



**UNIVERSIDAD POLITÉCNICA SALESIANA**  
**SEDE CUENCA**  
**CARRERA DE MECATRÓNICA**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE  
RECONOCIMIENTO AUTÓNOMO PARA EL SECTOR RURAL  
MEDIANTE UN DRONE**

Trabajo de titulación previo a la obtención  
del título de Ingeniero en Mecatrónica

AUTORES: FABRIZIO ISRAEL TORRES PINTADO  
JUAN ANDRÉS LLIGUISACA SEGARRA

TUTOR: ING. DIEGO PAUL CHACÓN TROYA, MDHD.

Cuenca – Ecuador

2023

# CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Juan Andrés Lliguisaca Segarra con documento de identificación N° 0105664635 y Fabrizio Israel Torres Pintado con documento de identificación N° 0106874837; 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.

Cuenca, 25 de julio del 2023

Atentamente,

---

Juan Andrés Lliguisaca Segarra  
0105664635

---

Fabrizio Israel Torres Pintado  
01068748370

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

Nosotros, Juan Andrés Lliguisaca Segarra con documento de identificación N° 0105664635 y Fabrizzio Israel Torres Pintado con documento de identificación N° 0106874837, 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 autores del Proyecto Técnico: "Diseño e implementación de un sistema de reconocimiento autónomo para el sector rural mediante un drone", el cual ha sido desarrollado para optar por el título de: Ingeniero en Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 25 de julio del 2023

Atentamente,

---

Juan Andrés Lliguisaca Segarra  
0105664635

---

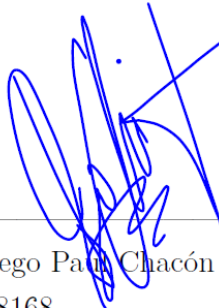
Fabrizzio Israel Torres Pintado  
0106874837

# **CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN**

Yo, Diego Paul Chacón Troya con documento de identificación N° 1900268168, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE RECONOCIMIENTO AUTÓNOMO PARA EL SECTOR RURAL MEDIANTE UN DRONE, realizado por Juan Andrés Lliguisaca Segarra con documento de identificación N° 0105664635 y Fabrizio Israel Torres Pintado con documento de identificación N° 0106874837, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 25 de julio del 2023

Atentamente,

A handwritten signature in blue ink, consisting of several loops and a long horizontal stroke, positioned above a horizontal line.

Ing. Diego Paul Chacón Troya, Mdhd.  
1900268168

# Dedicatoria

## ***Juan Andrés***

Gracias Dios por cada uno de los logros que he conseguido en mi vida y permitirme compartir con los seres que amo con todas mis fuerzas.

Dedico este proyecto de titulación a mis padres y a cada miembro de mi familia que siempre me apoyaron con el objetivo de verme convertido en un profesional.

Quiero dedicar a la persona más especial en mi vida, Adriana que siempre estuvo en los buenos y malos momentos apoyándome absolutamente en todo, cumpliéndole la promesa que nos hicimos un día de convertirnos en profesionales y tener una vida juntos con una familia bajo la bendición de Dios.

## ***Fabrizzio Israel***

Quiero dedicar este trabajo de titulación a mis padres y hermanos, que han sido un pilar fundamental en mi vida y por haberme brindado su cariño, paciencia y apoyo incondicional, gracias a ellos he logrado cumplir uno de mis objetivos planteados. Nuevamente gracias por estar en cada momento que los necesite.

A mis amigos que siempre han depositado su fé y apoyo en que logre cumplir todas mis metas, y ha todas las personas que formaron parte este proceso y me brindaron de una u otra forma su apoyo.

Finalmente, quiero dedicar este trabajo a Dios, quién ha sido mi guía, fortaleza y que siempre me ha acompañado y me acompañara en cada paso que dé, para llevar a cabo cada una de mis metas.

# **Agradecimientos**

## ***Juan Andrés***

Quiero agradecer a Dios por haberme permitido cumplir un objetivo más en mi vida, por brindarme salud, sabiduría y las fuerzas necesarias para no rendirme y seguir adelante en los momentos más duros y difíciles.

Agradezco a mis padres Eloy y Grimanesa, por siempre apoyarme en mi estudio y con sus consejos darme la fortaleza suficiente para culminar este proyecto. A mis hermanas por darme su apoyo absoluto. De igual manera a las personas que siempre estuvieron conmigo a mis amigos de aula con los cuales compartí momentos de alegría y tristezas de cada ciclo estudiantil y de manera especial a mi enamorada Adriana, por siempre estar conmigo en todo el camino de mi vida estudiantil apoyándome incondicionalmente.

Finalmente agradecer a mi tutor Ing. Diego Chacón, por su paciencia y ayuda necesaria en cada uno de los momentos de la elaboración de este proyecto.

## ***Fabrizio Israel***

Agradezo principalmente a Dios por haberme brindado salud y dado las fuerzas para seguir adelante y poder realizar este proyecto de titulación. A mis padres Manuel y Claudia, quiero agradecerles por siempre brindarme sus palabras de aliento a que siga superandome en la vida, y con su ejemplo de superación y dedicación se puede cumplir todos mis objetivos. A mi hermano y hermana por esta conmigo en todo momento y darme su apoyo incondicional.

Finalmente quiero agradecer a mi Tutor de Proyecto de Titulación Ing. Paul Chacón, por su guía y dedicación brindados al actual trabajo. Sus conocimientos y orientaciones han sido un pilar fundamental para culminar el presente trabajo.

Este documento fue realizado enteramente en L<sup>A</sup>T<sub>E</sub>X

# Índice

<b>Certificado de responsabilidad y autoría del trabajo de titulación</b>	<b>I</b>
<b>Certificado de cesión de derechos de autor del trabajo de titulación a la Universidad Politécnica Salesiana</b>	<b>II</b>
<b>Certificado de dirección del trabajo de titulación</b>	<b>III</b>
<b>Dedicatoria</b>	<b>IV</b>
<b>Agradecimientos</b>	<b>V</b>
<b>Resumen</b>	<b>XIV</b>
<b>Abstract</b>	<b>XV</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Problema</b>	<b>2</b>
2.1. Antecedentes . . . . .	2
2.2. Descripción del problema . . . . .	2
2.3. Importancia y alcances . . . . .	3
2.4. Delimitación . . . . .	3
2.4.1. Espacial o geográfica . . . . .	3
2.4.2. Temporal . . . . .	4
2.5. Problema General . . . . .	4
2.6. Problemas Específicos . . . . .	4
<b>3. Objetivos</b>	<b>4</b>
3.1. Objetivo General . . . . .	4
3.2. Objetivos Específicos . . . . .	4
<b>4. Marco Teórico</b>	<b>5</b>
4.1. Drone . . . . .	5
4.2. Tipos de drones por su definición . . . . .	5
4.3. Tipología de drones por número de brazos . . . . .	6
4.4. Partes de un drone . . . . .	10



4.4.1. Marco (frames) . . . . .	10
4.4.2. Batería . . . . .	10
4.4.3. Motores y hélices . . . . .	11
4.4.4. Radio receptor (mando) . . . . .	12
4.4.5. Sistema de posicionamiento global “GPS” . . . . .	13
4.4.6. Visión en primera persona “visores de realidad virtual o FPV” . . . . .	13
4.4.7. Controlador de vuelo/ placa controladora . . . . .	14
4.4.8. Estabilizador Gimbal . . . . .	15
4.4.9. Sensores a bordo de los drones – Cámara . . . . .	15
4.5. Drones programables . . . . .	16
<b>5. Marco metodológico</b>	<b>18</b>
5.1. Diagrama de funcionamiento del sistema de reconocimiento autónomo mediante un drone . . . . .	18
5.2. Fase 1 Diseñar el sistema de control autónomo. . . . .	19
5.2.1. Python . . . . .	21
5.2.2. Librería de Python . . . . .	21
5.2.3. Calibración de la cámara del drone . . . . .	21
5.2.4. Aruco . . . . .	24
5.2.5. Marcadores . . . . .	25
5.2.6. Código generado para marcador aruco . . . . .	25
5.3. Fase 2 Elaboración del sistema de reconocimiento. . . . .	29
5.3.1. Código . . . . .	29
5.4. Fase 3 Elaboración del sistema de comunicación. . . . .	33
5.4.1. Sensor PIR . . . . .	33
5.4.2. Módulo ESP32 . . . . .	33
5.4.3. Código . . . . .	34
5.5. Fase 4 Implementación software de monitoreo. . . . .	38
5.5.1. Librerías importadas . . . . .	38
<b>6. Resultados</b>	<b>42</b>
6.1. Sistema de control autónomo. . . . .	42
6.1.1. Sistema de control autónomo en zona interior de la vivienda. . . . .	43
6.1.2. Sistema de control autónomo en zona exterior . . . . .	47
6.2. Sistema de reconocimiento. . . . .	52

6.2.1. Sistema de reconocimiento en zona interior de la vivienda. . . . .	52
6.2.2. Sistema de reconocimiento en zona exterior de la vivienda. . . . .	54
6.3. Sistema de comunicación. . . . .	56
6.3.1. Sistema de comunicación en zona interior de la vivienda. . . . .	56
6.3.2. Sistema de comunicación en zona exterior de la vivienda. . . . .	58
6.4. Implementación software de monitoreo. . . . .	62
6.4.1. Funcionamiento del software de monitoreo en zona interior de la vivienda.	62
6.4.2. Funcionamiento del software de monitoreo en zona exterior de la vivienda.	64
<b>7. Conclusiones</b>	<b>65</b>
<b>8. Recomendaciones</b>	<b>66</b>
<b>Referencias</b>	<b>69</b>
<b>ANEXOS</b>	<b>70</b>

# Lista de Tablas

- 1. Características de drones de programación . . . . . 17
- 2. Resultado de las pruebas realizadas de la trayectoria 1 . . . . . 46
- 3. Resultado de las pruebas realizadas de la trayectoria 2 . . . . . 46
- 4. Resultado de las pruebas realizadas de la trayectoria 1 . . . . . 51
- 5. Resultado de las pruebas realizadas de la trayectoria 2 . . . . . 51
- 6. Resultado de las pruebas realizadas del sistema de reconocimiento . . . . . 54
- 7. Resultado de las pruebas realizadas del sistema de reconocimiento . . . . . 56
- 8. Resultado de las pruebas realizadas del sistema de comunicación . . . . . 62
- 9. Estados del drone . . . . . 71
- 10. Características técnicas del drone . . . . . 72

## Lista de Figuras

1.	Tricópetero . . . . .	6
2.	Cuadricópetero . . . . .	7
3.	Hexacópetero . . . . .	7
4.	Octocóptero . . . . .	8
5.	Coaxial . . . . .	9
6.	Marco . . . . .	10
7.	Batería 6 celdas . . . . .	11
8.	Motor . . . . .	11
9.	Hélices . . . . .	12
10.	Control genérico. . . . .	12
11.	Sensor GPS . . . . .	13
12.	Visor de realidad virtual . . . . .	14
13.	Módulo ardupilot . . . . .	14
14.	Estabilizador Gimbal . . . . .	15
15.	Cámara . . . . .	16
16.	Diagrama de funcionamiento . . . . .	19
17.	Drone DJI Tello . . . . .	20
18.	Cuadrícula de 9x6 . . . . .	23
19.	Cuadrícula con patrones generados . . . . .	23
20.	Resultados de la matriz de calibración . . . . .	24
21.	Marcador estándar . . . . .	24
22.	Marcador aruco impreso . . . . .	25
23.	Librerías . . . . .	26
24.	Variables Tello . . . . .	26
25.	Variable Aruco . . . . .	27
26.	Detección bordes de marcador aruco . . . . .	27
27.	Coordenadas de posición del Aruco con respecto a la cámara del drone . . . . .	28
28.	Proyección de los ejes de referencia . . . . .	28
29.	Importación de librerías . . . . .	29
30.	ID de cámara . . . . .	29
31.	Línea de código de objetos comunes en contexto . . . . .	30
32.	Clases de datos . . . . .	31
33.	Línea de código del contorno de la imagen. . . . .	32

34.	Objetos detectados . . . . .	32
35.	Sensor PIR . . . . .	33
36.	Módulo ESP32 . . . . .	34
37.	Registro de red y contraseña . . . . .	34
38.	Condiciones de la activación de alarma . . . . .	35
39.	BotFather . . . . .	35
40.	Creación del bot . . . . .	36
41.	Nombre del bot creado . . . . .	36
42.	Token API . . . . .	36
43.	ID del usuario . . . . .	37
44.	Ingreso del token e ID en el código . . . . .	37
45.	Alerta de movimiento detectado . . . . .	38
46.	Librerías importadas . . . . .	38
47.	Calibración de distorsión . . . . .	39
48.	Marcador Aruco . . . . .	39
49.	Calidad de imagen . . . . .	40
50.	Código de ejes de posición . . . . .	40
51.	Código de sistema de reconocimiento . . . . .	41
52.	Código de detección de movimiento . . . . .	41
53.	Trayectoria 1 . . . . .	42
54.	Trayectoria 2 . . . . .	42
55.	Inicio de la prueba de vuelo autónomo en zona interior . . . . .	43
56.	Vuelo autónomo en zona interior . . . . .	44
57.	Prueba de ejecución de vuelo en zona interior . . . . .	44
58.	Marcador Aruco . . . . .	45
59.	Reconocimiento de área . . . . .	47
60.	Reconocimiento de zona de vuelo . . . . .	48
61.	Inicio de la prueba en zona exterior . . . . .	48
62.	Ejemplo de los parámetros para el inicio de la prueba . . . . .	49
63.	Ejecución de la prueba . . . . .	49
64.	Drone ejecutando plan de vuelo . . . . .	50
65.	Marcador Aruco . . . . .	50
66.	Sistema de reconocimiento . . . . .	52
67.	Persona reconocida mediante cámara . . . . .	53
68.	Reconocimiento de objetos . . . . .	53

69.	Sistema de reconocimiento en ejecución . . . . .	54
70.	Reconocimiento de personas . . . . .	55
71.	Sistema de reconocimiento . . . . .	55
72.	Sensor PIR . . . . .	57
73.	Notificación de movimiento . . . . .	57
74.	Inicio de vuelo . . . . .	58
75.	Sensor PIR en zona exterior . . . . .	59
76.	Notificación de movimiento . . . . .	59
77.	Ejecución del vuelo . . . . .	60
78.	Código para el tiempo de respuesta del sistema de comunicación . . . . .	61
79.	Visualización del tiempo de respuesta . . . . .	61
80.	Código implementado . . . . .	63
81.	Ejecución del código . . . . .	63
82.	Código implementado . . . . .	64
83.	Software de monitoreo . . . . .	64
84.	Diagrama de flujo fase autónoma . . . . .	73
85.	Diagrama de flujo fase de reconocimiento . . . . .	74
86.	Diagrama de flujo fase de reconocimiento . . . . .	75

## Resumen

**E**l presente proyecto tiene como objetivo el diseñar e implementar un sistema de reconocimiento autónomo para el sector rural mediante un drone, el cual será diseñado por fases que componen las siguientes: fase de diseño autónomo, fase de sistema de reconocimiento y fase de comunicación.

En el marco teórico se presenta las características principales de un drone, esto permite identificar según nuestra necesidad cual es el drone idóneo que se puede utilizar, es importante tener en cuenta que el dispositivo debe ser programable, ya que permite a la persona generar un control de acuerdo a una trayectoria. Además, se presenta el software utilizado, en el cual se desarrolló las líneas de código que permitirán los diferentes movimientos, cada una de las fases tiene su propio código para su función específica, implementándolo en la fase final al dispositivo para las respectivas pruebas.

En los análisis de resultados se presenta cada una de las fases que se desarrolló, de una manera concisa y detallada especificando los códigos con una breve explicación, permitiendo al lector un mejor entendimiento del proceso. Finalmente, en conclusiones y recomendaciones se realiza una autoevaluación del proceso que se llevó a cabo, y se especifican algunas recomendaciones muy importantes para proyectos a futuro con respecto a temas similares del proyecto, esto con el objetivo de tener muy en cuenta que tipo de drone es el correcto para ser utilizado ante diferentes opciones que se presentan en el mercado.

**Palabras clave:** Autónomo, Drone, Reconocimiento, Implementación, Software

## Abstract

The objective of this project is to design and implement an autonomous recognition system for the rural sector using a drone, which will be designed in phases that make up the following: autonomous design phase, recognition system phase, and communication phase.

In the theoretical framework, the main characteristics of a drone are presented, this allows us to identify according to our needs which is the ideal drone that can be used, it is important to take into account that the device must be programmable, since it allows the person to generate a control according to a trajectory. In addition, the software used is presented, in which the lines of code that will allow the different movements were developed, each of the phases has its own code for its specific function, implementing it in the final phase to the device for the respective tests.

In the analysis of results, each one of the phases that was developed is presented, in a concise and detailed way, specifying the codes with a brief explanation, allowing the reader a better understanding of the process. Finally, in conclusions and recommendations, a self-evaluation of the process that was carried out is carried out, and some very important recommendations are specified for future projects with respect to similar issues of the project, this with the objective of taking into account what type of drone is the correct one to be used before different options that are presented in the market.

**Keywords:** Autonomous, Drone, Reconnaissance, Implementation, Software.



# 1. Introducción

En la provincia del Azuay el sector rural, actualmente se encuentra sin avances y desarrollos tecnológicos de gran alcance, el tema de la seguridad es una parte muy importante a ser considerada en la actualidad, es por ello que el “Diseño e implementación de un sistema de reconocimiento autónomo para el sector rural mediante un drone”, se considera como una solución innovadora que generara un gran beneficio ante diversos problemas de inseguridad. Por intermedio de un control autónomo de un drone se puede mejorar el monitoreo en la seguridad y la previsión de atentados cuando las residencias se encuentran deshabitadas. Cuando sucede una emergencia la respuesta por lo general no es inmediata, ya sean por varios factores como: la ubicación o acceso a la vivienda, esto dificulta la llegada de ayuda inmediata, además ante un atentado de robo al domicilio, la persona que se encuentra dentro de la residencia se haya en peligro, es por ello que el uso de un drone dotado con la capacidad tecnológica de reconocimiento puede considerarse como una solución eficaz.

El sistema de reconocimiento autónomo, posibilita al drone ejecutar vuelos o trayectorias programadas en puntos específicos, recolectando información ante situaciones sospechosas o de probables amenazas, brindando una percepción detallada del área. Mediante la inteligencia artificial, esta nos permite analizar el contenido captado por la aeronave, accediendo a una detección anticipada de una actividad ilícita como el robo o daño hacia la vivienda u otras acciones ilegales.

Los datos receptados son en tiempo real, brindándole al usuario confianza del funcionamiento del sistema, el cual desempeña una función importante, ofreciendo un monitoreo efectivo de la residencia en el sector rural.

## **2. Problema**

### **2.1. Antecedentes**

En el Ecuador en los últimos años existe un incremento de inseguridad, según la fiscalía General del estado, sus estadísticas presentadas desde Enero a Noviembre del 2020 existieron 6.643 robos a domicilio, en comparación de Enero a Noviembre del 2021 existieron 7.449 robos a domicilios, produciendo una variación en el incremento del 12.1%, lo cual nos deja ver la falta de seguridad que está atravesando el país. (Estado, 2021)

En la actualidad existe una gran variedad de drones para vigilancia, estos equipos son ideales para labores de seguridad, ya que se puede instalar cámaras térmicas, detectores de calor, vuelan cerca de ocho horas y cubren hasta 100 kilómetros de distancia. En el ámbito de vigilancia tienen un papel muy importante en el Ecuador, especialmente en protección y control de fronteras, monitoreo de propiedades privadas o en rescates terrestres y marítimos, además el servicio Nacional de Aduana del Ecuador SENA, controla los pasos fronterizos con Perú utilizando drones, estos permiten obtener información en tiempo real lo cual es una gran ayuda para sistemas de vigilancia o monitoreo.

Por otro lado, estos dispositivos tienen ciertas limitaciones, ya que estos equipos de vigilancia según sus características pueden llegar a costar entre 60 y 120 mil dólares, teniendo un uso aproximado de ocho horas si se los compara con un dron comúnmente usado para grabar videos publicitarios, estos utilizan baterías que permiten un uso de alrededor de 20 a 25 minutos y tienen una cobertura hasta de tres kilómetros de distancia. (Espol, 2017)

### **2.2. Descripción del problema**

Con el inicio de la pandemia, los estilos de vida de muchas personas cambiaron, provocando que abandonen su zona residencial, incrementando las construcciones de casas de campo en sectores rurales, para mantenerse alejados del estrés diario y la tensión que provocó el estar aislados, paulatinamente las personas han regresado a la normalidad, quedando sus propiedades y viviendas deshabitadas nuevamente. Por lo general las casas campestres no cuentan con ningún sistema de seguridad convirtiéndose en zonas vulnerables al vandalismo. Las cifras presentadas por el INEC nos reflejan el incremento de robos a domicilios que han existido, por ejemplo: Desde Enero a Mayo del 2021 existieron aproximadamente 3.198,

mientras que de Enero a Mayo del 2022, existió 3.413 produciendo una variación del 16.8% en el aumento de atracos a domicilios. (INEC, 2022)

Según la revista vistazo, Ecuador fue el tercer país de Latinoamérica con mayor índice de robos, a viviendas llevados a inicios del año 2022, el estudio que realizó la Consultoría Interdisciplinaria en Desarrollo Gallup (CIP), esto se debe a que las viviendas no cuentan con un sistema de seguridad, es por esto que recomienda que se instalen cámaras de seguridad, alarmas o cualquier otro tipo de sistema de seguridad. (Revista, 2022)

### **2.3. Importancia y alcances**

El estudio realizado contribuirá en el diseño e implementación de un sistema autónomo en un dron, el cual brindará seguridad a las residencias campestres, con el propósito de que la persona se sienta más segura cuando no se encuentre en su vivienda, los drones están en constante evolución debido a la dificultad de operarlos, pero esto ha ido cambiando con el pasar del tiempo, ya que cada vez existen nuevas mejoras, lo que permite que el control del dron sea de manera precisa y sencilla. En la actualidad, con estas mejoras se ha convertido en una herramienta de gran importancia en varios ámbitos como lo son: permitir recorrer áreas extensas en muy poco tiempo, llegar a lugares de difícil acceso y reducción en riesgos personales, etc. (Suh & Nam, 2020)

Al implementar un sistema autónomo en un dron, el usuario no tendrá la necesidad de controlar el dispositivo, obteniendo como resultado reconocer el área en un tiempo real, captando contenido multimedia, de tal manera el usuario podrá visualizar el perímetro de la residencia, dando seguridad al propietario.

### **2.4. Delimitación**

El problema de estudio se delimitará en las siguientes dimensiones:

#### **2.4.1. Espacial o geográfica**

El sistema será desarrollado para la zona rural de la provincia del Azuay, con énfasis en: los domicilios, quintas, casas de campo, debido a la inseguridad que está atravesando el país.

### **2.4.2. Temporal**

El proyecto está planificado para realizarse en la cátedra correspondiente a Trabajo de Titulación 1 que cuenta con 160 horas, seguido de Trabajo de Titulación 2 que cuenta con 240 horas, dando un total de 400 horas, el cual se llevara a cabo pruebas, simulaciones y ajustes para seleccionar los mejores parámetros para el desarrollo del proyecto.

### **2.5. Problema General**

- ¿Cuál es el diseño idóneo que permita brindar seguridad a las residencias del sector rural mediante un sistema autónomo en un drone?

### **2.6. Problemas Específicos**

- ¿Es posible desarrollar un sistema de control autónomo para el drone?
- ¿Se podrá establecer la comunicación del sistema autónomo en un drone con un sistema de seguridad?
- ¿Es posible adaptar en el sistema autónomo en un drone un software de monitoreo que permita visualizar la trayectoria del mismo?

## **3. Objetivos**

### **3.1. Objetivo General**

- Diseño e implementación de un sistema de reconocimiento autónomo para el sector rural mediante un drone.

### **3.2. Objetivos Específicos**

- Desarrollar el sistema de control autónomo mediante un drone.
- Establecer la comunicación entre un sistema autónomo mediante un drone con el sistema de seguridad.
- Adaptar un software de monitoreo que permita visualizar la trayectoria autónoma del drone.

## 4. Marco Teórico

### 4.1. Drone

Los drones son aquellos dispositivos dirigidos a distancia (UAV, vehículo aéreo no tripulado), es decir, dispositivos que son controlados por el usuario mediante un radio mando o control remoto a distancia. La cantidad de drones han aumentado alrededor del mundo para múltiples usos como, por ejemplo: grandes empresas los utilizan para cumplir funciones de paquetería, envíos, ingreso a zonas de difícil acceso e incluso para vigilancia. (Çetin, 2021)

La manipulación de los drones es un factor muy importante, según el estudio realizado por (Suh & Nam, 2020), nos mencionan de dos tipos de pilotaje: el externo y el interno, al manipular de forma interna la persona verifica la condición del dispositivo, mientras que al manipular de forma externa la persona controla al drone desde el punto de vista del usuario, lo cual presenta un gran problema, por consiguiente, genera algunos errores debido al punto de vista entre la persona y el drone, por esta razón, en el estudio se realizaron pruebas que permitan diseñar un sistema autónomo del dispositivo de forma centrada en la persona para que exista una manipulación más sencilla.

Existe información relevante acerca de los sistemas de reconocimiento por intermedio de drones autónomos, permitiendo implementarse en el tema de seguridad domiciliaria, según varios autores (Konstantoudakis, 2022), el control de drones mediante la realidad aumentada nos permite que la manipulación de un drone ya no sea complicada; al contrario, cualquier persona puede manipularlo mediante el reconocimiento de gestos como el movimiento de las manos.

### 4.2. Tipos de drones por su definición

La definición general de un drone es; una aeronave que no tiene un piloto humano a bordo, pero existen diferentes conceptos para podernos dirigir a estas aeronaves:

- **Drone**, esta palabra es de origen inglés, su significado es zángano, este término fue utilizado para dar referencia a los aviones militares no tripulados.
- **RPA o RPAS**, son las siglas en el idioma inglés de Remotely Piloted Aircraft y Remotely Piloted Aircraft System (Aeronaves Pilotadas a Distancia y Sistema de Aeronaves

Pilotadas) respectivamente, haciendo referencia al control remoto de la aeronave. (Gomis, 2018)

- **UAV o UAS**, que tiene como significado Unmanned Aerial Vehicle y Unmanned Aerial System (Vehículo aéreo no tripulado y Sistema aéreo no tripulado) respectivamente. Este término hace referencia la falta de un piloto de forma física a bordo de la aeronave. (Iberfdrone., 2021)

### 4.3. Tipología de drones por número de brazos

- Tricóptero.** Se componen de tres brazos y tres motores: Los delanteros trabajan en sentido contrario para resistir entre sí, mientras el posterior se desarrolla por medio de un servomotor y ayuda con la firmeza en el vuelo, como se muestra en la figura 1. (Iberfdrone., 2021)

**Figura 1**

*Tricóptero.*



**Nota:** Se presenta el modelo del dron tipo tricóptero de acuerdo con Ledesma (2019).

- Cuadricóptero.** Se constituye de cuatro brazos y cuatro motores, los cuales están situados en cada brazo. Son los más tradicionales en el ámbito comercial, como se observa en la figura 2. (Iberfdrone., 2021)

## Figura 2

*Cuadricóptero.*



**Nota:** Modelo del dron tipo cuadricóptero de acuerdo con Ledesma (2019).

- c Hexacóptero.** Son los drones más frecuentes en el entorno profesional, ya que proporciona una estabilidad muy aceptable para las tomas aéreas, en la figura 3, se observa que están constituidos por seis brazos, los cuales posibilitan aterrizar con seguridad en caso de que algún motor falle. (Iberfdrone., 2021)

## Figura 3

*Hexacóptero.*



**Nota:** Se muestra el modelo del dron tipo hexacóptero según Ledesma (2019).

**d Octocóptero.** Los ocho brazos generan mayor potencia y estabilidad en el aire, su desventaja es su tamaño, lo cual genera un manejo con mayor experiencia. La figura 4 nos muestra un ejemplo del drone tipo octocóptero. (Iberfdrone., 2021)

**Figura 4**

*Octocóptero.*



**Nota:** Drone tipo octocóptero según Ledesma (2019).

**e Coaxiales.** Su ventaja principal es el número de motores por cada brazo, por lo general dos, convirtiéndose en la mejor opción para desarrollar trabajos que exijan parámetros profesionales, se puede observar un ejemplo en la figura 5 (Iberfdrone., 2021).



## Figura 5

*Coaxial.*



**Nota:** Drone coaxial según Ledesma (2019).

### Drones según su método de control

El método de control hace referencia a las distintas técnicas para poder estabilizar la aeronave que se encuentra en vuelo y el seguimiento de una trayectoria en el caso para drones autónomos (Richard, 2018). Generalmente se tiene tres métodos de control:

- **Drones controlados remotamente:** El controlador de la aeronave recibe del piloto la posición angular y la fuerza que los rotores deben generar, la manera en que se estabiliza se limita de las habilidades del piloto (Richard, 2018).
- **Drones supervisados:** Es un punto medio entre drones controlados remotamente y drones autónomos, consiste en estabilizar a la aeronave a una altura definida por el usuario y una posición angular horizontal, puede realizar ejercicios de forma autónoma, pero con la supervisión de una persona (Richard, 2018).
- **Drones autónomos:** El dron precarga una trayectoria de vuelo programada por la persona, y posteriormente éste le sigue sin la intervención del piloto (Richard, 2018).

## 4.4. Partes de un drone

### 4.4.1. Marco (frames)

Se denomina la carcasa del drone, está conformada por aleaciones metálicas, dependiendo del modelo puede ser de otros materiales, su objetivo es generar la disminución de peso, además sobre la estructura se fijan elementos y sensores. Como característica importante es generar una gran resistencia a parámetros naturales como el viento o lluvia, como se observa en la figura 6 (Ledesma, 2019).

#### Figura 6

*Marco.*



**Nota:** Carcasa del dron según Ledesma (2019).

### 4.4.2. Batería

Proporciona energía suficiente para el respectivo funcionamiento del dispositivo, las más usadas son las de litio, ya que sus características cumplen los requisitos del drone para ejercer correctamente su funcionamiento. Sus celdas son de 3.7 V, no debemos confundir mientras más grande es la batería mayor rendimiento, ya que se debe considerar otros parámetros como la eficiencia de los motores, la figura 7 nos muestra una batería típica del drone (Ledesma, 2019).

## Figura 7

*Batería 6 celdas.*



**Nota:** Batería de litio Ledesma (2019).

### 4.4.3. Motores y hélices

Las figuras 8 y 9 nos muestra las hélices que permitirán elevar al drone por intermedio de la fuerza opuesta a la gravedad generada por los motores (Ledesma, 2019).

## Figura 8

*Motor.*



**Nota:** Motor Ledesma (2019).

## Figura 9

*Hélices.*



**Nota:** Hélices de acuerdo con Ledesma (2019).

### 4.4.4. Radio receptor (mando)

Recibe la señal que es enviada por medio de control remoto, el usuario ejecuta los movimientos o direcciones que desea, estos datos se envían al controlador para que lleve a cabo la instrucción como se muestra en la figura 10 (Ledesma, 2019).

## Figura 10

*Control genérico..*



**Nota:** Radio receptor Ledesma (2019).

#### 4.4.5. Sistema de posicionamiento global “GPS”

Permite determinar la posición del dispositivo, el margen de error es casi nulo, ya que el objetivo principal del GPS es la precisión, esto determina con exactitud la ubicación del dron. La mayoría de los drones incorporan un botón de GPS, el cual es accionado cuando el usuario ha perdido el alcance de controlarlo o si su batería está a punto de agotarse, permitiendo que el dron regrese de manera autónoma a su punto de origen, en la figura 11 se observa el sistema GPS (Ledesma, 2019).

#### Figura 11

*Sensor GPS.*



**Nota:** GPS del dron Ledesma (2019).

#### 4.4.6. Visión en primera persona “visores de realidad virtual o FPV”

Es un procedimiento de recepción y emisión de video, procesado por las cámaras del dispositivo en tiempo real. Lo cual permite a la persona visualizar todo lo que el dispositivo está captando en pleno movimiento, como se muestra en la figura 12 (Ledesma, 2019).

## Figura 12

*Visor de realidad virtual.*



**Nota:** Realidad virtual Ledesma (2019).

### 4.4.7. Controlador de vuelo/ placa controladora

Es una placa, en la cual se encuentra circuitos integrados en donde se conectan la mayoría de los componentes y sensores del drone, los cuales detectan cambios de dirección generadas por el usuario, el controlador debe ser apto para ejecutar misiones y lograr un aterrizaje adecuado. En la figura 13, se observa un controlador de vuelo (Ledesma, 2019).

## Figura 13

*Módulo ardupilot.*



**Nota:** Controlador de vuelo Ledesma (2019).

#### 4.4.8. Estabilizador Gimbal

Se lo considera como un estabilizador externo para la cámara, es decir; minimiza las perturbaciones generadas en el vuelo, como se observa en la figura 14 (Ledesma, 2019).

**Figura 14**

*Estabilizador Gimbal.*



**Nota:** Estabilizador de imagen Ledesma (2019).

#### 4.4.9. Sensores a bordo de los drones – Cámara

Las cámaras son las encargadas de capturar y transmitir el entorno donde se encuentra el drone, hay que considerar parámetros como: el peso, autonomía del drone, estabilización, etc., debe ser lo suficientemente capaz de capturar imágenes en movimiento que permitan una visualización agradable al usuario. De acuerdo con nuestro uso, existe una gran variedad de cámaras como térmicas o infrarrojas, como se muestra en la figura 15 (Ledesma, 2019).

## Figura 15

*Cámara.*



*Nota:* Cámara Ledesma (2019).

### 4.5. Drones programables

A continuación, en la tabla 1, se presenta algunos de los modelos de drones programables más utilizados con sus respectivas características.



**Tabla 1***Características de drones de programación*

<b>Modelo</b>	<b>Característica</b>
Ryze Tech	Cámara: 5MP Video HD 720 Tiempo de vuelo: 13 min Precio: 100 dólares americanos Sistema de posicionamiento de visión Protección a pruebas de fallas Alcance: 100 metros
Robolink CoDrone Mini	Cámara: No incorporada Tiempo de vuelo: 5 min Precio: 90 dólares americanos Una velocidad de vuelo Modos de vuelo inteligentes limitados Alcance: 160 ft
Parrot Mambo Fly	Cámara: HD 720p Tiempo de vuelo: 20 min Precio: 209 dólares americanos Sensores de estabilización Conectividad Bluetooth Alcance: 100 metros
Sky Viper Fury	Cámara: No incorporada Tiempo de vuelo: 8 min Precio: 113 dólares americanos Conectividad mediante control remoto Tecnología de escaneo de superficie para un vuelo estable Alcance: 30 metros
DJI Tello Quadcopter Drone	Cámara: HD 1280 x 720p Tiempo de vuelo: 15 min Precio: 119 dólares americanos Procesador Intel Tecnología de comunicación mediante Wi-Fi Alcance: 100 metros

**Nota:** En la tabla 1 se describe las características principales de los diferentes drones programables según ProjectGo (2022).

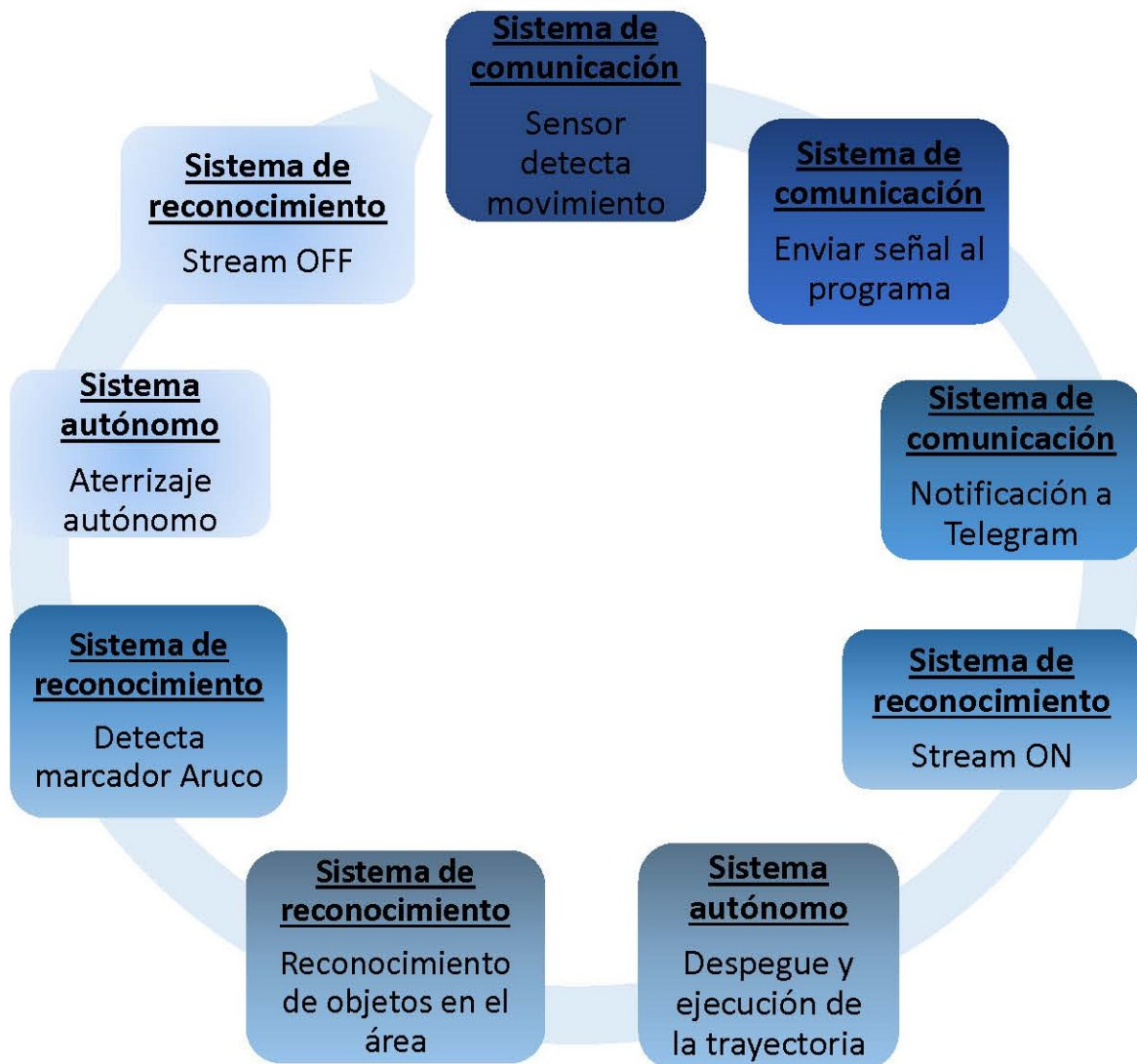
## **5. Marco metodológico**

### **5.1. Diagrama de funcionamiento del sistema de reconocimiento autónomo mediante un dron**

En la figura 16 se muestra el diagrama de funcionamiento del sistema de reconocimiento autónomo mediante un dron. La funcionalidad del sistema inicia cuando el sensor detecta un movimiento, el mismo que produce una señal al módulo ESP32, éste es el encargado de enviar una notificación a la aplicación Telegram del usuario y de transmitir la señal al centro de mando, el cual va activar el sistema de reconocimiento y la ejecución de la trayectoria del dron, una vez realizada dicho plan de vuelo preestablecido, procede el dron a realizar el aterrizaje autónomo, esto se da por la identificación del marcador Aruco mediante la cámara de la aeronave, por último una vez que aterrice, el stream de la cámara se apaga.

**Figura 16**

*Diagrama de funcionamiento.*



**Nota:** Se observa el diagrama de funcionamiento del sistema de reconocimiento autónomo.

## **5.2. Fase 1 Diseñar el sistema de control autónomo.**

Luego de revisar las características de los diferentes drones programables, la mejor opción es el modelo “DJI Tello Quadcopter Drone”, ya que sus prestaciones son muy tolerables para

la ejecución de las diferentes fases. A continuación, se presenta las diferentes prestaciones de la aeronave.

En la figura 17, se presenta al dron DJI Tello, el cual, es un vehículo aéreo no tripulado de tipo cuadricóptero, el cual se compone de un sistema de posicionamiento visual y una cámara integrada. El sistema de controlador de vuelo y de posicionamiento visual le permite volar en modo estacionario siendo muy apto para controlarlo en interiores. Sus funciones sofisticadas como: volteretas 8D, modo rebote, modo tirar y listo; permiten que el manejo del dispositivo sea muy entretenido. (Tech, 2018).

### **Figura 17**

*Dron DJI Tello.*



**Nota:** Vehículo aéreo no tripulado de acuerdo con Tech (2023)

La cámara integrada permite capturar fotos con una calidad de 5 megapíxeles y emitir videos en vivo en calidad HD 1280 x720p al móvil por medio de la aplicación Tello, además su tiempo límite de vuelo es de 15 minutos, con una distancia máxima de vuelo de 100 m. En caso de pérdida de conexión, la protección a prueba de fallas permite al dron aterrizar de forma segura. (Tech, 2018)

### 5.2.1. Python

El software Python permite desarrollar un lenguaje de programación que se puede aplicar en el crecimiento de software, desarrollo web y aprendizaje automático. Se lo utiliza por la razón de su eficiencia y simplicidad de aprendizaje y el costo de mantenibilidad y creación de programas facilita que personas trabajen en colaboraciones sin la necesidad de gran experiencia. (O'Shea, 2006)

### 5.2.2. Librería de Python

En el proyecto se utilizaron las siguientes librerías, las cuales nos permiten comunicarnos a través del ordenador con el drone.

- **Numpy**

Es una biblioteca utilizada por los programadores para la creación y administración de matrices, además realizar cálculos de algebra lineal. (O'Shea, 2006)

- **OpenCV-Python**

Se la utiliza para el procesamiento de imágenes con la finalidad de la adaptación de inteligencia artificial, además genera un ambiente en 2D y 3D. (O'Shea, 2006)

### 5.2.3. Calibración de la cámara del drone

Se inicia con la calibración de la cámara, esto nos permite corregir dos tipos de distorsiones como lo son la radial y la distorsión tangencial.

La distorsión radial produce que las líneas rectas se muestren en forma de curva, al alejarnos el efecto es más notorio.

Esto se resuelve con la siguiente expresión:

$$X_{corrected} = x(1 + K_1r^2 + K_2r^4 + K_3r^6) \quad (1)$$

$$Y_{corrected} = y(1 + K_1r^2 + K_2r^4 + K_3r^6) \quad (2)$$

La distorsión tangencial se produce cuando la toma de imágenes no se encuentra alineado paralelo al plano de la imagen, es decir algunas zonas de la imagen se aprecia muy cercanas.

Esto se resuelve con la siguiente expresión:

$$X_{corrected} = x + [2p_1xy + p_2(r^2 + 2x^2)] \quad (3)$$

$$Y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2xy] \quad (4)$$

Los parámetros intrínsecos y extrínsecos o también conocido como matriz de cámara son principales en una cámara ya que determina datos como la distancia focal “ $f_x, f_y$ ” y centros ópticos “ $C_x, C_y$ ”. Al obtener estos datos mediante la respectiva calibración estos pueden servir para proyectos a futuro, a continuación, se muestra en ecuación 5 la matriz de 3x3 con los respectivos parámetros. (Unipython, 2018)

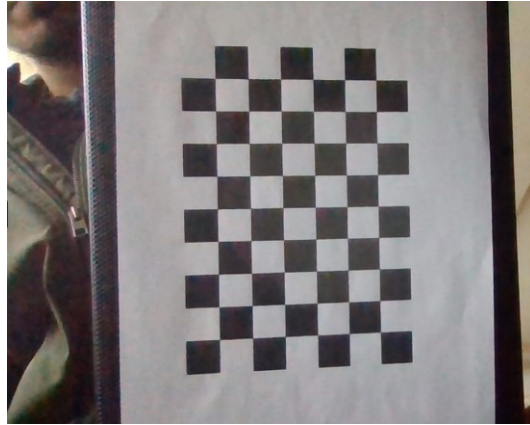
$$cameramatrix = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Donde se puede observar el orden de los parámetros de la matriz de 3x3 de la cámara.

Para hallar la respectiva configuración se utiliza el software Python y la función **cv2.findChessboardCorners()**, empleamos una cuadrícula de 9x6 como se muestra en la figura 18.

## Figura 18

*Cuadrícula de 9x6.*

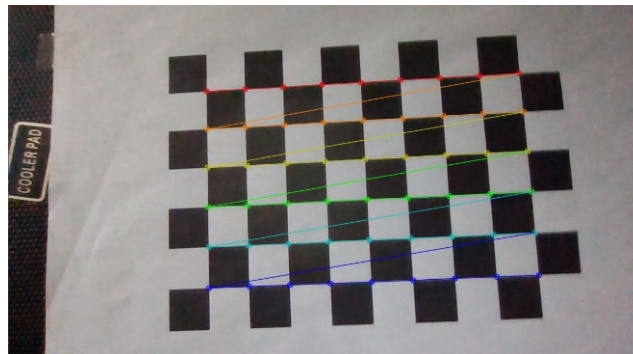


**Nota:** Se utiliza una cuadrícula de dimensiones de 9x6 para la calibración de cámara.

A continuación se muestra en la siguiente figura 19, los patrones generados utilizando la función `cv2.drawChessboardCorners()`.

## Figura 19

*Cuadrícula con patrones generados.*



**Nota:** Se observa los patrones dibujados mediante el respectivo comando.

Ahora utilizamos la función `cv2.calibrateCamera()`, la cual nos genera los respectivos coeficientes de distorsión, rotación y traslación, como se muestra en la figura 20.

## Figura 20

*Resultados de la matriz de calibración.*

```
[[943.83414414  0.      616.28778648]
 [ 0.          944.25798881 352.8975664 ]
 [ 0.           0.         1.         ]]
```

**Nota:** Los resultados obtenidos nos permite calibrar la cámara para tener una resolución óptima. .

### 5.2.4. Aruco

Se denomina a Aruco como aquellos marcadores de figura cuadrada con características de fondo negro y la matriz binaria ubicada de forma interna, constituye una biblioteca con códigos abiertos acorde a OpenCV, la cual admite identificar los distintos marcadores, de igual manera si la cámara que se va a utilizar se encuentra debidamente calibrada nos permite identificar la localización del marcador en relación a la cámara. Todos los marcadores se agrupan en diferentes “diccionarios”, el cual contiene una cantidad determinada y un mismo tamaño, es decir número de bytes. En la siguiente figura 21 se muestra un marcador estándar. (Jiménez Bravo, 2018)

## Figura 21

*Marcador estándar.*



**Nota:** Se presenta un modelo aruco tipo estándar según Jiménez Bravo (2018)

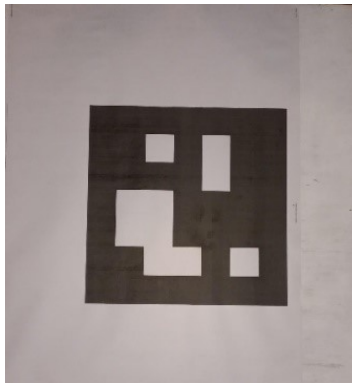


### 5.2.5. Marcadores

Presenta un color negro en su borde exterior y en su parte interna un patrón binario único que permite codificar de acuerdo a la necesidad del usuario. En relación al diccionario que pertenezca el marcador a utilizar, este presentara diferentes cantidades de bits pueden ser más o menos, mientras mayor sea el número de bits del marcador se minimiza el factor de confusión y determina una identificación correcta en el momento de la detección del marcador, se presenta en la figura 22 un marcador impreso. (Jiménez Bravo, 2018)

#### Figura 22

*Marcador aruco impreso.*



**Nota:** Se observa el marcador aruco impreso a utilizar.

### 5.2.6. Código generado para marcador aruco

En la figura 23, se observa las diferentes librerías importadas en el software Python, las cuales permiten ejecutar las diferentes funciones para el desarrollo del código.

### Figura 23

*Librerías.*

```
from djitellopy import Tello
import cv2 as cv
import numpy as np
```

**Nota:** Se observa las respectivas librerías que permiten ejecutar al código.

Se declara las variables respectivas para que puedan ser ejecutadas, en la figura 24 se muestra la línea 16, la cual permite habilitar el stream del drone.

### Figura 24

*Variables Tello.*

```
13 tello=Tello()
14 tello.connect()
15 print(tello.get_battery())
16 tello.streamon()
```

**Nota:** Código para la habilitación de la cámara para detectar el marcador aruco.

Se genera la declaración de la variable **cv.aruco**, como se muestra en la figura 25, la cual permite detectar el marcador cuando se activa la cámara del drone.

## Figura 25

*Variable Aruco.*

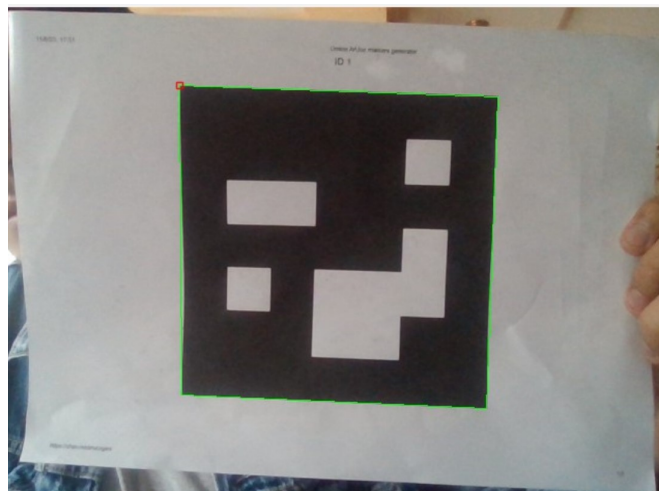
```
119 # -- Draw the detected marker and put a reference frame over it
120 cv.aruco.drawDetectedMarkers(frame, corners)
121 cv.drawFrameAxes(frame, camera_matrix, camera_distortion, rvec, tvec, 10)
122
123 # -- Print the tag position in camera frame
124 str_position = "MARKER Position x=%4.0f y=%4.0f z=%4.0f" % (tvec[0], tvec[1], tvec[2])
125 cv.putText(frame, str_position, (0, 100), font, 1, (0, 255, 0), 2, cv.LINE_AA)
```

**Nota:** Como se muestra en la figura el código para generar el marcador aruco.

Al ejecutar el código generado se puede observar en la figura 26, la creación de un contorno color verde el cual indica la detección del marcador.

## Figura 26

*Detección bordes de marcador Aruco.*



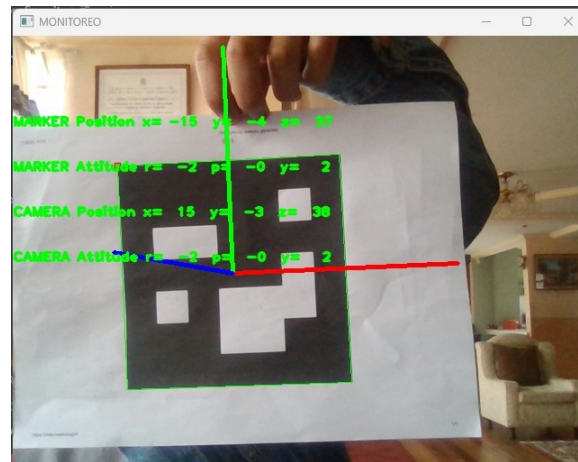
**Nota:** La cámara identifica el borde y el marcador Aruco.

En la figura 27 se observa las diferentes coordenadas de posición detectadas por la cámara, estas pueden ser diferentes dependiendo de la distancia del marcador Aruco, mientras que en la figura 28 se observa la detección de los ejes de referencia por medio de la cámara del dron, el sistema infrarrojo de posicionamiento reconocerá el marcador, permitiendo que el aterrizaje

sea autónomo.

### Figura 27

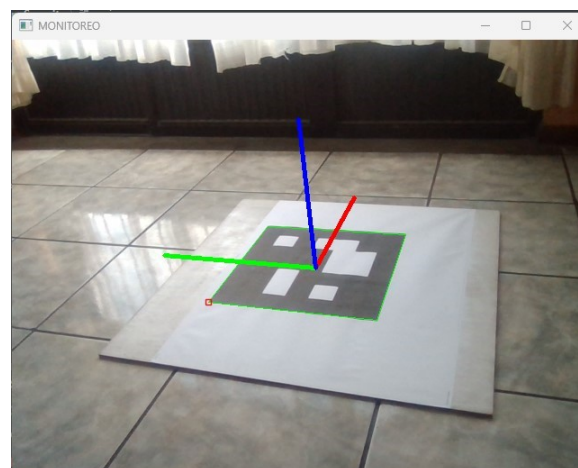
*Coordenadas de posición del Aruco con respecto a la cámara del dron.*



**Nota:** En la figura se observa los ejes generados con sus respectivas coordenadas de posición detectadas por el dron

### Figura 28

*Proyección de los ejes de referencia.*



**Nota:** En la figura se muestra la detección de los ejes mediante la cámara del dron.

### 5.3. Fase 2 Elaboración del sistema de reconocimiento.

Se explica el desarrollo del sistema de reconocimiento, en el cual el dron reconocerá las diferentes presencias como personas u objetos en la zona de vuelo.

#### 5.3.1. Código

El código se lo desarrollo en el software Python, como se observa en la figura 29 importamos las respectivas librerías a utilizar en este caso “**import cv2, import urlopen, import numpy**”

#### Figura 29

*Importación de librerías.*

```
1 import cv2
2 from urllib.request import urlopen
3 import numpy as np
```

**Nota:** Se observa en la figura las librerías a utilizar.

En la figura 30 se observa las siguientes líneas de código, en este caso en la línea 6 insertamos el “id” de la cámara, con la cual se inicia el stream del video. El cual al iniciar el código se observará el contenido captado por la cámara.

#### Figura 30

*ID de cámara.*

```
6 stream = urlopen('http://192.168.1.16:81/stream')
7 bytes = bytes()
```

**Nota:** El ID de la cámara permite inicializar el stream del video.

A continuación, se importa las diferentes clases de la base de datos denominada “COCO”, esto quiere decir “objetos comunes en contexto”, nos permite la detección de los diferentes objetos con sus respectivas etiquetas, cuando se inicia el stream del video, en la figura ?? se muestra las líneas de código empleadas.

### Figura 31

*Línea de código de objetos comunes en contexto.*

```
9  classNames = []
10 classFile = 'coco.names'
11 with open(classFile, 'rt') as f:
12     classNames = f.read().rstrip('\n').split('\n')
13
14 configPath = 'ssd_mobilenet_v3_large_coco_2020_01_14.ptxt'
15 weightsPath = 'frozen_inference_graph.pb'
```

**Nota:** Se muestra las respectivas líneas de código para la detección de objetos.

En la figura 32 se muestra algunas de las clases importadas de la base de datos “COCO” para el sistema de reconocimiento, en total son 91, es decir; reconocerá dicha cantidad de objetos en la zona de vuelo.

### Figura 32

*Clases de datos.*

```
1  persona
2  bicycle
3  car
4  motorcycle
5  airplane
6  bus
7  train
8  truck
9  boat
10 traffic light
11 fire hydrant
12 street sign
13 stop sign
14 parking meter
15 bench
16 bird
17 cat
18 dog
19 horse
20 sheep
21 cow
22 elephant
23 bear
24 zebra
25 giraffe
26 hat
```

**Nota:** Se muestra ciertas clases de datos a utilizar para la detección de objetos.

Se elabora las siguientes líneas de código, las cuales tiene la función de mostrar el contorno del recuadro y las etiquetas de las clases de cada uno de los objetos identificados, como se muestra en la figura 33.

### Figura 33

*Línea de código del contorno de la imagen.*

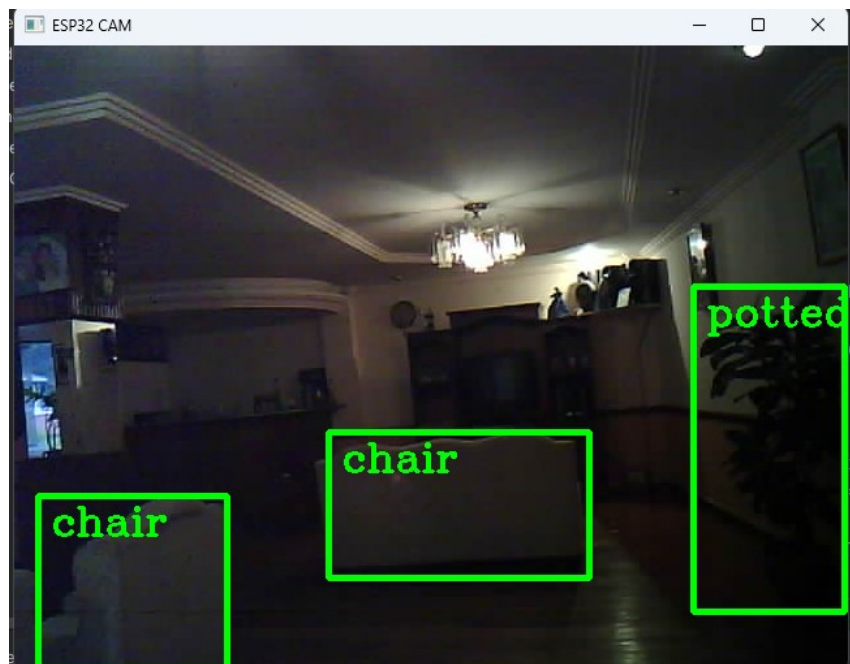
```
for classId, confidence, box in zip(classIds.flatten(), confs.flatten(), bbox):
    cv2.rectangle(img, box, color=(0, 255, 0),
                  thickness=3) # mostramos en rectangulo lo que se encuentra
    cv2.putText(img, classNames[classId - 1], (box[0] + 10, box[1] + 30), cv2.FONT_HERSHEY_COMPLEX, 1,
                (0, 255, 0), 2)
```

**Nota:** Las líneas de código generan el recuadro de la respectiva clase identificada.

En la figura 34 se puede observar el código ejecutado, mostrando como resultado los diferentes objetos detectados por la cámara del dron.

### Figura 34

*Objetos detectados.*



**Nota:** Se puede observar los objetos detectados por el sistema de reconocimiento del dron.



## 5.4. Fase 3 Elaboración del sistema de comunicación.

Se explica el desarrollo del sistema de comunicación, por medio del sensor de movimiento PIR.

### 5.4.1. Sensor PIR

El sensor PIR hc-sr501 como se muestra en la figura 35, se lo utilizó por su accesibilidad y costo en el mercado, además por sus características de captar la radiación electromagnética infrarroja que emite los seres vivos mediante su temperatura corporal, por lo tanto es idóneo para este tipo de pruebas, por consiguiente, envía una notificación de alerta al usuario del movimiento detectado.

#### Figura 35

*Sensor PIR.*



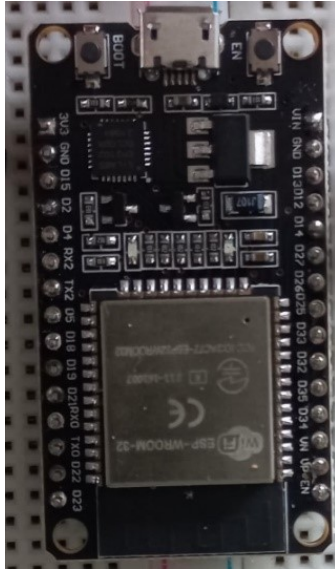
**Nota:** Se observa el sensor de movimiento PIR que activara el sistema de comunicación.

### 5.4.2. Módulo ESP32

El módulo ESP32 como se observa en la figura 36, implementa un sistema de Wifi/Bluetooth que permite el monitoreo y comunicación por medio de una conexión de red internet, de esta manera establece la transmisión de datos.

## Figura 36

*Módulo ESP32.*



**Nota:** El módulo ESP 32 permite la transferencia de datos recibidos, al sistema de comunicación.

### 5.4.3. Código

En la figura 37, se observa las líneas de código empleadas en las cuales se inicia con el registro de red Wi-Fi y su respectiva contraseña, permitiendo la comunicación del módulo ESP32 con la red local para establecer la comunicación de datos.

## Figura 37

*Registro de red y contraseña.*

```
// Reemplazar con sus datos de su red WIFI
const char* ssid = "RED ANDRES";
const char* password = "andresmarzo25@";
```

**Nota:** Se observa el respectivo registro el cual permite la comunicación de la red local con el módulo ESP32.

Se observa en la figura 38 las condiciones en la que el usuario al recibir la alerta pueda activar la alarma mediante la respectiva aplicación.

### Figura 38

*Condiciones de la activación de alarma.*

```
if (text == "/alarma_on") {  
    bot.sendMessage(chat_id, "Alarma activada", "");  
    digitalWrite(buzPin, HIGH);  
}  
  
if (text == "/alarma_off") {  
    bot.sendMessage(chat_id, "Alarma desactivada", "");  
    digitalWrite(buzPin, LOW);  
}
```

**Nota:** Se muestra las condiciones que permiten la activación de alarma.

Para comunicar la alerta al usuario se desarrolló un “BOT” en la aplicación Telegram, por intermedio de “BOTFATHER” como se muestra en la figura 39, para ello la persona debe estar registrado mediante su número celular y luego proceder a la respectiva creación.

### Figura 39

*BotFather.*



**Nota:** Modo BotFather.

En la figura 40 y 41, se observa la opción “/newbot”, la cual nos permite crear un nuevo bot, luego se escoge un nombre, en este caso se lo ha denominado “Security Drone Bot”.

## Figura 40

*Creación del bot.*

You can control me by sending these commands:

```
/newbot - create a new bot  
/mybots - edit your bots
```

**Nota:** Se escoge la opción /newbot para iniciar la creación.

## Figura 41

*Nombre del bot creado.*



**Nota:** Se observa en la figura el nombre del bot “Security Drone Bot.”

Se observa en la figura 42 el token generado por la aplicación, el cual es un código alfanumérico único del usuario que permite el acceso a la API Bot.

## Figura 42

*Token API.*

```
Use this token to access the HTTP API:  
6320301830:AAHDSiErRSSDPG0IWfsWfrPjtGAPZojGDQI  
Keep your token secure and store it safely, it can be used by  
anyone to control your bot.
```

**Nota:** Como se muestra en la imagen se genera un código alfanumérico exclusivo del usuario.

En la siguiente figura 43, mediante la opción “Get My ID”, proporciona al usuario un código numérico, el cual será ingresado en el código del sistema de comunicación para proporcionar la alerta.

### Figura 43

*ID del usuario.*

```
Your user ID: 6339387410
Current chat ID: 6339387410
```

**Nota:** El ID que se muestra en la figura permite el registro para recibir mensajes del sistema.

En la figura 44 se muestra el código token y el ID generado se registra como se observa en la imagen, esto permitirá al usuario recibir el mensaje cuando se ha detectado un movimiento, como se muestra en la figura 45.

### Figura 44

*Ingreso del token e ID en el código.*

```
#define BOTtoken "6320301830:AAHDSiErRSSDPG0IWfsWfrPjtGAPZojGDQI" // Tu Bot Token (Obtener de Botfather)
// Usa @myidbot para averiguar el chat ID
// Tambien necesita hacer click en "start" antes de enviarle mensajes el bot
#define CHAT_ID "6339387410"
```

**Nota:** En la figura se observa el respectivo registro en el código.

**Figura 45**

*Alerta de movimiento detectado .*



**Nota:** En la figura se puede observar la alerta registrada en la cuenta del usuario.

## **5.5. Fase 4 Implementación software de monitoreo.**

Posteriormente se presenta la fase final, en la cual se implementa el código generado para el respectivo funcionamiento.

### **5.5.1. Librerías importadas**

Se utiliza las librerías que se observa en la figura 46, las cuales nos permite importar diferentes funciones que serán ejecutadas en el vuelo autónomo del dron.

**Figura 46**

*Librerías importadas.*

```
import numpy
from djitellopy_reduced import Tello
import cv2 as cv
import numpy as np
```

**Nota:** En la figura se observa las librerías utilizadas para la ejecución del programa.

Se nombra la variable “calibpath” como se muestra en la figura 47, la cual permite ejecutar la calibración de la cámara, esto genera que el dron pueda reconocer el área de vuelo y los diferentes objetos detectados por el sistema de reconocimiento por medio de su cámara, además de acuerdo a los parámetros generados previamente evita la distorsión de imágenes captadas.

### Figura 47

*Calibración de distorsión .*

```
calib_path = ""
camera_matrix = np.loadtxt(calib_path+'cameraMatrix_webcam.txt', delimiter=',')
camera_distortion = np.loadtxt(calib_path+'cameraDistortion_webcam.txt', delimiter=',')
```

**Nota:** Se presenta en la figura la respectiva función que permite calibrar la cámara del dron.

Se incluye las respectivas líneas de código del marcador aruco como se observa en la figura 48, en este caso se nombra con una dimensión de 5x5, el cual ha sido identificado por la cámara del dron, esto permitirá la ejecución del aterrizaje autónomo.

### Figura 48

*Marcador Aruco.*

```
aruco_dict = cv.aruco.getPredefinedDictionary(cv.aruco.DICT_5X5_100)
parameters = cv.aruco.DetectorParameters()
```

**Nota:** Se presenta en la figura el código que permite la identificación del marcador aruco 5x5.

Se configura la calidad de la cámara del dron, en este caso se utiliza una capacidad de 1280x720 como se muestra en la figura 49.

## Figura 49

*Calidad de imagen.*

```
cap.set(cv.CAP_PROP_FRAME_WIDTH, 1280)
cap.set(cv.CAP_PROP_FRAME_HEIGHT, 720)

#-- Font for the text in the image
font = cv.FONT_HERSHEY_PLAIN
```

**Nota:** Se escoge la calidad de imagen a ser utilizada para la cámara del dron.

Se configura los ejes de referencia x, y, z para el respectivo aterrizaje, en la figura 50 se observa el código empleado.

## Figura 50

*Código de ejes de posición.*

```
corners, ids, rejected = cv.aruco.detectMarkers(image=gray, dictionary=aruco_dict, parameters=parameters)

if ids is not None and ids[0] == id_to_find:
    # -- ret = [rvec, tvec, ?]
    # -- array of rotation and position of each marker in camera frame
    # -- rvec = [[rvec_1], [rvec_2], ...] attitude of the marker respect to camera frame
    # -- tvec = [[tvec_1], [tvec_2], ...] position of the marker in camera frame
    ret = cv.aruco.estimatePoseSingleMarkers(corners, marker_size, camera_matrix, camera_distortion)

    # -- Unpack the output, get only the first
    rvec, tvec = ret[0][0, 0, :], ret[1][0, 0, :]

    # -- Draw the detected marker and put a reference frame over it
    cv.aruco.drawDetectedMarkers(frame, corners)
    # cv2.aruco.drawAxis(frame, camera_matrix, camera_distortion, rvec, tvec, 10)
    cv.drawFrameAxes(frame, camera_matrix, camera_distortion, rvec, tvec, 10)

    # -- Print the tag position in camera frame
    str_position = "MARKER Position x=%4.0f y=%4.0f z=%4.0f" % (tvec[0], tvec[1], tvec[2])
    cv.putText(frame, str_position, (0, 100), font, 1, (0, 255, 0), 2, cv.LINE_AA)
```

**Nota:** El código que se muestra en la imagen permite que el dron identifique sus ejes de aterrizaje.

Se implementa el código como se presenta en la figura 51, del sistema de reconocimiento, el



cual detectará los diferentes objetos en la zona de vuelo del dron, además genera el recuadro con las respectivas etiquetas de los nombres de las clases.

### Figura 51

*Código de sistema de reconocimiento.*

```
cv.rectangle(frame, (0, 0), (frame.shape[1], 40), (0, 0, 0), -1)
color = (0, 255, 0)
texto_estado = "Estado: clase"

# Especificamos los puntos extremos del área a analizar
area_pts = np.array([[50, 300], [600, 300], [600, frame.shape[0]], [50, frame.shape[0]]])

# Con ayuda de una imagen auxiliar, determinamos el área
# sobre la cual actuará el detector de movimiento
imAux = np.zeros(shape=(frame.shape[:2]), dtype=np.uint8)
imAux = cv.drawContours(imAux, [area_pts], -1, (255), -1)
image_area = cv.bitwise_and(gray, gray, mask=imAux)
```

**Nota:** Los parámetros que se muestra en la figura permite la detección de objetos.

En la figura 52 se observa el código de la detección de movimiento con los diferentes parámetros establecidos del área de vuelo.

### Figura 52

*Código de detección de movimiento.*

```
# Obtendremos la imagen binaria donde la región en blanco representa
# la existencia de movimiento
fgmask = fgbg.apply(image_area)
fgmask = cv.morphologyEx(fgmask, cv.MORPH_OPEN, kernel)
fgmask = cv.dilate(fgmask, None, iterations=2)

# Encontramos los contornos presentes en fgmask, para luego basándonos
# en su área poder determina si existe movimiento
cnts = cv.findContours(fgmask, cv.RETR_EXTERNAL, cv.CHAIN_APPROX_SIMPLE)[0]
for cnt in cnts:
    if cv.contourArea(cnt) > 500:
        x, y, w, h = cv.boundingRect(cnt)
        cv.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
```

**Nota:** La configuración que se muestra en la figura permite visualizar en el monitor cuando se alerta un movimiento.

## 6. Resultados

### 6.1. Sistema de control autónomo.

En las figuras 53 y 54 se muestra las trayectorias establecidas para las pruebas 1,2 y 3, las cuales se realizará con diferentes distancias y ambientes; es decir, zona interior y exterior de una vivienda.

**Figura 53**

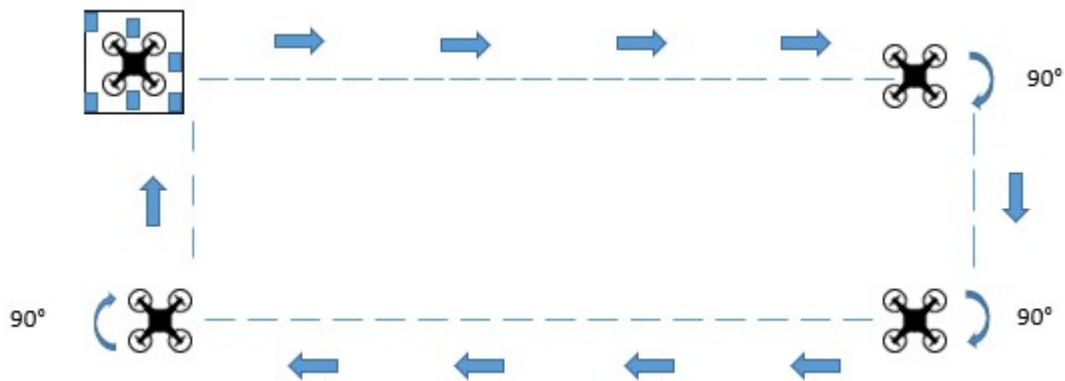
*Trayectoria 1.*



**Nota:** Se observa la primera trayectoria que se establecerá al dron

**Figura 54**

*Trayectoria 2.*



**Nota:** Se observa la segunda trayectoria que se establecerá al dron

### 6.1.1. Sistema de control autónomo en zona interior de la vivienda.

A continuación, se presenta en las siguientes figuras 55, 56, 57, 58, en las cuales se puede observar los resultados de las pruebas del control autónomo del drone; mediante su trayectoria establecida, además por medio de la cámara, el marcador aruco identifica su punto de aterrizaje.

#### Figura 55

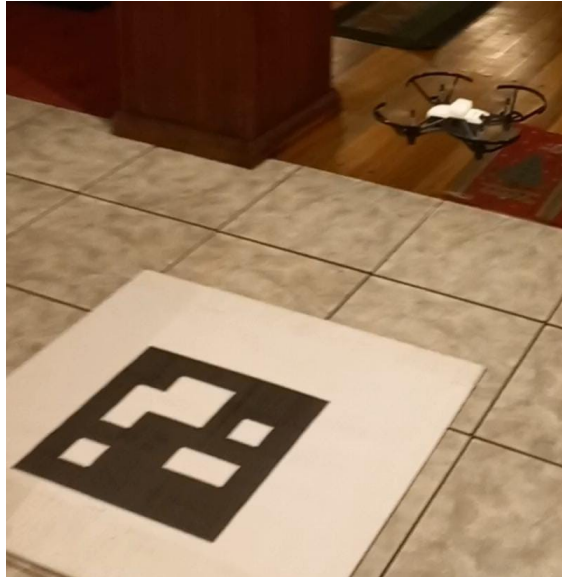
*Inicio de la prueba de vuelo autónomo en zona interior.*



**Nota:** La prueba de control autónomo esta listo para ser ejecutado en una zona interior de la vivienda como se muestra en la figura.

**Figura 56**

*Vuelo autónomo en zona interior.*



**Nota:** Se muestra el resultado de la prueba del control autónomo en ejecución.

**Figura 57**

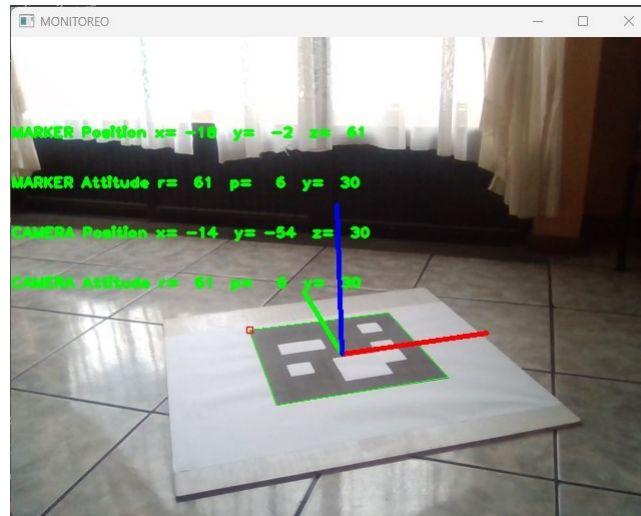
*Prueba de ejecución de vuelo en zona interior.*



**Nota:** Se observa del resultado de la prueba de vuelo en una trayectoria de forma autónoma.

**Figura 58**

*Marcador Aruco.*



**Nota:** La cámara del dron identifica el marcador para su aterrizaje como se observa en la figura.

Se observa en las siguientes tablas 2 y 3 los resultados obtenidos al realizar las pruebas de la trayectoria 1 y 2 respectivamente.

Para el cálculo del error porcentual se utiliza la siguiente fórmula:

$$Ep = \frac{|Vm - Vr|}{Vr} * 100 \quad (6)$$

Donde:

Ep = Error porcentual

Vm = Valor medido

Vr = Valor real

Se multiplica por 100, para obtener en valor porcentual.

**Tabla 2***Resultado de las pruebas realizadas de la trayectoria 1.*

Prueba	Altura	Trayectoria establecida [m]	Trayectoria medida [m]	Tiempo de vuelo [min]	Error Porcentual de trayectoria [%]
1	2	3	3	1.3	0
2	2	4	4.10	2	2.5
3	2	5	4.90	2.3	2
				Promedio error [%]	1.5

**Nota:** En la tabla 2 se describe los parámetros de entrada y salida que se obtuvo al realizar las pruebas.

**Tabla 3***Resultado de las pruebas realizadas de la trayectoria 2.*

Prueba	Altura	Trayectoria establecida [m]	Trayectoria medida [m]	Tiempo de vuelo [min]	Error Porcentual de trayectoria [%]
1	2	6	5.8	2.3	3.33
2	2	7	7.15	3	2.15
3	2	8	7.90	3.3	1.25
				Promedio error [%]	2.25

**Nota:** En la tabla 3 se describe los parámetros de entrada y salida que se obtuvo al realizar las pruebas.

Al realizar las pruebas de la trayectoria 1 y 2 con el dron DJI Tello, se puede comprobar que tiene un error promedio porcentual máximo de 2.25% , ya que se encuentra en un ambiente cerrado en donde la comunicación Wi-Fi es óptima y no existe perturbaciones externas como el viento, que pueda afectar el funcionamiento del dron. El error se procede porque el dron tiene sensores infrarrojos, el cual no permite aterrizar sobre el marcador Aruco que es de color negro, de tal manera el dron por autonomía se desplaza a un espacio del marcador donde no sea de ese color y aterriza automáticamente.

### 6.1.2. Sistema de control autónomo en zona exterior

A continuación, se muestra las figuras 59, 60, 61, 63, 64, 65, 66; los resultados de las trayectorias realizadas en una zona exterior de una vivienda ubicada en la zona rural del cantón “Paute”; parroquia “San Cristóbal”; sector “El Descanso”; en el barrio “Guachun chico”.

#### Figura 59

*Reconocimiento de área.*



**Nota:** Se realiza un reconocimiento del lugar para la ejecución de la prueba.

**Figura 60**

*Reconocimiento de zona de vuelo.*



**Nota:** Se observa la respectiva verificación de la zona de vuelo.

**Figura 61**

*Inicio de la prueba en zona exterior.*



**Nota:** Se realiza la respectiva configuración de los parámetros para el inicio de la prueba.



## Figura 62

*Ejemplo de los parámetros para el inicio de la prueba.*

```
1 from djitellopy import Tello
2 import time
3
4 tello = Tello()
5 tello.connect()
6 print(tello.get_battery())
7
8 tello.takeoff()
9
10 tello.move_forward(500)
11 tello.rotate_clockwise(90)
12 tello.move_forward(150)
13 tello.rotate_clockwise(90)
14 tello.move_forward(150)
15 tello.rotate_clockwise(90)
16 tello.move_forward(150)
17 tello.rotate_counter_clockwise(90)
18 tello.move_forward(340)
19
20 tello.land()
```

**Nota:** Ejemplo de los parámetros de prueba para realizar una trayectoria.

## Figura 63

*Ejecución de la prueba.*



**Nota:** Se muestra el inicio de la prueba del vuelo autónomo.

**Figura 64**

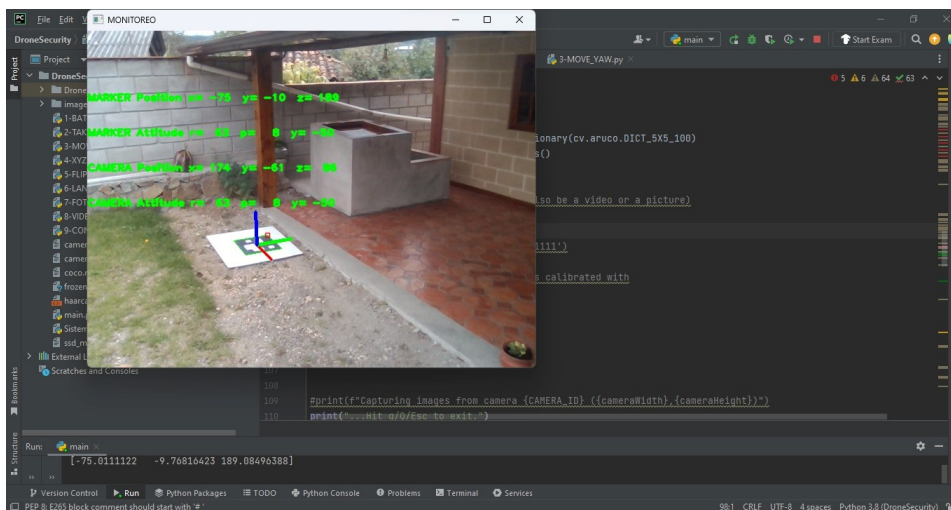
*Drone ejecutando plan de vuelo.*



**Nota:** El drone inicia el plan de vuelo de la prueba.

**Figura 65**

*Marcador Aruco.*



**Nota:** Se observa las respectivas coordenadas para el aterrizaje del drone.

Se observa en las siguientes tablas 4 y 5 los resultados obtenidos al realizar las pruebas en el exterior de la vivienda, con la trayectoria 1 y 2 respectivamente.

**Tabla 4**

*Resultado de las pruebas realizadas de la trayectoria 1.*

Prueba	Altura	Trayectoria establecida [m]	Trayectoria medida [m]	Tiempo de vuelo [min]	Error Porcentual de trayectoria [%]
1	8	16	15.80	2	1.25
2	8	20	19.40	2.2	3
3	8	24	23.35	3	2.71
				Error promedio [%]	2.32

**Nota:** En la tabla 4 se describe los parámetros de entrada y salida que se obtuvo al realizar las pruebas.

**Tabla 5**

*Resultado de las pruebas realizadas de la trayectoria 2.*

Prueba	Altura	Trayectoria establecida [m]	Trayectoria medida [m]	Tiempo de vuelo [min]	Error Porcentual de trayectoria [%]
1	8	24	23.4	2.3	2.5
2	8	30	29	3	3.33
3	8	34	32.5	3.3	4.4
				Error promedio [%]	3.4

**Nota:** En la tabla 5 se describe los parámetros de entrada y salida que se obtuvo al realizar las pruebas.

Al realizar las respectivas pruebas de la trayectoria 1 y 2, en la zona exterior de la vivienda, se puede observar que el error promedio aumentó con respecto a las pruebas realizadas con anterioridad, esto se da por diferentes factores; por ejemplo la señal de la red Wi-Fi disminuye al encontrarse en el exterior, de tal manera que la transmisión de datos del centro de mando (computador) con el dron no sean las óptimas, y de igual manera un factor importante es el viento, el cual afecta al dron DJI Tello, ya que por sus características, no es un dron

robusto para que esté expuesto a este tipo de ambiente, de tal forma afecta la trayectoria del dron haciendo que este pierda distancia entre movimientos y no cumpla de forma efectiva la trayectoria planificada.

## 6.2. Sistema de reconocimiento.

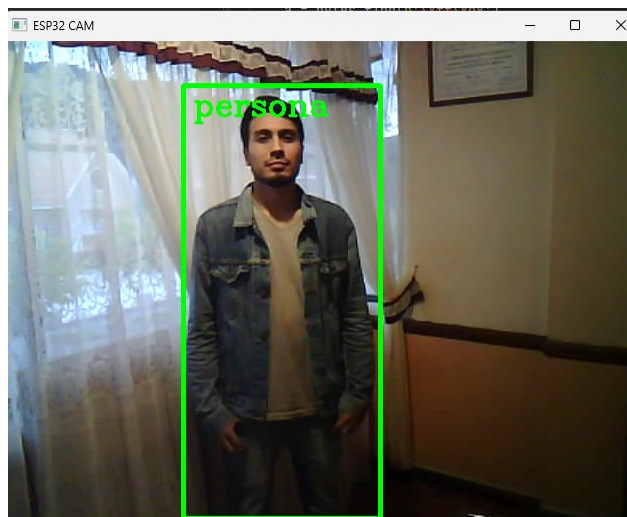
A continuación, se muestran las siguientes figuras con las respectivas pruebas realizadas en diferentes ambientes.

### 6.2.1. Sistema de reconocimiento en zona interior de la vivienda.

Se observa en las figuras 66, 67 y 68 como el sistema de reconocimiento del dron identifica a la persona mediante la detección de objetos de diferentes clases, además otros objetos a su contorno. La prueba se realiza en una zona interior de la vivienda.

#### Figura 66

*Sistema de reconocimiento.*



**Nota:** Identificación de la persona con su respectivo nombre de etiqueta de la respectiva clase.

### Figura 67

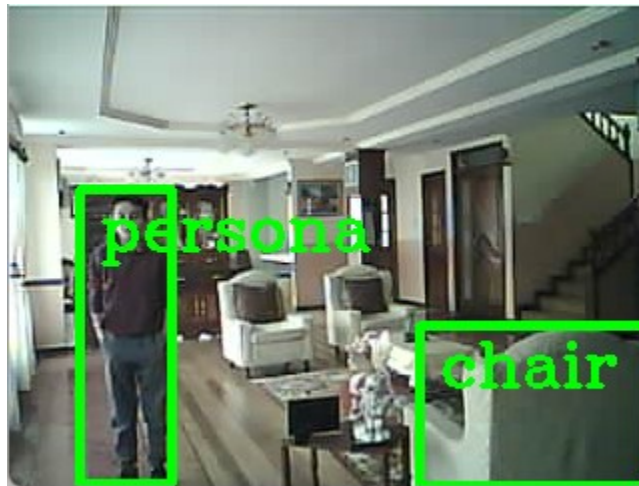
*Persona reconocida mediante cámara.*



**Nota:** El sistema de reconocimiento como se muestra en la figura reconoce una persona en una zona interna de la vivienda, mediante la cámara del dron.

### Figura 68

*Reconocimiento de objetos.*



**Nota:** Se muestra en la figura objetos reconocidos en su contorno.

**Tabla 6**

*Resultado de las pruebas realizadas del sistema de reconocimiento.*

Prueba	Distancia con respecto a la cámara del dron [m]	Cantidad de objetos del área	Cantidad de objetos detectados	Error Porcentual de Reconocimiento [%]
1	3	8	6	25
2	5	9	6	33
3	7	9	5	44
			Promedio error [%]	34

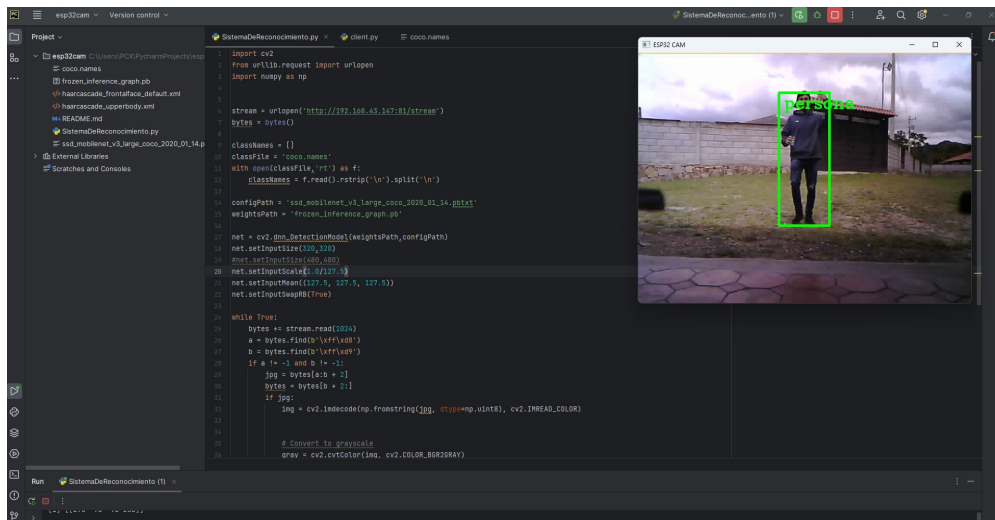
**Nota:** En la tabla 6 se describe los parámetros de entrada y salida que se obtuvo al realizar las pruebas de reconocimiento.

### 6.2.2. Sistema de reconocimiento en zona exterior de la vivienda.

Se presenta las figuras 69, 70 y 71 las cuales son el resultado del sistema de reconocimiento en una zona exterior de la vivienda.

**Figura 69**

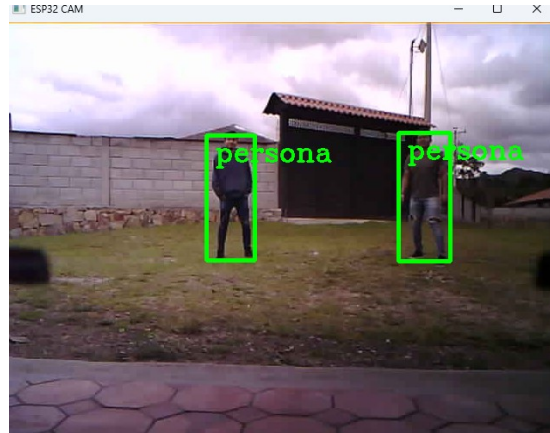
*Sistema de reconocimiento en ejecución.*



**Nota:** El sistema de reconocimiento detecta a dos personas.

**Figura 70**

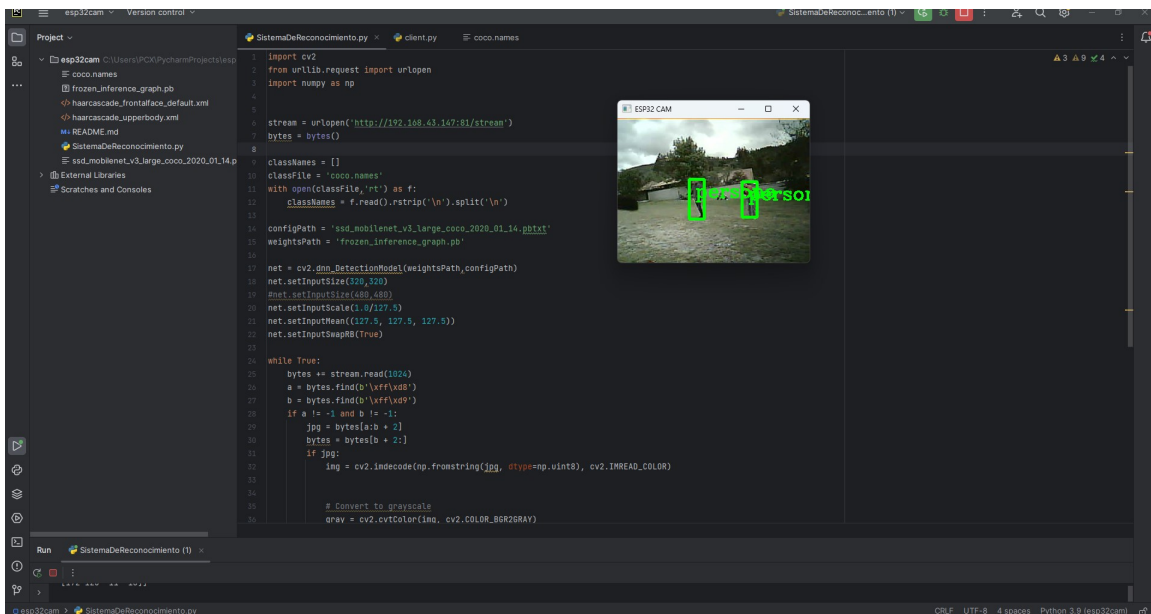
*Reconocimiento de personas.*



**Nota:** Sistema de reconocimiento detectando a las personas en zona exterior.

**Figura 71**

*Sistema de reconocimiento.*



**Nota:** Se observa la ejecución del sistema de reconocimiento en zona de vuelo del dron.

**Tabla 7**

*Resultado de las pruebas realizadas del sistema de reconocimiento.*

Prueba	Distancia con respecto a la cámara del dron [m]	Cantidad de objetos del área	Cantidad de objetos detectados	Error Porcentual de Reconocimiento [%]
1	15	10	8	20
2	18	10	7	30
3	20	13	5	62
			Promedio error [%]	37

**Nota:** En la tabla 7 se describe los parámetros de entrada y salida que se obtuvo al realizar las pruebas de reconocimiento.

Al realizar la prueba de reconocimiento en el exterior de la vivienda el error promedio del sistema es del 37%, ya que existe una variedad de objetos que no están entrenados en "COCO dataset", de igual forma el error se produce por la distancia de los objetos que se encuentra con respecto a la cámara del dron, mientras que el error promedio en las pruebas realizadas en el interior de la vivienda es de 34%, obteniendo como resultado mayor precisión de reconocimiento dentro de la vivienda.

### **6.3. Sistema de comunicación.**

A continuación, se muestra los resultados del sistema de comunicación realizado en una zona interior y exterior.

#### **6.3.1. Sistema de comunicación en zona interior de la vivienda.**

En la figura 72 se observa al sensor PIR detectando la presencia de movimiento, luego en la figura 73 se observa una notificación mediante la aplicación telegram al bot creado, denominado "Security Drone Bot", de esta manera comunicando al usuario de la presencia de movimiento, en la figura 74 se puede observar al dron iniciando su plan de vuelo de forma autónoma.



**Figura 72**

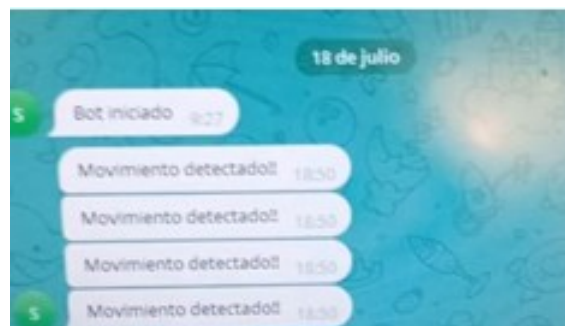
*Sensor PIR.*



**Nota:** En la figura se observa el sensor PIR detectando movimiento.

**Figura 73**

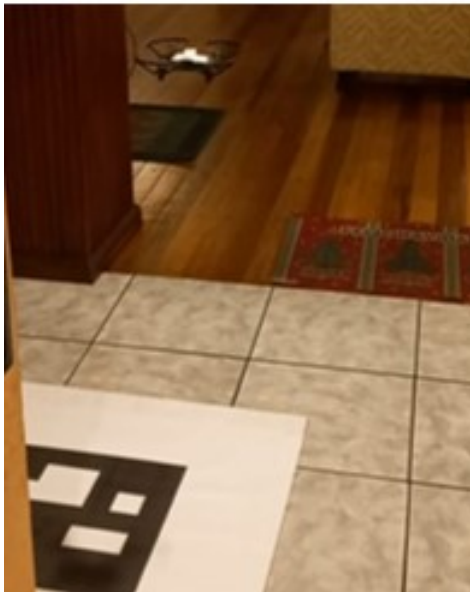
*Notificación de movimiento.*



**Nota:** Se muestra en la figura el mensaje de movimiento detectado.

## Figura 74

*Inicio de vuelo.*



**Nota:** En la figura se presenta el inicio del vuelo del dron de forma autónoma.

### 6.3.2. Sistema de comunicación en zona exterior de la vivienda.

En la figura 75 se observa cuando la persona activa el sensor de movimiento externo mediante su presencia, en la figura76 la notificación al usuario y en la figura 77 el dron iniciando su vuelo.

**Figura 75**

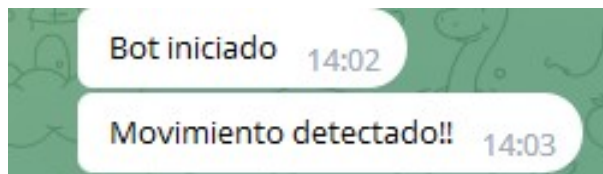
*Sensor PIR en zona exterior.*



**Nota:** Se observa la activación del sensor por medio de la presencia de la persona.

**Figura 76**

*Notificación de movimiento.*



**Nota:** Notificación al usuario del movimiento detectado en la zona.

## Figura 77

*Ejecución del vuelo.*



**Nota:** El drone inicia el vuelo autónomo en la zona exterior.

Al realizar la prueba de comunicación se ha tomado tres distancias diferentes de 6, 9 y 12 metros del sensor con respecto al centro de mando. El sensor en el momento que detecta un movimiento la señal es receptada por el ESP32, el cual, éste es el encargado de transmitir al centro de control mediante Wi-Fi.

## Figura 78

*Código para el tiempo de respuesta del sistema de comunicación.*

```
void loop() {
  int numNewMessages = bot.getUpdates(bot.last_message_received + 1);

  while(numNewMessages) {
    handleNewMessages(numNewMessages);
    numNewMessages = bot.getUpdates(bot.last_message_received + 1);
  }

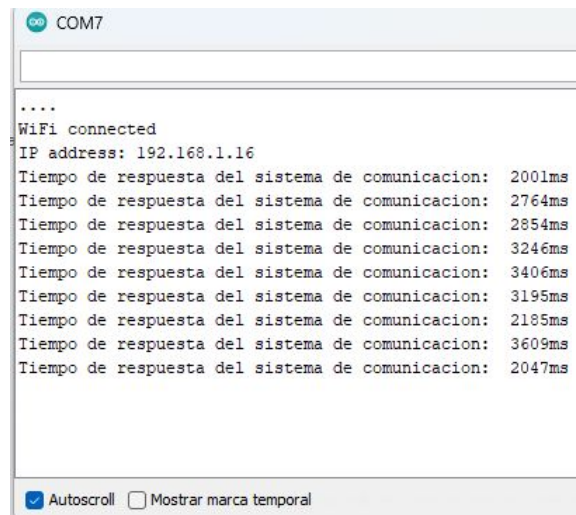
  if(motionDetected){
    unsigned long tiempoInicio = millis();
    bot.sendMessage(CHAT_ID, "Movimiento detectado!!", "");
    motionDetected = false;
    unsigned long tiempoFin = millis();
    unsigned long lapso = tiempoFin - tiempoInicio;
    Serial.print("Tiempo de respuesta del sistema de comunicacion: ");
    Serial.print(lapso);
    Serial.println("ms");

    delay(1000);
  }
}
```

**Nota:** Este código permite visualizar el tiempo de respuesta en el monitor serial del software Arduino IDE.

## Figura 79

*Visualización del tiempo de respuesta.*



The screenshot shows the serial monitor window for COM7. The output text is as follows:

```
....
WiFi connected
IP address: 192.168.1.16
Tiempo de respuesta del sistema de comunicacion: 2001ms
Tiempo de respuesta del sistema de comunicacion: 2764ms
Tiempo de respuesta del sistema de comunicacion: 2854ms
Tiempo de respuesta del sistema de comunicacion: 3246ms
Tiempo de respuesta del sistema de comunicacion: 3406ms
Tiempo de respuesta del sistema de comunicacion: 3195ms
Tiempo de respuesta del sistema de comunicacion: 2185ms
Tiempo de respuesta del sistema de comunicacion: 3609ms
Tiempo de respuesta del sistema de comunicacion: 2047ms
```

At the bottom of the window, there are two checkboxes: "Autoscroll" (checked) and "Mostrar marca temporal" (unchecked).

**Nota:** Se puede observar el tiempo de respuesta en distancias diferentes.

**Tabla 8**

*Resultado de las pruebas realizadas del sistema de comunicación.*

Prueba	Distancia con respecto al centro de mando [m]	Tiempo de respuesta [ms]
1	6	2001
2	9	2764
3	12	3609

**Nota:** En la tabla 8 se describe los resultados que se obtuvo al realizar las pruebas de comunicación.

Los tiempos de respuesta del sistema de comunicación fueron tomadas mediante el software Arduino IDE. Estos resultados presentaron variaciones ya que el sistema de comunicación se alejaba del alcance de la señal Wi-Fi.

## **6.4. Implementación software de monitoreo.**

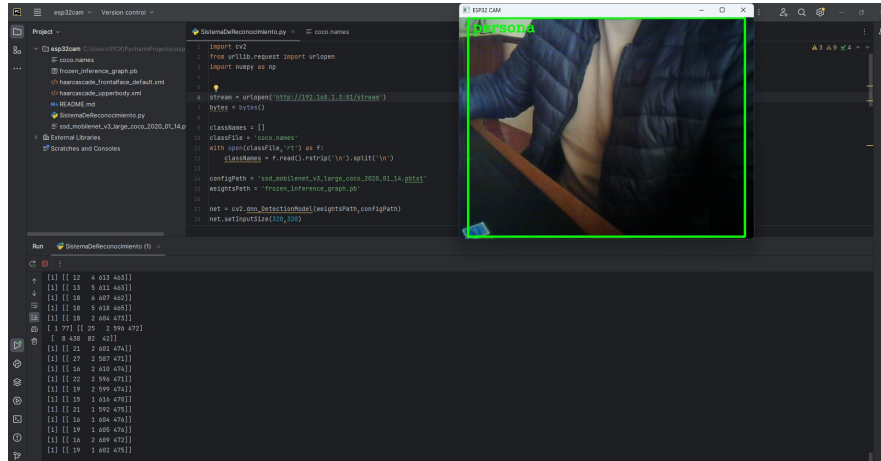
Se muestra las siguientes figuras del resultado de la implementación del respectivo software de monitoreo funcionando en diferentes ambientes de la vivienda.

### **6.4.1. Funcionamiento del software de monitoreo en zona interior de la vivienda.**

A continuación, se muestra en la figura 80 el código implementado en el software, en la figura 81 se muestra la ejecución en el monitor de la computadora, en la cual se podrá monitorear el control autónomo.

**Figura 80**

*Código implementado.*



**Nota:** Se observa en la figura el código en ejecución en zona interior de la vivienda.

**Figura 81**

*Ejecución del código.*



**Nota:** Se muestra en la figura la respectiva ejecución del programa implementado.

#### 6.4.2. Funcionamiento del software de monitoreo en zona exterior de la vivienda.

En las figuras 82 y 83 se muestra el respectivo funcionamiento realizado en un diferente ambiente de prueba con el código implementado en el software de monitoreo

##### Figura 82

*Código implementado.*



**Nota:** Ejecucion del código implementado.

##### Figura 83

*Software de monitoreo.*



**Nota:** Se observa en la figura el monitoreo del sistema ejecutado.



## 7. Conclusiones

Para el desarrollo del sistema de control autónomo se obtuvo la ejecución del objetivo específico planteado, es decir lograr una autonomía del dron sin la presencia de una persona para su pilotaje, además permite realizar movimientos o trayectorias con diferentes niveles de dificultad, que requieren previamente cierto grado de experiencia en este tipo de dispositivos.

El sistema de reconocimiento permite identificar diferentes objetos que se encuentren en la zona de vuelo, de esta manera facilita la identificación de objetos mediante inteligencia artificial. Los objetos detectados tienen su propia etiqueta con su respectivo nombre, de tal manera evita confusiones cuando se ejecuta el vuelo. El sistema de comunicación mediante el sensor de movimiento externo detecta la presencia de la persona, y envía al usuario una notificación en tiempo real, por lo tanto, el sistema ejecuta el vuelo autónomo del dron para realizar el respectivo vuelo en la zona.

El diseño e implementación de un sistema de reconocimiento autónomo para el sector rural mediante un dron sin duda es una propuesta innovadora ya que con esta implementación se puede ayudar a la sociedad en diversos ámbitos, con la combinación de la tecnología e inteligencia artificial. Este tipo de sistema ofrece grandes beneficios para distintas áreas, con la capacidad de detectar objetos como: animales, vehículos o personas; favorece para detectar y prevenir problemas a futuro, además el uso de un dron autónomo impulsa al desarrollo y bienestar de la comunidad. Tomando en cuenta que la privacidad de las personas es algo fundamental y es importante implementar este tipo de proyectos de forma respetuosa y ética.

El dron al ser autónomo en su vuelo permite ser más eficiente en tiempo de planeación con diferentes trayectorias, sin la necesidad de tener a una persona que lo controle, para realizar un plan de vuelo y reconocimiento, se necesita una programación del dispositivo, ya que esto permite la ejecución de los movimientos que son enviados mediante el respectivo software, obteniendo como resultado la recepción de datos beneficiosos para el monitoreo los cuales serán útiles para el usuario.

## 8. Recomendaciones

Se realiza las siguientes recomendaciones para proyectos a futuro con temas relacionados a “Sistemas autónomos mediante drones con reconocimiento de objetos”, se debe tener en cuenta la capacidad de programación que permite el dispositivo, ya que esta puede ser muy limitada y podría presentar inconvenientes al momento de ejecutar los códigos de programación de movimientos o trayectorias generadas, es decir, tener en cuenta la capacidad del procesador del drone, se debe buscar un dispositivo con un sistema operativo actual y que permita actualizaciones, esto permite al drone obtener funciones mejoradas. Además, si se desea implementar un sistema de reconocimiento de objetos las bibliotecas o librerías importadas deben ser las correctas, ya que al no serlas estas no pueden reconocer los objetos en su contorno y presenta problemas de calidad de imagen, se recomienda utilizar librerías propias del drone o del software, ya que estas permiten el soporte necesario de las líneas de código generadas.

Se recomienda que el drone debe tener una capacidad de batería aceptable, como por ejemplo la que posee el drone “Parrot Anafi” que incluye una batería de 2700 mAh que permite un tiempo de vuelo de 25 minutos, además una autonomía de vuelo de 3.8 km y su peso es de 1632 gramos, en comparación con el “DJI Tello” la capacidad de batería es de apenas 1100 mAh con un tiempo de vuelo de 13 minutos y su autonomía de vuelo es de 100 m, al momento de cargar un programa el drone no ejecutara los movimientos adecuados, ya que si la batería se encuentra en un porcentaje menos del 60% no iniciara el vuelo del dispositivo de forma autónoma, se recomienda hacer una previa investigación de diferentes dispositivos programables y verificar las características principales que estos presentan, facilitando la elección del drone.

Para utilizar el sistema de posicionamiento en condiciones de luz no favorables, por medio de los sensores infrarrojo del drone “DJI Tello”, no es el adecuado, se recomienda utilizar como periférico una cámara de visión nocturna que permita la toma de contenido multimedia por medio del calor, en un ambiente sin luz como por ejemplo la “Zemuse XT”, con una capacidad de resolución de 640 x 512 px y un zoom digital máximo x8.

Si el clima presenta variaciones como la velocidad del viento, el “DJI Tello” no es una opción a considerar, es muy limitado en estas condiciones, ya que podría ocasionar accidentes hacia la persona u otros objetos, es recomendable utilizar un drone con mejores prestaciones técnicas, es decir que los motores sea lo suficientemente capaces de enfrentar este tipo de

eventualidades es recomendable utilizar motores brushless KDE con “Controlador electrónico de velocidad” (ESC), ya que controla y ajusta la velocidad y aceleración ante este tipo de eventualidades en el momento de que se ejecute el vuelo.

## Referencias

- Espol. (2017). Drones para vigilancia aérea. *Empresas*, 8-9.
- Estado, F. G. (2021). *Fiscalia general del estado*. Actualizado 2021. [Recuperado de: <https://www.fiscalia.gob.ec/estadisticas-de-robos/> ; Accedido el: 05 de noviembre del 2022].
- Gomis, D. (2018). Rpas, uav y drones: ¿cuáles son las diferencias?
- Iberfdrone. (2021). *Tipos de drones y características*. Actualizado 2021. [Recuperado de: <https://iberfdrone.es/tipos-drones-y-caracteristicas/> ; Accedido el: 20 de noviembre del 2022].
- INEC. (2022). *Instituto nacional de estadísticas y censos*. Actualizado 2022. [Recuperado de: <https://www.ecuadorencifras.gob.ec/justicia-y-crimen/> ; Accedido el: 05 de noviembre del 2022].
- Jiménez Bravo, R. (2018). *Sistema de seguimiento de objetos usando opencv, aruco y filtro de kalman extendido* (Inf. Téc.). Publicacion de la Escuela Técnica Superior de Ingeniería.
- Konstantoudakis, C. K. T. D. S. D. A. G. D. A. . D. P., K. (2022). Drone control in ar: An intuitive system for single-handed gesture control, drone tracking, and contextualized camera feed visualization in augmented reality. *Drones*, 43..
- Ledesma, R. F. . P. A., N. (2019). Cátedra de teledetección los drones y sus aplicaciones a la ingeniería.
- O’Shea, J. (2006). *¿qué es python?* [Recuperado de: <https://aws.amazon.com/es/what-is/python/> ; Accedido el: 18 de Julio de 2023].
- ProjectGo, C. (2022). *Projectgo corporation web site*. [Recuperado de: <https://projectgo.pro/es/quienes-somos/> ; Accedido el: 18 de Julio de 2023].
- Revista, V. (2022). *Ecuador fue el tercer país de latinoamerica con mas robos a viviendas llevados a cabo durante el inicio del 2022 segun estudio lh3977195*. Actualizado 2022. [Recuperado de: <https://www.vistazo.com/actualidad/nacional/ecuador-fue-el-tercer-pais-de-latinoamerica-con-mas-robos-a-viviendas-llevados-a-cabo-durante-el-inicio-del-2022-segun-estudio-LH3977195> ; [Accedido el: 05 de noviembre del 2022].
- Richard, A. (2018). *Drones la tecnologia a control remoto*. Actualizado 2021. [Recuperado de: <https://dialoguemos.ec/2018/04/drones-la-tecnologia-a-control-remoto/> ; Accedido el: 01 de diciembre del 2022].
- Suh & Nam, C. (2020). A study on operational ability comparing drone-centric and user-centric control in external piloting.

Tech, R. (2018). Manual de usuario v1.0. *Ryze Tech*.

Tech, R. (2023). *Ryze tello robotics*. [Recuperado de: <https://www.ryzerobotics.com/tello>; Accedido el: 18 de Julio de 2023].

Unipython. (2018). *Calibración de la cámara opencv*. [Recuperado de: <https://unipython.com/calibracion-la-camara-opencv/> ].

Çetin, B. C. . P. E., E. (2021). Improving real-time drone detection for counter-drone systems. *The Aeronautical Journal, 1871-1896*.

# **ANEXOS**

## Anexo A: Estados del drone

La tabla 9 presenta los estados del drone.

**Tabla 9**

*Estados del drone.*

Estados normales	Color	Patrón	Estado de la aeronave
	Intermitente entre rojo, verde y amarillo.	Parpadea	Encendido y realizando pruebas de autodiagnóstico
	Verde	Parpadea dos veces de forma constante	Sistema de posicionamiento visual activado
	Amarillo	Parpadea lentamente	No disponible el sistema de posicionamiento visual, modo Atti activado
<b>Estado de carga</b>			
	Azul	Luz fija	Carga completa
	Azul	Parpadea lentamente	Cargando
	Azul	Parpadea rápidamente	Error de carga
<b>Estados de advertencia</b>			
	Amarillo	Parpadea rápidamente	Pérdida de señal del control remoto
	Rojo	Parpadea lentamente	Advertencia de nivel
	Rojo	Parpadea rápidamente	Batería baja
	Rojo	Luz fija	Error de carga

**Nota:** En la tabla 9 se describe los estados del drone.

## Anexo B: Especificaciones importantes del drone

La tabla 10 presenta las especificaciones importantes del drone.

**Tabla 10**

*Características técnicas del drone.*

Especificaciones	Distancia con respecto al sensor PIR [m]
Peso	10.087 [kg]
Velocidad	17.9 [mph]
Tiempo de vuelo	15 [min]
Cámara	1280 x 720 HD
Capacidad batería	1100 [mAh]
Potencia de carga	10 [W]
Grado de temperatura adecuado	32° - 104° F
Grado de temperatura de carga	42° - 114° F
Voltaje	3.8 [V]

**Nota:** En la tabla 10 se describe las características técnicas del drone.

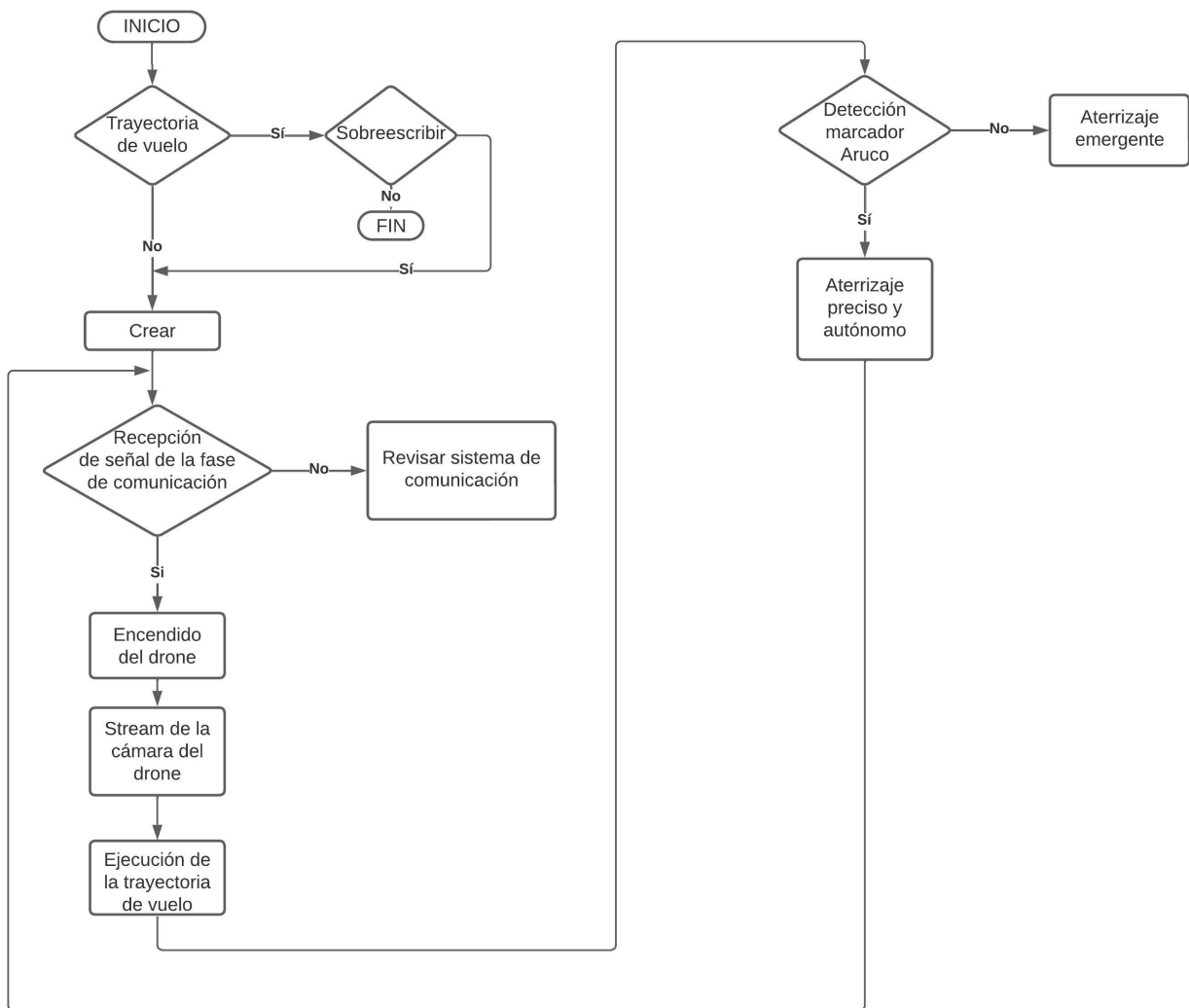


## Anexo C: Diagrama de flujo de la fase autónoma.

En el diagrama de flujo de la figura 84 se observa el funcionamiento de la programación de la fase autónoma.

**Figura 84**

*Diagrama de flujo fase autónoma.*



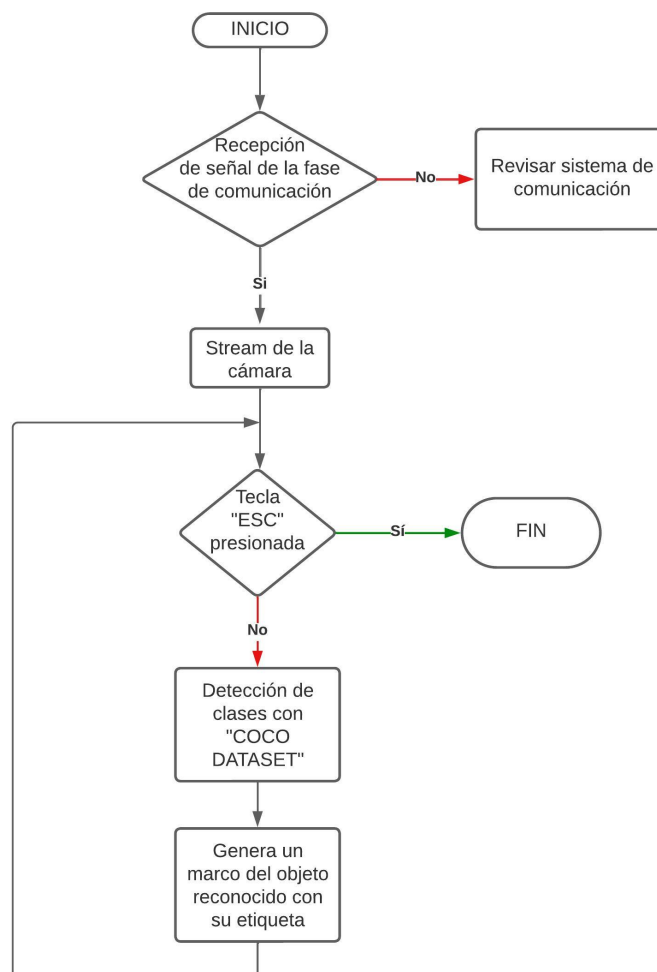
**Nota:** Funcionamiento de la programación de la fase autónoma.

## Anexo D: Diagrama de flujo de la fase de reconocimiento.

En la figura 85 se observa mediante un diagrama de flujo el funcionamiento de la programación de la fase de reconocimiento

**Figura 85**

*Diagrama de flujo fase de reconocimiento.*



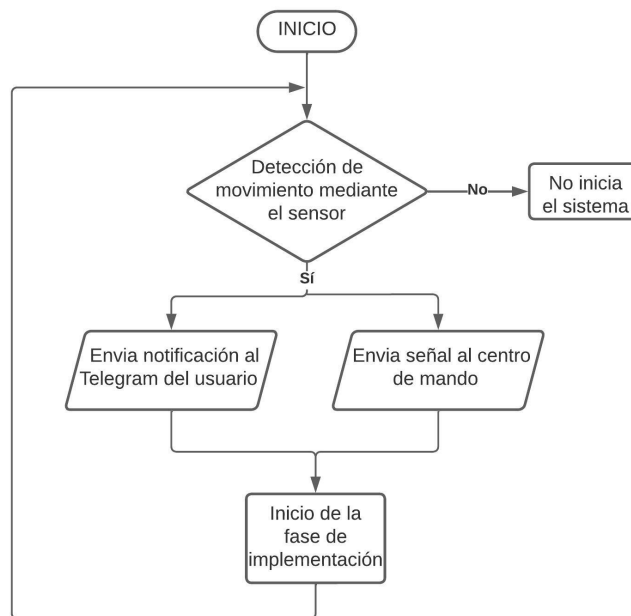
**Nota:** Funcionamiento de la programación de la fase de reconocimiento.

## Anexo E: Diagrama de flujo de la fase de comunicación.

El diagrama de la figura 86 se observa el funcionamiento de la programación de la fase de comunicación.

**Figura 86**

*Diagrama de flujo fase de comunicación.*



**Nota:** En el diagrama se observa el funcionamiento de la programación de la fase de comunicación.