadas en la inferencia de esta, se obtuvo una buena detección del dron DJI Tello. La detección realizada debe de ser indicada en el programa ya que esta tiene que ser visible por el operario para que, mediante un control inicial de vuelo, que se lo realiza desde la posición inicial del dron indicada como home, hacia la posición final de vuelo que es donde está la estación, el programa genera un plan de vuelo que se ejecutará cada vez que ocurra un error en la estación, haciendo que el dron pueda ir las veces que sean necesarias de manera autónoma e indique los errores que se produzcan en la misma.

Se realizó un estudio en las hélices mediante la herramienta de simulación FluidSim, ya que estas son parte fundamental a la hora de que el dron despegue y ejecute el plan de vuelo generado por el programa, indica que al ejercer con un cambio en el ángulo de incidencia de 20° a 35° , el vuelo del dron se vería beneficiado significativamente.

Para lograr que el dron nos indique los errores generados en la estación, se diseñó un programa de detección de fallos, mediante un algoritmo de realidad aumentada basado en OpenCV y marcadores arucos en Python. De esta manera se logra tener una ayuda visual para que el operador aparte de poder observar la estación en tiempo real, pueda tener información adicional de los posibles fallos que tenga la estación.

Al momento de realizar las pruebas de funcionamiento de los programas realizados, se buscó un lugar adecuado para ejecutar las pruebas de vuelo del dron, la detección del mismo mediante el programa, la ejecución del plan de vuelo autónomo cuando la estación indique un error de

funcionamiento, y la detección de fallos mediante realidad aumentada y marcadores arucos; con el fin de determinar si todos los programas cumplen con su propósito designado y ver que cambios se podrán realizar dentro de cada uno de estos para que todo funcione de la manera más idónea posible.

5.2. Recomendaciones

Para que el dron se mantenga funcionando de manera idónea, se necesita que su batería este entre 100% - 60%, debido a que, ya una vez que baja empieza a presentar fallos, tanto en el vuelo como en la detección del marcador aruco, por ende, se debe estar en constante cambio de las baterías por parte del usuario.

Para evitar que la inteligencia artificial genere algún tipo de fallo en la detección del dron, es recomendable entrenarla con la toma de varias imágenes del dron creadas en el Dataset, y pueda ser detectado inmediatamente en la computadora y visualizado por el respectivo operador.

De acuerdo a los análisis realizado de la hélice se determinó que es recomendable utilizar las hélices con un ángulo de incidencia de 35° , ya que, con este ángulo la fuerza de empuje del viento iba a ser mayor y dada sus dimensiones, no iba a tener interferencia con el eje del motor como lo hacían las hélices con grados superiores a esta, a más de ser más pesadas que las hélices originales; al ser un modelo de pruebas, este fue fabricado mediante impresión 3D; por lo que las pruebas en el dron se vieron afectadas por el tipo de fabricación de la misma, ya que esta tenía capas que no están completamente lisas, como la hélice original, si no que estas son líneas unidas una tras otra, provocando un factor de error al momento del vuelo.

El espacio de trabajo del dron debe mantenerse en un ambiente controlado, permitiendo un óptimo funcionamiento de este dispositivo, debido a su tamaño, es fácilmente manipulable por fuerzas externas como el viento, entre otras.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Moreno, D., Toledo, G., Ochoa, A., Torres, V., & Estrada, F. (2021). Evaluation of Drones for Inspection and Control in Industry 4.0. Springer Nature Switzerland, pp. 5-10.
- [2] Yakovlev. K., Makarov. D., & Baskin. S. (2015). Automatic Path Planning for an Unmanned Drone with Constrained Flight Dynamics. *Iskusstvennyi Intellekt i Prinyatie Reshenii*, 2014, No. 4, pp. 3–17.
- [3] Alwater. M., Loke. S., & Niroshinie. F. (2021). Drones-as-a-service: a simulation-based analysis for on-drone decision-making. Springer Nature Switzerland, pp. 3-18.
- [4] Besada. J., Campaña. I., Bergesio. L., Bernardos. A., & Gonzalo. M. Drone flight planning for safe urban operations. Springer Nature Switzerland, pp. 4-18.
- [5] Sellali, B. B., & Allali, A. (2017). Neuro-Fuzzy methods coupled to operational PID, to improve the flight parameters of a drone. 2017 18th.
- [6] Yamamoto, N., & Uchida, N. (2018). Improvement of Image Processing for a Collaborative Security Flight Control System with Multiple Drones. 2018 32nd
- [7] Lozano, V. M. (2015). Sistema de control que permite el vuelo autónomo de drones. Universidad de Alicante.[4] M. Drela, «XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils,» 1989. Obtenido de: http://web.mit.edu/drela/Public/papers/xfoil_sv.p
- [8] Fermin J. (2017). ¿Cómo funcionan las hélices de un dron? Obtenido de: <https://www.comprardrones.online/como-funcionan-las-helices-de-un-drone/>

- [9] Aníbal Carmona (en español). Aerodinámica y actuaciones del avión, 12va. edición (2004). Editorial Paraninfo.
- [10] Valluerca, A. A. (2018). TelefonicaSservicios Audiovisuales. Obtenido de Transmisión de Video y Audio-Protocolos de streaming
- [11] Valluerca, A. A. (2018). TelefonicaSservicios Audiovisuales. Obtenido de Transmisión de Video y Audio-Protocolos de streaming
- [12] BASAÑEZ, L., Y SUÁREZ, R. Teleoperation. In Springer Handbook of Automation, S. Y. Nof, Ed. Springer Berlin Heidelberg, 2009, pp. 449-468.
- [13] NIEMEYER, G., PREUSCHE, C., Y HIRZINGER, G. Telerobotics. In Springer Handbook of Robotics, B. Siciliano and O. Khatib, Eds.
- [14] FURHT, B. Handbook of Augmented Reality. Springer Science+Business Media. Springer, 2011.
- [15] NIEMEYER, G., PREUSCHE, C., Y HIRZINGER, G. Telerobotics. In Springer Handbook of Robotics, B. Siciliano and O. Khatib, Eds. Springer Berlin Heidelberg.
- [16] Descripción del producto MATLAB - MATLAB & simulink - MathWorks américa latina. (2020). MATLAB.
- [17] Santander Universidades. (2021, 27 de octubre). Python: qué es y por qué deberías aprender a utilizarlo. Becas Santander.
- [18] Zhiqiang, W. y Jun, L. Una revisión de la detección de objetos basada en redes neuronales convolucionales. En 2017 36th Chinese Control Conference (CCC) (págs. 11104-11109). IEEE.

- [19] Sultana F., Sufian A., Dutta P. Una revisión de los modelos de detección de objetos basados en redes neuronales convolucionales. *Computación inteligente: aplicaciones basadas en procesamiento de imágenes* (2020), págs. 1-16
- [20] Y. LeCun, Y. Bengio y G. Hinton, "Aprendizaje profundo", *Nature*, vol. 521, págs. 436, 2015.
- [21] Johnson, K. (2021). "Comparación de las características de los IDEs para el desarrollo de aplicaciones en Python". *Revista de Desarrollo de Software*, 15(3), 28-36.
- [22] Rodriguez, C. (2019). "Introducción a las capacidades y aplicaciones del Arduino Nano en proyectos electrónicos". *Revista de Robótica y Electrónica*, 5(2), 12-20.
- [23] Smith, J. (2017). "Aplicaciones de los Sensores de Final de Carrera en la Automatización Industrial". *Revista de Automatización y Control*, 7(1), 45-50.
- [24] Brown, M. (2018). "Mejoras en la Transmisión de Datos a través de Cables". *Revista de Tecnología de Datos*, 8(4), 56-62.
- [25] Johnson, K. (2015). "Avances en la Tecnología LED". *Revista de Iluminación*, 15(2), 20-25.
- [26] Smith, J. (2010). "Resistencias en Circuitos Eléctricos". *Revista de Electrónica*, 23(5), 46-52.
- [27] Grob, B. E., & Bube, R. R. (2011). *Basic Electronics: Theory and Practice* (7th ed.). McGraw-Hill Education.
- [28] Redmon, J., & Farhadi, A. (2021). YOLOv5: Real-Time Object Detection. arXiv preprint arXiv:2104.13650.

- [29] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). TensorFlow: Large-scale machine learning on heterogeneous systems.
- [30] Viola, P., & Jones, M. J. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001 (Vol. 1, pp. I-511-I-518). IEEE.
- [31] Bishop, C. M. (1995). Neural networks for pattern recognition. Oxford University Press.

ANEXOS

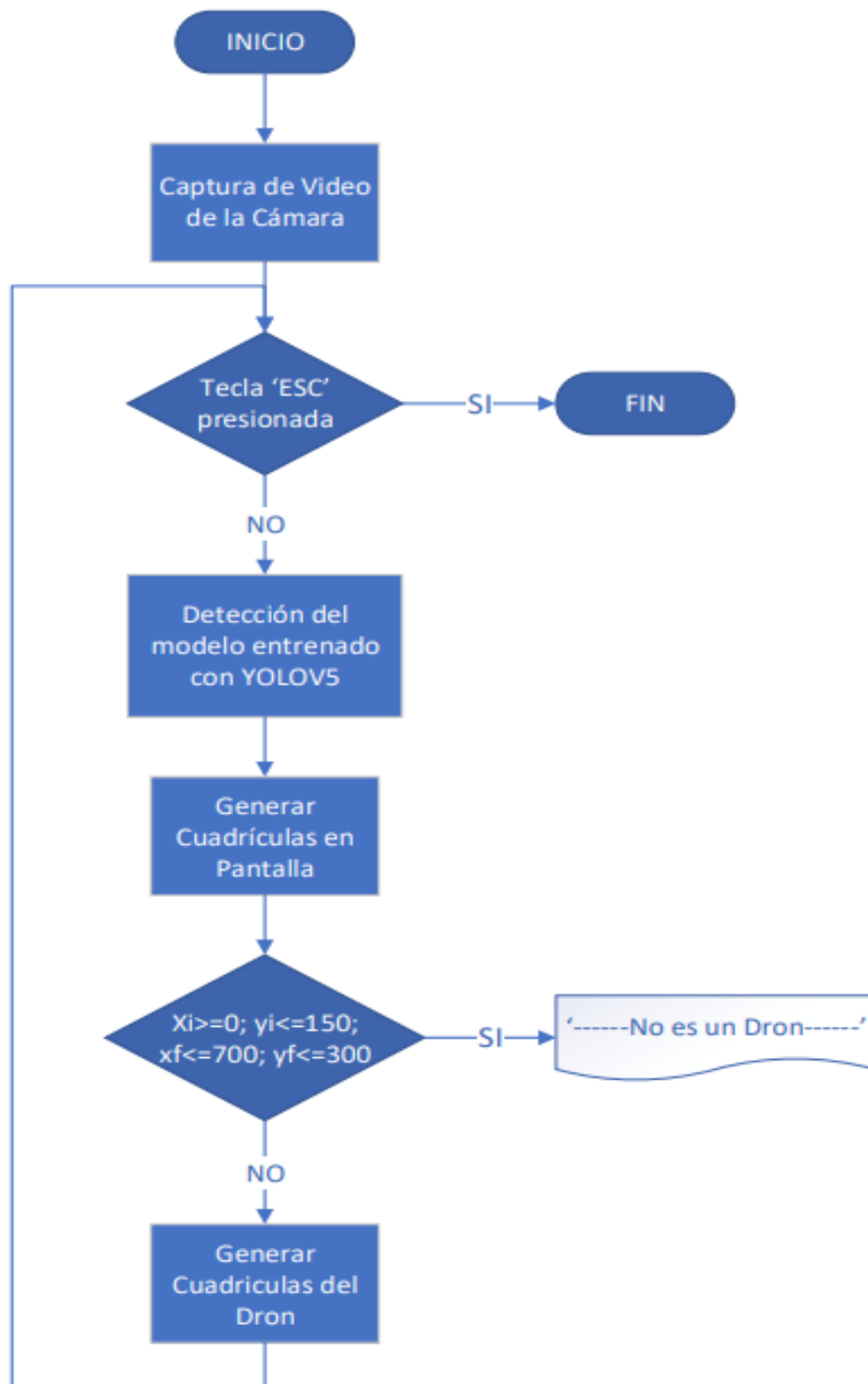
ANEXO 1

Tabla de especificaciones de dron DJI Tello.

| Especificaciones | |
|--|------------------------|
| Peso (Incluyendo las hélices) | 87 g |
| Velocidad máxima | 17,8 mph |
| Tiempo máximo de vuelo | 13 minutos |
| Rango de temperatura de funcionamiento | 32° - 104°F |
| Rango de frecuencia de movimiento | 2,4-2,4835 GHz |
| Transmisor (PIRE) | 20 dBm |
| Cámara | |
| Tamaño máximo de imagen | 2592x1936 |
| Modos de grabación de video | HD: 1280x720 30p |
| Formato de video | MP4 |
| Batería de vuelo | |
| Capacidad | 1100 mAh |
| Voltaje | 3,8 V |
| Tipo de batería | LiPo |
| Energía | 4,18Wh |
| Peso neto | 25g |
| Rango de temperatura de carga | 41°-113°F |
| Potencia máxima de carga | 10 W |

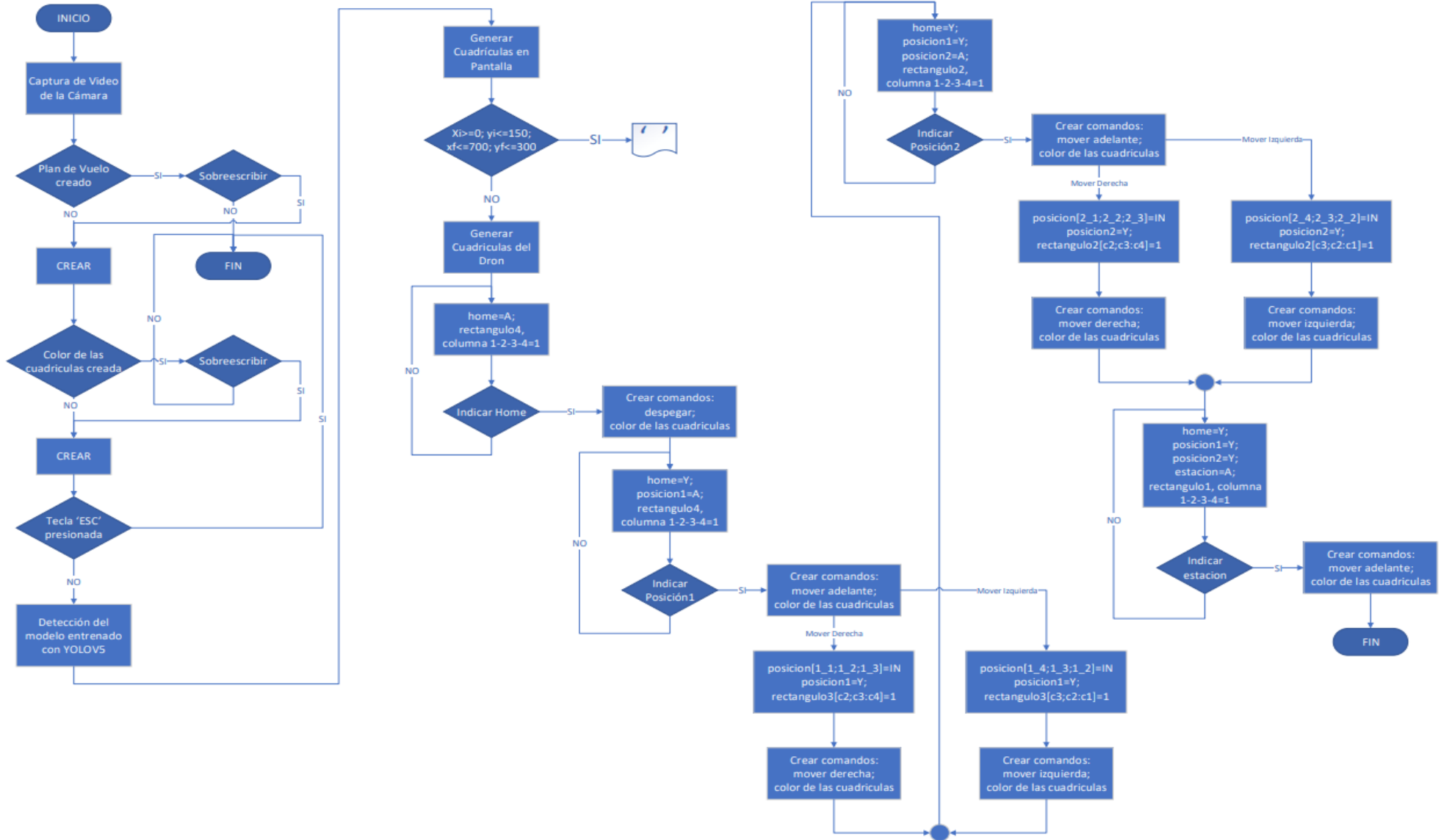
ANEXO 2

Diagrama de flujo del detector de objetos, aplicado al dron.



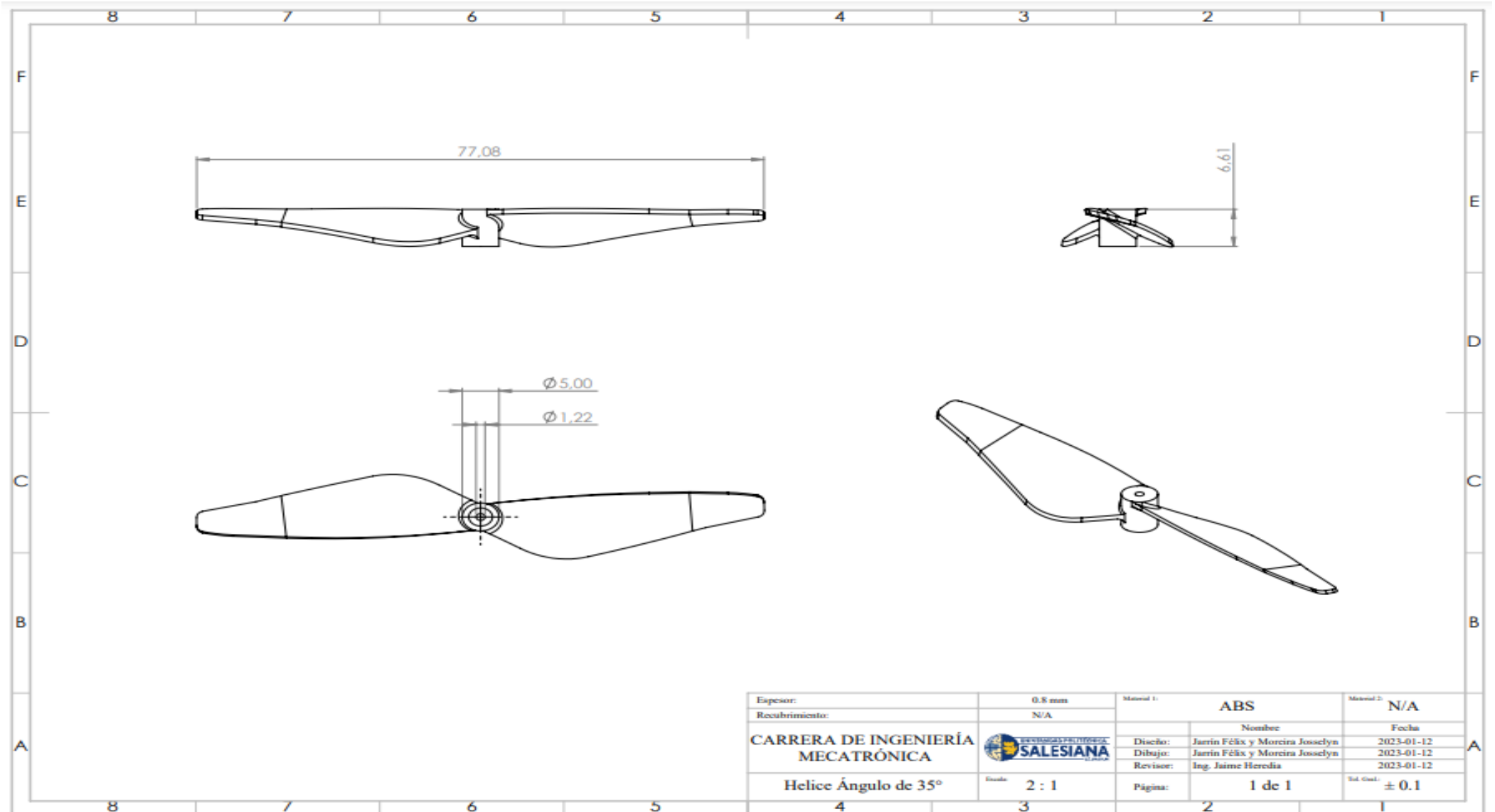
ANEXO 3


Diagrama de flujo de la implementación del plan de vuelo desde home.



ANEXO 5

Plano de la hélice con ángulo de 35°.



| | | | | | |
|--------------------------------------|---|---|--------|------------------|-----|
| Epesor: | 0.8 mm | Material 1: | ABS | Material 2: | N/A |
| Recubrimiento: | N/A | | | | |
| CARRERA DE INGENIERÍA MECATRÓNICA |  | Nombre | | Fecha | |
| | | Diseño: Jarrín Félix y Morcira Josselyn | | 2023-01-12 | |
| | | Dibujó: Jarrín Félix y Morcira Josselyn | | 2023-01-12 | |
| | | Revisó: Ing. Jaime Heredia | | 2023-01-12 | |
| Helice Ángulo de 35° | Escala: 2 : 1 | Página: | 1 de 1 | Tol. Genl: ± 0.1 | |