



**SISTEMA VISUAL CON REFERENCIA DE APRILTAGS
PARA DESPLAZAMIENTO DE UN DRÓN**

TESIS

PARA OBTENER EL GRADO DE

**MAESTRO EN
SISTEMA INTELIGENTES MULTIMEDIA**

PRESENTA

**ING. FRANCISCO JAVIER CAMPOS NAVARRO
ASESOR: DR. LUIS ENRIQUE RAMOS VELASCO**

QUERÉTARO, QUERÉTARO, SEPTIEMBRE DE 2020.

CARTA DE LIBERACIÓN DEL ASESOR



26 de febrero del 2020.

Mtro. Geovany González Carlos
Coordinador Académico de Posgrado
CIATEQ, A.C.

El abajo firmante, miembro del Comité Tutorial del Ing. Francisco Javier Campos Navarro, una vez revisado su Proyecto Terminal de tesis, titulado "Sistema visual con referencia de AprilTags para desplazamiento de un drone" autorizo que el citado trabajo sea presentado por el alumno para su revisión, con el fin de alcanzar el grado de Maestría en Sistemas Inteligentes Multimedia.

Sin otro particular por el momento, agradezco la atención prestada.

A handwritten signature in black ink, consisting of several loops and a long horizontal stroke at the end, positioned above a solid horizontal line.

Dr. Luis Enrique Ramos Velasco

Asesor Académico

CARTA DE LIBERACIÓN DEL REVISOR



**GOBIERNO DE
MÉXICO**



CONACYT
Consejo Nacional de Ciencia y Tecnología



Pachuca, Hidalgo, 12 de julio del 2020.

Dra. María Guadalupe Navarro Rojero
Directora de Posgrado
CIATEQ, A.C.

Por medio de la presente me dirijo a usted en calidad de Revisor del proyecto terminal del alumno Francisco Javier Campos Navarro, cuyo título es:

"SISTEMA VISUAL CON REFERENCIA DE APRILTAGS
PARA DESPLAZAMIENTO DE UN DRÓN"

Después de haberlo leído, corregido e intercambiado información con el alumno, y realizado los cambios que le fueron sugeridos, puede ser autorizada su impresión, a fin de que se inicien los trámites correspondientes para su defensa.

Sin otro particular por el momento, y en espera de que mis sugerencias sean tomadas en cuenta en beneficio del estudiante y la Institución, agradezco la atención prestada.

Atentamente,

Dra. Carolina Reta Castro.
Investigadora en CIATEQ. Cédula: 09122140.
Doctora en ciencias en el área de Ciencias Computacionales.

"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

DEDICATORIA

A todas mis fuentes de inspiración, sombras y luces, los busco siempre tener cerca, siempre que puedo y también a mis archienemigos (miedos y límites) por llevarme hasta donde estoy hoy, a casi pararme a un lado de gigantes y ver hacia adelante.

AGRADECIMIENTOS

No es solo jugar con la superficie, es poder entender las piezas pequeñas y llegar a crear algo nuevo con ellas; así empezó con preguntas pequeñas a papá y a mamá cuestionando hasta la simple sombra de una hermana traviesa y a buscar la forma más elaborada de unir legos con los hermanos pequeños y desarmar el mundo en pequeñas piezas para buscar soluciones complejas a cuestiones simples.

Gracias a mamá y papá por ser como son; sin ustedes no sería como soy.

Gracias a todos, asesores, profesores y compañeros, en especial a mi compañera de equipo de toda la maestría que me brindó su apoyo para la realización de mis estudios de Posgrado.

Francisco Javier Campos Navarro

Querétaro, Querétaro

julio de 2020

* * *

RESUMEN

En esta tesis se presenta un algoritmo de asistencia de navegación en el desplazamiento de un micro drón para vuelo semiautónomo en interiores por medio del sistema de visión y marcadores AprilTag, como alternativa a otras infraestructuras. El desplazamiento del drón es guiado por la detección de las AprilTags por medio del sistema de visión; el algoritmo asigna los movimientos o acciones según el marcador visual detectado. El algoritmo de desplazamiento provee la funcionalidad de volar de forma semiautónoma, al colocar estas etiquetas bidimensionales en el campo de vuelo. El drón utilizado para este trabajo es un modelo Tello. El lenguaje de programación utilizado es Python con librerías de OpenCV, AprilTag que son de código abierto.

Se evaluó la repetibilidad del algoritmo realizando pruebas en 2 escenarios en interior con condiciones de luz controladas, sin viento y sin obstáculos en el campo de vuelo. Se logró la detección de AprilTags en vuelo y recorrido de los trayectos. Los resultados se obtuvieron comparando las distancias obtenidas de los experimentos de los movimientos individuales contra la distancia de los experimentos finales; donde se combinan acciones y movimientos. Las distancias se midieron desde el punto de despegue al punto de aterrizaje. Se consideraron las distancias obtenidas a través de los experimentos individuales como distancias finales teóricas a comparar contra los resultados de los experimentos de los escenarios finales. En el escenario 1, de los 10 vuelos se tiene una distancia al punto de despegue de 2.88 m con un error relativo de 6% y en el escenario 2 se obtiene una distancia al punto de despegue de 2.93 m con un error relativo de 3 %, logrando un aterrizaje dentro de un área de 35 cm del destino final teórico. Se pudo observar que el sistema de visión dota al drón con la capacidad de guiar al drón en vuelo semiautónomo en interior, para completar una trayectoria con la detección de los marcadores AprilTags. Mediante el algoritmo desarrollado y nuevas mejoras se podrán cubrir actividades para los drones en interiores.

Palabras clave: Drón vuelo semiautónomo, Sistema de visión, Detección de AprilTags, Tello, TelloPy.

ABSTRACT

In this thesis, an algorithm for navigation assistance in the movement of a micro drone for semi-autonomous flight in indoors is presented. The algorithm uses the AprilTag vision system and markers, as an alternative to other infrastructures. The drone movement is guided by the detection of AprilTags through the vision system; the algorithm assigns movements or actions according to the detected visual marker. The navigation assistance algorithm provides the functionality to fly semi-autonomously by placing two-dimensional AprilTags labels in the flight field.

The repeatability of the algorithm was evaluated by carrying out tests in 2 indoor scenarios with controlled light conditions, without wind and obstacles in the flight field. In the scenarios, the following actions were performed: takeoff, forward displacement, then another lateral displacement, and finally landing. This allows achieving the detection of AprilTags in flight and completion of flying routes. The results were obtained by comparing the distances from the experiments of the individual movements against the distance of the final experiments, where drone actions and movements were combined. Distances were measured between take-off and landing points through individual experiments. Theoretical final distances were used to compare achieved results in the final scenarios. In scenario 1, based on 10 flights, there is a distance of 2.88 m between take-off and landing points, with a relative error of 6% against ground truth. In scenario 2, there is a distance of 2.93 m between take-off and landing points, and a relative error of 3% against ground truth. Conducted experimentation reveals that the drone can achieve a landing within an area of 35 cm from the theoretical destination. Also, it was observed that the vision system provides the drone with the ability to fly semi-autonomously in indoors and to complete a flying trajectory through the detection of AprilTags markers.

By the developed algorithm and new improvements, it will be possible to cover indoor activities for drones that will help to perform repetitive or dangerous tasks for humans.

Keywords: Semi-autonomous flight drone, Vision system, Detection of AprilTags, Tello, Tellopy.

ÍNDICE DE CONTENIDO

CARTA DE LIBERACIÓN DEL ASESOR	I
CARTA DE LIBERACIÓN DEL REVISOR	II
DEDICATORIA	III
AGRADECIMIENTOS.....	IV
RESUMEN	V
ABSTRACT	VI
ÍNDICE DE CONTENIDO	VII
ÍNDICE DE FIGURAS.....	X
ÍNDICE DE TABLAS	XIII
LISTA DE ABREVIATURAS	XV
ACRÓNIMOS	XVI
CAPÍTULO 1 INTRODUCCIÓN.....	1
1.1 ANTECEDENTES	1
1.2 PLANTEAMIENTO DEL PROBLEMA	2
1.3 JUSTIFICACIÓN.....	2
1.4 HIPÓTESIS	3
1.5 OBJETIVOS	3
1.5.1 Objetivo general.....	3
1.5.2 Objetivos específicos	3
1.6 METODOLOGÍA	4
1.7 ALCANCES Y LIMITACIONES.....	4
CAPÍTULO 2 MARCO TEÓRICO.....	6
2.1 EVOLUCIÓN DE LA NAVEGACIÓN.....	6
2.2 ELEMENTOS Y SENSORES DE UN DRÓN	7
2.3 NAVEGACIÓN.....	9
2.4 NAVEGACIÓN INERCIAL.....	9
2.4.1 Giroscopio.....	10
2.4.1 Acelerómetro	10
2.5 SISTEMAS DE POSICIONAMIENTO	12
2.5.1 Sistemas de posicionamientos para exteriores	13
2.5.2 Sistemas de posicionamiento para interiores.....	13
2.6 CLASIFICACIÓN Y TIPOS DE DRONES	16
2.6.1 Clasificación de drón por peso promedio de despegue (PPD).....	16
2.6.2 Clasificación del drón por el nivel de autonomía	17
2.6.3 Clasificación del drón por el número de motores y por la forma de su estructura	18

2.7 CUADRICÓPTERO	19
2.7.1 Movimientos de un cuadricóptero	20
2.7.2 Yaw o movimiento de guiñada.....	21
2.7.3 Pitch o movimiento de cabeceo	21
2.7.4 Roll o movimiento de alabeo	22
2.8 CUADRICÓPTERO TELLO.....	22
2.9 MARCADORES VISUALES	24
2.9.1 AprilTags	27
2.9.2 Funcionamiento del algoritmo de las AprilTag.....	30
CAPÍTULO 3 TRABAJO PROPUESTO	36
3.1 ALGORITMO DE ASISTENCIA DE NAVEGACIÓN EN EL DESPLAZAMIENTO DE UN DRÓN	36
3.2 COMPONENTES DEL SISTEMA.....	37
3.2.1 Cuadricóptero Tello.....	37
3.2.2 Sistema de control de tierra	38
3.2.3 Marcadores Visuales AprilTag.....	38
3.2.4 Campo de vuelo	39
3.3 ENTORNO DE DESARROLLO	39
3.3.1 Iluminación del campo de vuelo.....	41
3.3.2 Vuelos del drón controlados desde sistema de control de tierra	43
3.3.3 Transmisión de imágenes del drón al sistema de control de tierra.....	43
3.3.4 Detección de AprilTags de imágenes enviadas desde el drón al sistema de control de tierra.....	44
3.3.5 Angulo de detección de AprilTag	47
3.3.6 Vuelo con detección de AprilTag.....	49
3.3.7 Capas del sistema y su estructura.....	53
3.3.8 Desplazamientos medidos después de detección de AprilTag.....	54
3.4 ESCENARIOS CON COMBINACIÓN DE MOVIMIENTOS CON APRILTAG.	60
3.5 DESTINO FINAL TEÓRICO PARA ESCENARIOS CON COMBINACIÓN DE MOVIMIENTOS CON APRILTAG.....	62
CAPÍTULO 4 RESULTADOS.....	65
4.1 MEDICIÓN DE DESPLAZAMIENTO DE LOS ESCENARIOS.....	65
4.1.1 Resultados de escenario uno: avanzar y desplazarse a la derecha	66
4.1.2 Resultados de escenario dos: avanzar y desplazarse a la izquierda.....	68
4.2 POSICIONES DE LOS ATERRIZAJES DEL DRÓN DE LOS ESCENARIOS 1 Y 2	70
4.3 CÁLCULO DE DESPLAZAMIENTOS TEÓRICOS ESPERADOS	72
4.4 COMPARACIÓN DEL DESTINO FINAL TEÓRICO Y LA DISTANCIA AL PUNTO DE DESPEGUE, ERROR ABSOLUTO Y ERROR RELATIVO DE LOS RESULTADOS ESCENARIO UNO Y DOS.....	76
CAPÍTULO 5 CONCLUSIONES.....	81
TRABAJOS A FUTURO.....	82

APORTACIÓN DE LA TESIS	83
APORTACIÓN SOCIAL DE LA TESIS	84
REFERENCIAS BIBLIOGRÁFICAS	85
ANEXO	

ÍNDICE DE FIGURAS

Figura 1 Componentes básicos de un drón	8
Figura 2 Elementos de un drón por zonas [potencia, transmisión, sensores y controlador]	9
Figura 3 Principio de funcionamiento de un acelerómetro.....	11
Figura 4 INU plataforma estabilizada (gimbaled) [10].....	12
Figura 5 IMU plataforma fija (strapdown) [10]	12
Figura 6 Laboratorio de UWB de Sport Science Laboratory-Jacques Rogge en la Universidad de Ghent [15]	14
Figura 7 Banco de pruebas o Testbed en interiores [16].....	14
Figura 8 Ejemplo de posicionamiento basado en marcadores visuales [21].	15
Figura 9 Clasificación de los vehículos aéreos no tripulados de acuerdo con el tipo de despegue y aterrizaje.....	17
Figura 10 Estructura multirrotores [24].....	19
Figura 11 Modelo del cuadricóptero con marco de referencia inercial respecto a tierra {I} y con marco de referencial inercial respecto al cuerpo del drón {C}.....	20
Figura 12 Movimiento de guiñada (yaw) [24].....	21
Figura 13 Movimiento de cabeceo (pitch) [24].	22
Figura 14 Movimiento de alabeo (roll) [24].	22
Figura 15 Drón Tello Ryze Tech [26].....	23
Figura 16 Sistema visual de posicionamiento [26].	23
Figura 17 Mensaje alerta y advertencia de la aplicación Tello [27].	24
Figura 18 Ejemplo de marcadores visuales de izquierda a derecha primera fila: ARToolKit, Circular Data Matriz, ARToolKit Plus y Maxicode, segunda fila de izquierda a derecha: RuneTag, BlurTag, CanTag y Fourier Tag [29].	25
Figura 19 Contorno módulos píxel [20].....	28

Figura 20 Generación de AprilTag 36h11 a) Descomposición número hexadecimal en número binario 8bits, b) Matriz de llenado de hexadecimal a binario, c) Colocación de 9 dígitos hexadecimal a binario en matriz, d) AprilTag 36h11 ID 0.	29
Figura 21 Familias de AprilTags Tag16h5, Tag25h9 y Tag36h11 [19].....	29
Figura 22 Marco de coordenadas de AprilTag [19].....	30
Figura 23 Imagen de referencia para procesar [19].	31
Figura 24 (izquierda) Magnitud de gradiente, (derecha) Dirección del gradiente. [19]	32
Figura 25 Ponderado de bordes [19].	33
Figura 26 Segmentación lineal [19].	34
Figura 27 Detección de Quads con adquisición de carga útil [19].	34
Figura 28 Etapas de funcionamiento del algoritmo.....	36
Figura 29 Distribución de fuentes de luz en el campo de vuelo y áreas de medición. ..	42
Figura 30 Pruebas transmisión de imagen.	44
Figura 31 Transmisión de imágenes. A) transmisión de imágenes del drón a la laptop (1), B) transmisión de imagen del drón(2) respuesta de acción o movimiento a realizar por el drón cuando AprilTag es detectada (3).	45
Figura 32 AprilTag Tag_ID 7 y Tag_ID 1 foto capturada desde Tello.....	46
Figura 33 AprilTag Tag_ID 7, Tag_ID 1 y Tag_ID 0 foto capturada desde Tello.....	46
Figura 34 AprilTag con un ángulo respecto a la cámara del drón a 0°, 45°, 60°.	47
Figura 35 Detección de AprilTag en vuelo. Ciclo anidado de detección de AprilTags (Etapa 2, 3 y 4).	50
Figura 36 Lógica del algoritmo de detección de AprilTags.	52
Figura 37 Ventana de detección de AprilTag (A), Consola de ejecución del programa (B), Ventana de IDE de programación (C).	53
Figura 38 Esquema de componentes de software y comunicación entre dispositivos. ..	54
Figura 39 Resultados movimiento adelante (distancia en metros).	55

Figura 40 Resultados movimiento atrás (distancia en metros).....	56
Figura 41 Resultados movimiento derecha (distancia en metros).	57
Figura 42 Resultados movimiento izquierda (distancia en metros).....	58
Figura 43 Escenario movimiento derecha: Tag_ID 9, 1, 3, 0.....	61
Figura 44 Escenario movimiento izquierda: Tag_ID 9, 1, 4, 0.....	62
Figura 45 Distancia final teórica en escenario uno y dos.....	64
Figura 46 Medición de desplazamiento del punto de despegue al punto de aterrizaje. Los triángulos no muestran aterrizajes reales, solo son demostrativos.	66
Figura 47 Gráfica de resultados desplazamiento escenario movimiento derecha Tag_ID 9, 1, 3, 0.	68
Figura 48 Gráfica de resultados desplazamiento escenario movimiento izquierda Tag_ID 9, 1, 4, 0.	70
Figura 49 Mapa de resultados. Escenario uno (lado izquierdo con triángulos en color morado) y escenario dos (lado derecho con triángulos en color amarillo).....	71
Figura 50 Mapa de resultados con áreas de aterrizaje (circunferencias con 35cm de radio y 50 cm de radio). Escenario uno (lado izquierdo con triángulos en color morado) y escenario dos (lado derecho con triángulos en color amarillo.....	72
Figura 51 Destino calculado final del drón.....	73

ÍNDICE DE TABLAS

Tabla 1 Clasificación de los drones de acuerdo con el criterio peso promedio de despegue [22].....	16
Tabla 2 Clasificación de los drones según nivel de autonomía [23].....	17
Tabla 3 Atributos principales marcadores visuales.....	26
Tabla 4 Familia AprilTag ID etiquetas [19].....	27
Tabla 5 Iluminación medida de las áreas del campo de vuelo.....	43
Tabla 6 Resultados de detección de AprilTags 8x8cm y de 16.5x16.5cm.....	48
Tabla 7 Tag_ID asignada a cada movimiento o instrucción.....	51
Tabla 8 Resultados movimiento adelante.....	55
Tabla 9 Resultados movimiento atrás.....	56
Tabla 10 Resultados movimiento derecha.....	57
Tabla 11 Resultados movimiento izquierda.....	58
Tabla 12 Resultados desplazamiento AprilTag Tag_ID.....	59
Tabla 13 Desplazamientos promedios de las AprilTag.....	63
Tabla 14 Distancias escenario uno. A) Distancia del punto de despegue al punto de aterrizaje. B) Relación de aterrizajes dentro de áreas de aterrizaje con circunferencia de 35cm y de 50cm de radio.....	67
Tabla 15 Distancias escenario dos. A) Distancia del punto de despegue al punto de aterrizaje. B) Relación de aterrizajes dentro de áreas de aterrizajes con circunferencia de 35cm y de 50cm de radio.....	69
Tabla 16 Cálculo de destino final de los escenarios uno (movimientos adelante y a la derecha Tag_ID 9, 1, 3, 0) y dos (movimientos adelante y a la izquierda Tag_ID 9,1, 4, 0).	75
Tabla 17 Distancia del escenario uno comparado contra valor teórico, error absoluto y error relativo, a partir de resultados tomados de la Tabla 14.....	77
Tabla 18 Error absoluto y error relativo de distancia promedio del escenario uno.	78

Tabla 19 Distancia escenario dos comparado contra valor teórico, error absoluto y error relativo, a partir de resultados tomados de la Tabla 15.....	79
Tabla 20 Error absoluto de distancia promedio y error relativo de distancia promedio del escenario dos.	80

LISTA DE ABREVIATURAS

Abreviatura	Descripción
AGC	Controlado guía del Apollo (en inglés <i>Apollo Guidance Computer</i>)
ESC	Controlador electrónico de velocidad (en inglés <i>Electronic Speed Control</i>)
FC	Controlador de vuelo (en inglés <i>Flight Controller</i>)
GNSS	Sistema global de navegación por satélite (en inglés <i>Global Navigation Satellite System</i>)
GPS	Sistema de posicionamiento global (en inglés <i>Global Positioning System</i>)
IMU	Unidad Inercial de Medición (en inglés <i>Inertial Measuring Unit</i>)
INS	Sistema Navegación Inercial (en inglés <i>Inertial Navigation System</i>)
IPS	Sistema de Posicionamiento en Interiores (en inglés <i>Indoor Positioning System</i>)
IRU	Unidad de Referencia Inercial (en inglés <i>Inertial Reference Unit</i>)
LiPo	Polímero de Litio
MEMS	Sistemas Micro Electromecánicos (en inglés <i>Micro-Electro-Mechanical Systems</i>)
RC	Radio Control
UAV	Vehículo aéreo no tripulado (en inglés <i>Unmanned Aerial Vehicle</i>)
VPS	Sistema visual de posicionamiento (en inglés <i>Vision Positioning System</i>)
VPU	Unidad de Procesamiento de Visión (en inglés <i>Virtual Processing Unit</i>)

ACRÓNIMOS

16h5	Familia AprilTag matiz de 4x4 distancia Hamming 5
25h7	Familia AprilTag matiz de 5x5 distancia Hamming 7
36h11	Familia AprilTag matiz de 6x6 distancia Hamming 11
6-DOF	Seis grados de libertad
ASCII	American Standard Code for Information Interchange
CW	Sentido horario
CCW	Girando en sentido antihorario
D. Máximo	Desplazamiento máximo
D. Mínimo	Desplazamiento mínimo
D. Prom	Desplazamiento promedio
F1, F2, F3, F4	Fuerzas de empuje
FOV	Campo de visión
mAh	miliampere hora
PPD	Peso promedio de despegue
RPA	Remotely Piloted Aircraft
RPAS	Remotely Piloted Aircraft System
Tag_ID	Identificador de etiqueta
UWB	Radio de banda ultra ancha
VEA	Valor de experimentos anteriores
VPTM	Valor Promedio de todos los movimientos
VPU	Unidad de procesamiento de visión
VR	Valor Real
$\omega_1, \omega_2, \omega_3, \omega_4$	Movimientos giratorios

CAPÍTULO 1 INTRODUCCIÓN

1.1 ANTECEDENTES

Los drones son vehículos aéreos controlados de forma remota o autónoma; los sensores acoplados a estos los dotan para diferentes escenarios. Los vuelos en exteriores se asisten del Sistema global de navegación por satélite (GNSS, por sus siglas en inglés) para una precisa navegación, pueden contar con 3 antenas de GNSS y varios sensores de sistemas inerciales de referencia (IRS, por sus siglas en inglés) para tener redundancia en vuelos precisos en exteriores; los drones son versátiles porque se les pueden acoplar infinidad de elementos para realizar toda clase de actividades en vuelos en exteriores, como detección de sembradíos sanos y fumigación de sembradíos donde son detectadas plagas; los drones realizan trabajos que son peligrosos para los humanos como revisión de estructuras a grandes alturas. Los drones cada vez son más económicos y accesibles, permitiendo su aplicación en distintas áreas, por ejemplo: transporte, mensajería, topología, agricultura, vigilancia, captura de información geoespacial.

En interiores la señal del GNSS es intermitente o nula, por ello, los drones en interiores no se pueden valer de este sistema para su navegación; los drones pueden realizar vuelos autónomos o semiautónomos en interiores con los sensores adecuados o en los laboratorios especializados que están equipados con cámaras o antenas de emisión de ondas de radio para una correcta navegación del drón. Aplicando tecnologías de posicionamiento basado en marcadores visuales para la navegación de drones en interiores se evita el uso de equipos costosos o laboratorios especializados para vuelos de drones en interiores. Así se podrían generar nuevas actividades para los drones en interiores que ayudaría a tareas repetitivas o peligrosas para el humano, algunas de estas nuevas actividades pueden ser: inspección con histórico de fotografías, detección de mantenimiento preventivo en lugares confinados, inventario de estanterías altas, etc.

1.2 PLANTEAMIENTO DEL PROBLEMA

Los drones no pueden navegar de forma autónoma o semiautónoma en interiores si no se encuentran en un laboratorio con sistema de antenas emisoras de señales de radio frecuencia o sistema de cámaras para detección de movimiento. La necesidad de infraestructura para vuelos en interiores no dota al drón de ser un ente independiente y le resta autonomía. Para este problema, en esta tesis se propone un sistema para la navegación en interiores, utilizando un drón con sensores para vuelo en interior y un sistema de visión para la aplicación de tecnologías de marcadores visuales. El sistema de visión detecta las AprilTags, las cuales son colocadas en el campo de vuelo, asignando acciones o movimientos para la navegación del drón.

El sistema de visión planteado efectuará la detección e identificación de la AprilTags para desplazarse en el campo de vuelo para completar una trayectoria.

1.3 JUSTIFICACIÓN

Esta investigación se explora una alternativa económica para la navegación de drones en interiores, para prescindir de la necesidad de la infraestructura costosa de los laboratorios.

Se propone la navegación del drón con la detección de las AprilTag con el sistema de visión. El algoritmo asigna una instrucción o acción a la aeronave, según el AprilTag detectado para recorrer un trayecto.

Todos los sistemas dinámicos que cuenten con un sistema de visión que requieran orientación, navegación y control en interiores se pueden verse fácilmente beneficiados de este sistema para el desplazamiento por medio de la detección de las AprilTags.

1.4 HIPÓTESIS

Utilizando el sistema de visión del drón se puede crear un efectivo algoritmo de asistencia del desplazamiento en interiores, el cual mediante el procesamiento de imágenes detecte etiquetas AprilTag y controle los movimientos del drón para completar satisfactoriamente trayectorias de vuelo con un error relativo menor al 30%.

1.5 OBJETIVOS

1.5.1 Objetivo general

Diseñar un algoritmo para realizar la asistencia de navegación en el desplazamiento de un drón para completar una trayectoria, en un vuelo semiautónomo en interiores, mediante la detección de las AprilTags por medio del sistema de visión del drón.

1.5.2 Objetivos específicos

1. Determinar la iluminación el campo de vuelo para tener una correcta detección de la altura del drón con respecto al suelo.
2. Conocer por medio de experimentación la distancia de colocación y ángulo de visión en donde los AprilTags son detectados correctamente por el sistema de visión del drón.
3. Conocer las distancias desplazadas del drón después de la detección de los AprilTags con movimientos asignados.
4. Diseñar un algoritmo que realice la acción o movimiento según el AprilTags detectado.
5. Realizar pruebas del sistema en dos escenarios con diferentes AprilTags colocadas en el campo de vuelo para completar una trayectoria.

1.6 METODOLOGÍA

Se utiliza como base el lenguaje Python para el desarrollo del algoritmo de asistencia de navegación en el desplazamiento de un drón; se usa código libre y paquetes de Python y código libre para el drón Tello.

El funcionamiento secuencial cíclico del algoritmo se basa en los siguientes puntos:

1. Captura de imágenes por medio del sistema de visión del drón.
2. Transmisión de imágenes a la laptop.
3. Detección de AprilTags en las imágenes transmitidas.
4. Respuesta a la transmisión con la acción o movimiento a realizar por el drón.
5. Realización de la acción o el movimiento en el drón.

Para evaluar la efectividad del algoritmo se realizarán pruebas en 2 escenarios en interior con condiciones de luz controladas, sin viento y sin obstáculos en el campo de vuelo.

Se evaluará la repetibilidad del algoritmo midiendo el trayecto, desde el punto de despegue y el punto de aterrizaje.

1.7 ALCANCES Y LIMITACIONES

El algoritmo implementado en el drón Tello se acota a los siguientes señalamientos:

- El algoritmo desarrollado es un sistema de asistencia a la navegación, el cual asigna a la aeronave una instrucción o acción de desplazamientos al detectar una AprilTag a la vez.
- No es un sistema de navegación completo, no cuenta con posicionamiento relativo ni absoluto ni planificación de trayectorias ni evasión de obstáculos.
- El campo de vuelo debe estar iluminado para que el vuelo se realice de forma adecuada como lo advierte el manual de usuario del Tello.
- El vuelo debe de realizarse en interiores, sin viento y sin obstáculos.

- El AprilTag Tag_ID 0 tiene asignada la acción el aterrizaje, este AprilTag se muestra de forma manual.
- La orientación del drón no es variable, durante la ejecución del algoritmo mantiene su orientación desde el despegue hasta el aterrizaje.

CAPÍTULO 2 MARCO TEÓRICO

En este capítulo se presentan generalidades de los drones, como los elementos principales para operación, se exploran algunas de las clasificaciones de los drones y los sensores necesarios para su control en vuelo.

También, se presentan las especificaciones del drón utilizado en este trabajo y se explora el funcionamiento de los marcadores visuales AprilTags y su comparación contra otros marcadores.

2.1 EVOLUCIÓN DE LA NAVEGACIÓN

El hombre fue mejorando sus técnicas de navegación descubriendo su entorno y creando nuevas técnicas con los recursos a su alcance. La navegación aérea en un inicio se basó en la utilización de un reloj y brújula. Este tipo de navegación llevó el nombre de navegación estima; el piloto debía estimar con ayuda del reloj el tiempo de vuelo y con la brújula, la ubicación y dirección de la aeronave. Este tipo de navegación conlleva varios tipos de errores tanto instrumentales como humanos, factores externos como la intensidad del viento y su dirección afectaban la navegación [1]. Los errores son inherentes a todo tipo de navegación, aunque se han reducido sustancialmente con el tiempo.

En la radio navegación aérea es indispensable saber la ubicación del origen y orientación en el trayecto de dónde se encuentra la aeronave en el vuelo, para corregir el rumbo y tener una ruta óptima. Esto se logra a través de estaciones de radio transmisoras de referencia fijas colocadas en la superficie terrestre [2]. Los instrumentos y receptores a bordo del vehículo detectan las señales emitidas y calculan su posición relativa respecto de las estaciones emisoras de referencia. El desplazamiento en la frecuencia de la portadora de la señal debido al efecto Doppler permite medir su velocidad radial respecto de las estaciones emisoras [1].

La navegación inercial se basa en dos tipos de dispositivos, los sensores acelerómetro y giroscopio creando un sistema de navegación inercial (INS, por sus siglas en inglés *Inertial Navigation System*) que dan información precisa a corto plazo, con el inconveniente que después de un periodo largo de medición tienden a acumular errores, por ello se requiere de tecnologías adicionales como filtros para mejorar su precisión [3].

Los sistemas de navegación autónomos comenzaron su desarrollo en la década de los años 70s, sin depender de ayuda terrestre para poder determinar la posición de la aeronave [4]. Para el transbordador Apolo se desarrolló el *Apollo Guidance Computer* (AGC, por sus siglas en inglés) que ayudó en la navegación de las misiones Apolo para llevar al hombre a la Luna [5]. La unidad inercial de medición (IMU, por sus siglas en inglés *Inertial Measuring Unit*) mide la variación de rotación (mediante el uso de giróscopos) y la aceleración (usando acelerómetros) para proporcionar información acerca de la posición y de la orientación de la aeronave [1]. Este sistema se utilizó para realizar el viaje y orbitar en el espacio sin necesidad de radio-ayudas [1].

A pesar de los avances tecnológicos respecto a los sistemas de posicionamiento global (GPS, por sus siglas en inglés *Global Positioning System*), éstos presentan dos desventajas: bloqueos de la señal debido a obstrucciones y baja potencia de las señales. La combinación del GPS con INS proporciona una solución ideal, que aborda la debilidad del GPS y también el error INS que asciende con el tiempo cuando funciona por sí solo. El sistema GPS/INS integrado puede proporcionar una solución de posición y velocidad incluso durante una interrupción del GPS [2].

2.2 ELEMENTOS Y SENSORES DE UN DRÓN

Los drones en el año 2010 se popularizaron como un producto comercial, durante décadas anteriores eran objetos costosos y de grandes dimensiones para investigaciones exclusivas de uso militar [6]. El acontecimiento que hizo posible a los drones como producto comercial, fue en el año 2006 cuando la Agencia Federal de Aviación de Estados Unidos (FAA, por sus siglas en inglés *Federal Aviation Administration*) emitió el primer permiso para el uso de drones de forma comercial.

Un drón se compone de un controlador de vuelo (FC, por sus siglas en inglés *Flight Controller*) y la unidad inercial de medición (IMU); ellos en conjunto con otros sensores son el cerebro del drón, gobiernan los movimientos, midiendo la variación de rotación (giroscopios), la aceleración en cada uno de los tres ejes de movimiento X, Y y Z (acelerómetros) y proporcionan la información acerca de la posición y la orientación de la aeronave con respecto de los polos magnéticos del planeta (magnetómetros) [7].

También, se debe de contar con los elementos del radio control (RC) para el control del drón a distancia, el sensor de barómetro para calcular la altura o el módulo GPS que proporciona la latitud, longitud y altura. Un sistema alternativo para la posición del cuadricóptero, cuando no se cuenta con barómetro o GPS, es una cámara de flujo óptico (en inglés *optical flow*) la cual, apuntando hacia abajo, toma secuencia de fotografías del suelo para saber la posición. Según la configuración, el drón puede llevar motores con escobillas o sin escobillas (*brushless*); para estos últimos se necesita integrar un controlador electrónico de velocidad (ESC, por sus siglas en inglés *Electronic Speed Control*) [8]. Todos estos elementos son alimentados por una batería normalmente de celdas de polímero de litio (LiPo) de una o varias celdas según las aplicaciones y tamaño del cuadricóptero. En la Figura 1 se presenta una configuración habitual de un drón cuadricóptero.

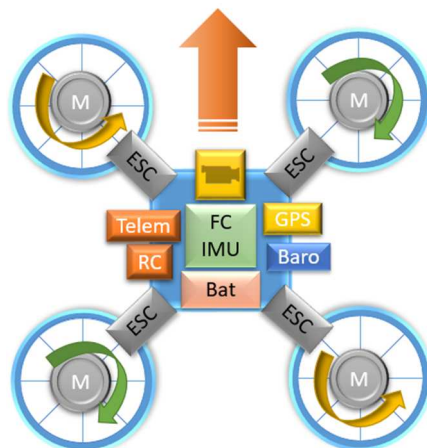


Figura 1 Componentes básicos de un drón

Fuente: elaboración propia.

En la Figura 2 se muestran los elementos de un drón por zonas, siendo las zonas de potencia y control las esenciales con la batería; en la figura se muestran también las interacciones entre los elementos del drón.

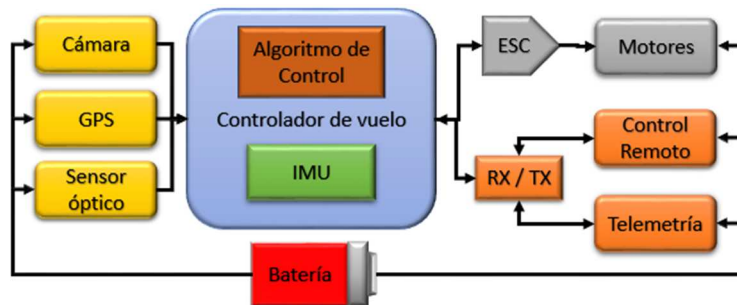


Figura 2 Elementos de un drón por zonas [potencia, transmisión, sensores y controlador]

Fuente: elaboración propia.

2.3 NAVEGACIÓN

La navegación es el conjunto de técnicas utilizadas para determinar la posición y el rumbo de un cuerpo en desplazamiento con respecto de un plano de referencia conocido por medio de la geometría o las señales de radio, además permite planificar una trayectoria de un lugar a otro para evitar y prevenir obstáculos [5].

2.4 NAVEGACIÓN INERCIAL

La inercia es la propiedad de los cuerpos para mantener una velocidad de rotación y de traslación constante, a menos que sean perturbados por fuerzas o pares, respectivamente (la primera ley de movimiento de Newton) [2].

La navegación inercial se basa en conocer la posición inicial, velocidad y actitud, y la medición de los índices de actitud y aceleraciones. La actitud es la orientación de la aeronave respecto al horizonte. La operación de los sistemas de navegación inercial (INS) depende de las leyes de Newton de la mecánica clásica. Es la única forma de navegación que no se basa en referencias externas. Un marco de referencia inercial es

un marco de coordenadas en el que las leyes de movimiento de Newton son válidas. Los marcos de referencia inercial no giran ni aceleran. Los sensores inerciales miden la *velocidad de rotación* y la *aceleración* [4]. A continuación, se describen los sensores utilizados principalmente en los sistemas de navegación inercial.

2.4.1 Giroscopio

Un giróscopo mantiene su eje de rotación (en el espacio inercial) frente a perturbaciones. Este efecto se conoce como rigidez giroscópica. Dichas perturbaciones generan un movimiento de precesión y nutación que se puede medir. Por ejemplo, al forzar la rotación de un giróscopo en un eje distinto a su eje de giro se produce un efecto que permite estimar la velocidad de rotación. Por tanto, los giróscopos tienen un eje en torno al cual giran permanentemente, otro eje en el cual se detectan perturbaciones y otro eje en el cual se miden dichas perturbaciones [5].

Los giroscopios de velocidad miden la *velocidad de rotación* y los giroscopios integradores (también llamados giroscopios de ángulo completo) miden el *ángulo de rotación* [5].

2.4.1 Acelerómetro

Los acelerómetros son sensores para medir la *aceleración*. Sin embargo, los acelerómetros no pueden medir la aceleración gravitacional. Es decir, un acelerómetro en caída libre (o en órbita) no tiene entrada detectable [1].

La configuración básica de un acelerómetro consiste en una masa testigo contrabalanceda por un elemento elástico fijo a la carcasa del instrumento (fija al vehículo en el caso *strapdown* o a la plataforma inercial estabilizada), como se muestra en la Figura 3.

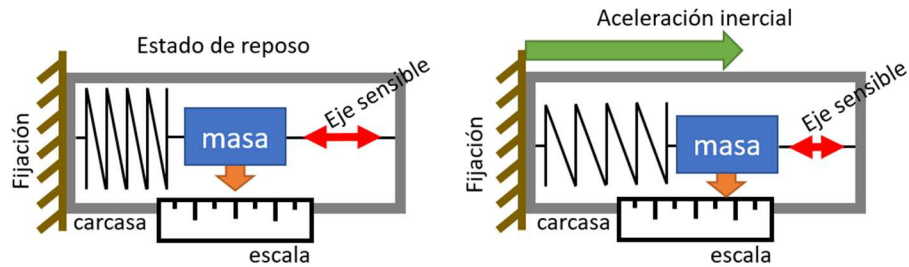


Figura 3 Principio de funcionamiento de un acelerómetro

Fuente: elaboración propia.

De este modo, de acuerdo con la 2ª ley de Newton, una fuerza inercial (no gravitatoria) actuante sobre la carcasa produce un desplazamiento de la masa testigo respecto de la primera. Este desplazamiento es transformado en una señal medible [1].

El eje de entrada de un sensor de inercia define qué componente de vector mide. Los sensores multi-eje miden más de un componente.

Una unidad de medida inercial (IMU) o unidad de referencia inercial (IRU) contiene un conjunto de sensores: acelerómetros (tres o más, pero generalmente tres) y giroscopios (tres o más, pero generalmente tres). Estos sensores están montados rígidamente en una base común para mantener la misma orientación relativa.

Hay muchos diseños diferentes de INS con diferente rendimiento y características, pero generalmente se dividen en dos categorías: técnicas de plataforma estabilizada (gimbaled) y fijas al sistema (strapdown), como se muestran en las Figuras 4 y 5.

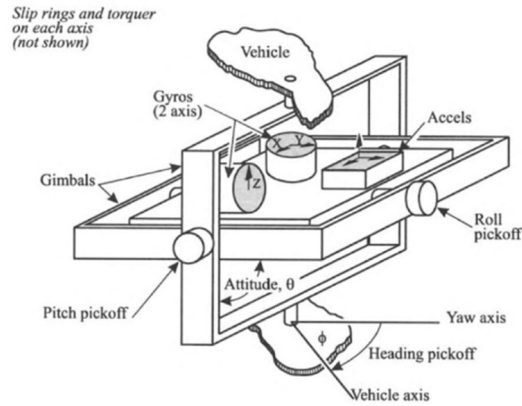


Figura 4 INU plataforma estabilizada (gimbaled) [10]

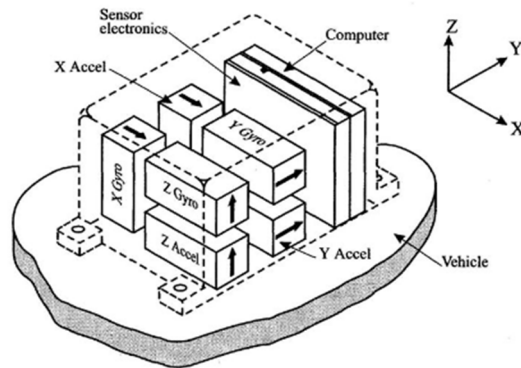


Figura 5 IMU plataforma fija (strapdown) [10]

2.5 SISTEMAS DE POSICIONAMIENTO

Para que un sistema pueda moverse en un espacio necesita adquirir información del entorno por medio de sensores para poder localizarse, logrando navegar y moverse por superficies adecuadas sin colisiones, evitando zonas u obstáculos para llegar a un destino. Según la necesidad se puede usar sistemas de posicionamiento para exteriores o para interiores. En la localización en exteriores se suele hacer uso de GPS, odometría mecánica, etc., en interiores es posible conseguir resultados con sensores ópticos como cámaras, odometría laser, etc [11].

2.5.1 Sistemas de posicionamientos para exteriores

Para aplicaciones en entornos en el exterior generalmente se utiliza el sistema de navegación global por satélite GNSS, basado en una constelación de satélites para conocer la posición en la que se encuentra el objeto; esa posición global unida a un mapa permite al sistema conocer su localización con respecto al planeta [12].

2.5.2 Sistemas de posicionamiento para interiores

Un sistema de posicionamiento en interiores IPS (en inglés *indoor positioning system*) es una red de dispositivos utilizados para localizar inalámbricamente objetos o personas dentro de un edificio [13]. Un IPS se basa en anclajes próximos (nodos con una posición conocida), que localizan activamente etiquetas o proporcionan contexto ambiental a los dispositivos sensores. A continuación, se presentan las tecnologías en las que pueden estar basados estos sistemas.

2.5.2.1 Tecnologías basadas en ondas de radio

Las tecnologías basadas en ondas de radio originalmente consistían en una serie de transmisores de radio instalados en posiciones fijas y bien conocidas. De este modo, se puede usar un sistema de triangulación de señal para determinar la distancia del drón con respecto a tres o más balizas y, por lo tanto, con respecto a cualquier punto conocido en el mapa [14]. Hoy en día se usan también balizas ópticas que son captadas por cámaras e identificadas para inferir la posición. También han evolucionado los tipos de balizas de radio, adaptándose a nuevas tecnologías como Wi-Fi, Bluetooth, Zigbee, RFID, radio de banda ultra ancha (UWB), etc [6].

En la Figura 6 se muestra un laboratorio equipado con antenas y equipo para usar la tecnología UWB; esta tecnología es capaz de ofrecer localización precisa en exterior e interior a distancias de menos de 10 centímetros [15].

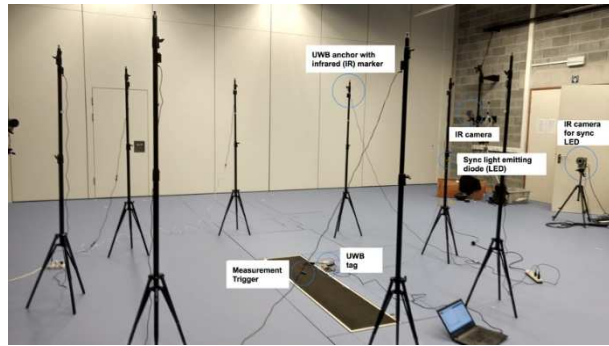


Figura 6 Laboratorio de UWB de Sport Science Laboratory-Jacques Rogge en la Universidad de Ghent [15]

2.5.2.2 Tecnologías basadas en sistemas visuales (cámaras estáticas)

La odometría visual es una técnica que permite inferir la posición del objeto en base a las variaciones en la imagen captada por una o más cámaras situadas en puntos estratégicos del objeto [6]. Un ejemplo de un sistema de posicionamiento en interiores por medio de sistemas visuales es el Testbed del CATEC (Centro Avanzado de Tecnologías Aeroespaciales) [16], el cual se muestra en la Figura 7. El sistema de posicionamiento es un sistema VICON que cuenta con 20 cámaras; este sistema es capaz de calcular en tiempo real la posición y la actitud de cualquier persona y objeto móvil que se encuentre en un volumen de 15 x 15 x 5 m, un sistema eficiente, pero costoso [17].



Figura 7 Banco de pruebas o Testbed en interiores [16].

2.5.2.3 Tecnologías basadas en mediciones inerciales

Los sensores de inercia de sistemas microelectromecánicos (*MEMS*, por sus siglas en inglés *micro-electro-mechanical systems*) sufren ruidos internos que resultan en un error de posición de crecimiento cúbico con el tiempo [18]. Para reducir el crecimiento de errores en tales dispositivos, a menudo se utiliza un enfoque basado en el filtrado de Kalman [18].

2.5.2.4 Tecnologías basadas en posicionamiento basado en marcadores visuales

Un sistema de posicionamiento visual puede determinar la ubicación de un dispositivo móvil con cámara mediante la decodificación de coordenadas de ubicación a partir de marcadores visuales [19]. En dicho sistema, los marcadores se colocan en ubicaciones específicas en un lugar; cada marcador codifica las coordenadas de esa ubicación: latitud, longitud y altura del piso. La medición del ángulo visual desde el dispositivo hasta el marcador permite al dispositivo estimar sus propias coordenadas de ubicación en referencia al marcador [20]. En la Figura 8 se presenta un ejemplo de uno de los usos de los marcadores visuales; en este caso el robot Atlas de la empresa Boston dynamics está identificando un objeto en el ambiente [21].



Figura 8 Ejemplo de posicionamiento basado en marcadores visuales [21].

2.6 CLASIFICACIÓN Y TIPOS DE DRONES

A continuación, se presenta la clasificación de drones por peso promedio de despegue, tipo de despegue, nivel de autonomía, estructura y número de motores.

2.6.1 Clasificación de drón por peso promedio de despegue (PPD)

Existen criterios para la clasificación de un vehículo aéreo no tripulado (UAV por sus siglas en inglés *Unmanned Aerial Vehicle*). En este documento tomaremos como base la clasificación por peso promedio de despegue (PPD) que es el peso que soporta un drón al momento del despegue el cual está limitado, por ejemplo, por sus límites estructurales. En la Tabla 1 se muestran los nombres que reciben de acuerdo con esta clasificación [22].

Tabla 1 Clasificación de los drones de acuerdo con el criterio peso promedio de despegue [22].

Número	Peso promedio de despegue	Nombre
0	Menos de 1 Kg	Micro
1	Hasta 1 Kg	Mini
2	Hasta 13.5 Kg	Pequeño
3	Hasta 242 Kg	Ligero / Ultraligero
4	Hasta 4 332 Kg	Normal
5	Más de 4 332 Kg	Grande

La clasificación de los vehículos aéreos no tripulados de acuerdo con el tipo de despegue y aterrizaje se muestra en la Figura 9. En este documento se utiliza un vehículo aéreo no tripulado de despegue y aterrizaje vertical con ala rotativa un multirrotores de cuatro motores.

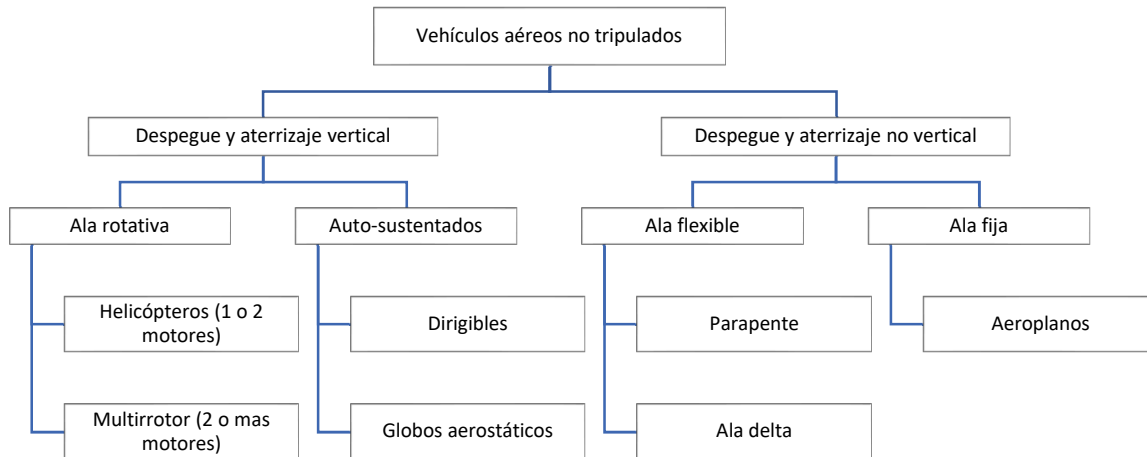


Figura 9 Clasificación de los vehículos aéreos no tripulados de acuerdo con el tipo de despegue y aterrizaje.

Fuente: elaboración propia.

2.6.2 Clasificación del drón por el nivel de autonomía

Otra forma en la que pueden clasificarse los drones es referente a su nivel de autonomía como se muestra en la Tabla 2. El nivel de autonomía es un aspecto importante en cualquier ampliación real [22].

Tabla 2 Clasificación de los drones según nivel de autonomía [23].

Nivel de autonomía	Nivel 0	Nivel 1	Nivel 2	Nivel 3	Nivel 4	Nivel 5
Interacción humana / sistema autónomo						
Grados de autonomía	Nula automatización	Poca automatización	Parcial automatización	Condicional automatización	Alta automatización	Completa automatización
Descripción	Control del dron 100% manual. operación de forma remota.	Piloto mantiene el control. El dron tiene control de alguna función principal. Realizar misiones preplaneada.	Piloto mantiene responsabilidad de un vuelo seguro. Dron puede mantener altura bajo ciertas condiciones. Respuesta robusta ante fallas.	Piloto actúa como sistema de respaldo. Dron puede realizar funciones autónoma bajo ciertas condiciones. Coordinación entre vehículos en tiempo real, vehículo adaptable ante fallas.	Piloto esta afuera de la operación de control. Dron tiene sistemas de respaldo si falla la plataforma continua en operación. Cooperación entre vehículos en tiempo real, conocimiento del espacio de operación	Vehículo completamente autónomo. Dron será capaz de usar IA, herramientas para planear vuelos con sistema de aprendizaje autónomo.
Evasión de obstáculos	Ninguna	Sensado y alertas		Sensado y evitar obstáculos	Sensado y navegación	

Los drones se controlan de forma remota o autónoma y pueden ser utilizados para realizar diferentes tipos de misiones y tareas. Los drones pueden ser controlados o manejados de diferentes formas [22]:

- **Manual.** Muy pocos equipos en la actualidad se manejan de forma manual. Quizá los equipos más importantes en la actualidad que se siguen pilotando de forma manual son los drones de carrera.
- **Semi automático.** Ésta es la forma en la que la mayoría de los pilotos de drones vuelan cuando inician. Éste es un modo en el que el operador del equipo tiene el control principal de las funciones del drón. Sin embargo, el drón está asistido por el sistema GPS, giroscopios y otros sistemas de seguridad del equipo.
- **Automático.** En este modo, el drón actúa de forma similar a un piloto automático. La labor del operador profesional del equipo es preprogramar una misión que el drón ejecutará y una vez la máquina se encuentre realizando la tarea, el piloto debe estar atento para asistir en caso de emergencia.
- **Autónomo.** Por ahora en muchos países la utilización de estos equipos de forma autónoma está prohibida y éste es un modo en el que el drón toma decisiones por sí sólo. Apoyándose en la tecnología de Inteligencia Artificial, el drón aprende y decide de acuerdo con su entorno. En este caso la presencia humana es accesoria y no es necesaria.

En esta tesis, el vuelo del drón se realiza de forma semiautomática; el algoritmo de asistencia de navegación en el desplazamiento de un drón reemplazará el control principal que realiza el piloto por medio de la detección de los AprilTags.

2.6.3 Clasificación del drón por el número de motores y por la forma de su estructura

Los vehículos aéreos multirrotores se pueden subdividir por el número de motores y por la forma de su estructura. Como se puede observar en la Figura 10, existen los tricópteros, cuadricópteros, hexacópteros u octocópteros. En este trabajo se utiliza un vehículo aéreo con cuatro multirrotores con estructura Quad-rotor X.

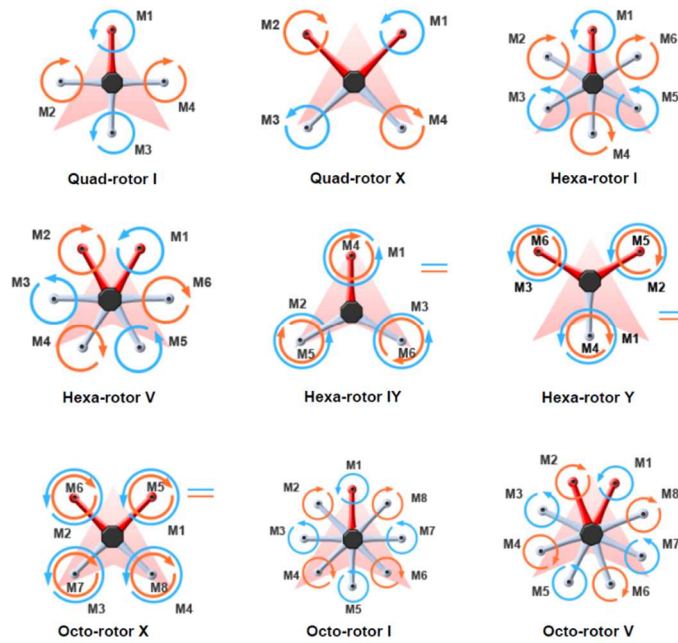


Figura 10 Estructura multirrotores [24].

2.7 CUADRICÓPTERO

Un cuadricóptero posee cuatro motores acoplados al chasis de la aeronave, que a su vez son acoplados a hélices con un ángulo de ataque, según el sentido del giro para generar fuerza de empuje, con una cierta distancia al centro de gravedad del drón. Debido a los cuatro motores (actuadores), el cuadricóptero se conoce como un sistema subestimado porque el cuadricóptero tiene seis grados de libertad (6-DOF), 3 por los desplazamientos en los ejes “x, y, z”, y otros 3 debido a los giros respecto a dichos ejes; el cuadricóptero no puede dar una aceleración en una dirección sin cambiar el comportamiento dinámico en otro grado de libertad [22]. Dichos grados de libertad condicionan los movimientos básicos, dos grados para permitir el desplazamiento horizontal (gírar = pitch (θ), rotar = roll (ϕ)), y uno para la orientación del cuadricóptero (guiñar = yaw (ψ) alrededor de los ejes x, y, z) [22].

2.7.1 Movimientos de un cuadricóptero

Al variar la velocidad de los motores, se pueden cambiar las fuerzas de empuje entregadas. El drón puede mantener el movimiento estacionario o puede volar en cierta dirección. En una configuración de cuadricóptero en X, el sentido de giro de las hélices es según se muestra en la Figura 11, con los motores 1 y 2 girando en sentido antihorario (CCW) y los motores 3 y 4 giran sentido horario (CW) enfrentados 2 a 2. La Figura 11 muestra el cuadricóptero con fuerzas de empuje (F_1 - F_4), los posibles movimientos giratorios ($\omega_1 - \omega_4$) y el sistema de ejes con orientaciones para el armazón del cuerpo del drón cuadricóptero y el armazón inercial con respecto a la tierra.

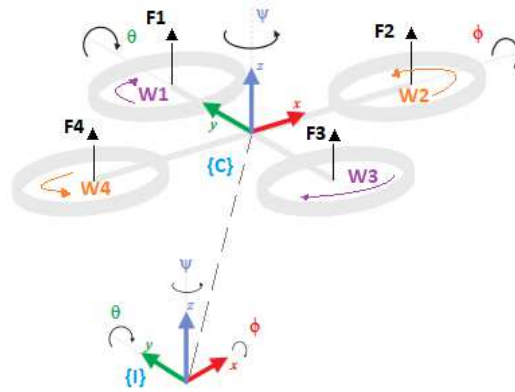


Figura 11 Modelo del cuadricóptero con marco de referencia inercial respecto a tierra $\{I\}$ y con marco de referencial inercial respecto al cuerpo del drón $\{C\}$

Fuente: elaboración propia.

Para volar un drón del tipo cuadricóptero, las velocidades del rotor deberían aumentar para subir, o mantener la velocidad del rotor nivelada para mantener el movimiento. Para los modos de vuelo del drón cuadricóptero es necesario cambiar las velocidades del rotor de los cuatro rotores. Cuando la velocidad del rotor del tercer y cuarto rotor aumenta, el cuadricóptero da un movimiento de inclinación hacia el suelo. Por lo tanto, las velocidades del primer y segundo rotor deben aumentar en velocidad para mantener un vuelo estable (produciendo un momento de contraataque).

El cuadricóptero puede rodar si aumenta la velocidad del rotor del segundo y tercer rotor. Esto hace que el cuadricóptero gire; para mantener el vuelo estable es necesario que el primer y cuarto rotor aumenten la velocidad para igualar la velocidad de los otros motores y tener velocidades similares para un comportamiento estático en el drón. Para el cambio en la dirección del drón cuadricóptero, las velocidades de los rotores segundo y cuarto deben aumentar para mover el fuselaje hacia la izquierda. Por lo tanto, si se necesita un vuelo recto, la velocidad del rotor del primer y del tercer rotor debe aumentarse (contrataque) [25].

2.7.2 Yaw o movimiento de guiñada

El cuadricóptero debe volar siempre en una posición horizontal para permanecer estable. Es por ello por lo que el único giro estable lo realiza respecto a su eje vertical (z) llamado guiñada, o yaw. Esto permite al cuadricóptero orientarse en la dirección correcta. Esto es posible aumentando la velocidad de giro de dos motores opuestos, frente a los restantes, como se muestra en la Figura 12.

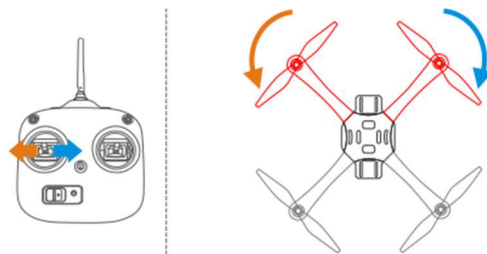


Figura 12 Movimiento de guiñada (yaw) [24].

2.7.3 Pitch o movimiento de cabeceo

El movimiento de cabeceo, o *pitch*, permite al cuadricóptero inclinarse hacia delante o hacia atrás. Esto permite a la aeronave avanzar y retroceder, de manera idéntica a un helicóptero. Para ello se aumenta la velocidad de giro de los motores traseros (para avanzar) o delanteros (retroceder), como se puede observar en la Figura 13.

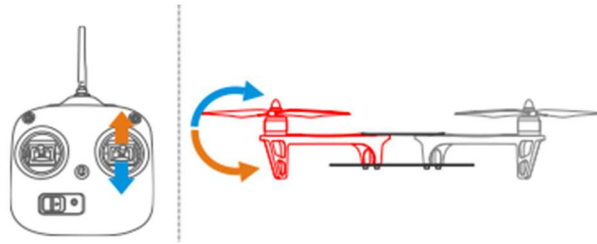


Figura 13 Movimiento de cabeceo (pitch) [24].

2.7.4 Roll o movimiento de alabeo

El alabeo (o *roll*) inclina el cuadricóptero gracias al aumento de potencia en dos motores consecutivos. Solo que en este caso se trata de los motores del lado izquierdo (para desplazarse hacia la derecha) o del lado derecho (para desplazarse hacia la izquierda), como se observa en la Figura 14 [8].

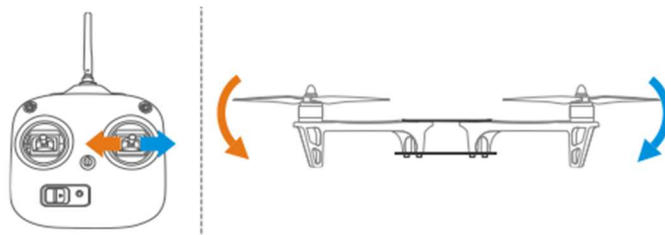


Figura 14 Movimiento de alabeo (roll) [24].

2.8 CUADRICÓPTERO TELLO

El drón Tello es un drón comercial de uso recreativo de la marca de Ryze Tech construido en colaboración con DJI, el cual se muestra en la Figura 15. Éste es un drón cuadricóptero liviano de 80 g de dimensiones: 98 × 92.5 × 41 mm.

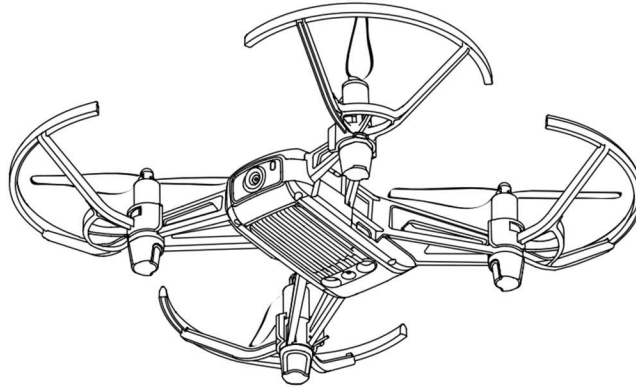


Figura 15 Drón Tello Ryze Tech [26].

El drón Tello puede ser programado con su SDK en Scratch. Tello cuenta con características de Auto Takeoff, Auto Landing y Low Battery protection; esto es posible gracias a la combinación de la unidad de procesamiento de visión (VPU por sus siglas en ingles Virtual Processing Unit) Movidius de Intel y la tecnología de estabilización de vuelo de DJI [26].

El sistema visual de posicionamiento (VPS, por sus siglas en ingles Vision Positioning System), presentado en la Figura 16, mantiene al drón en una posición estable y fija durante el vuelo, sin la necesidad de un GPS, mientras que el viento sea mínimo y el drón vuele relativamente cerca del suelo (0.3 a 6 m) [26]. Los sensores infrarrojos permiten un despegue y aterrizaje suave, éstos apuntan hacia abajo a los lados del sistema visual de posicionamiento.

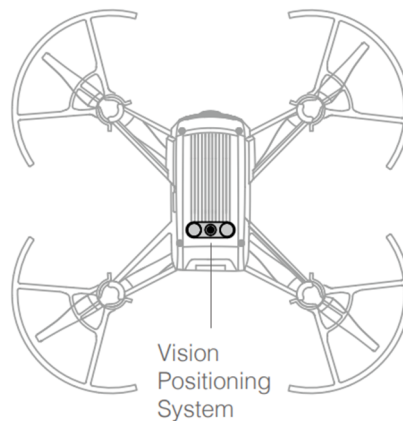


Figura 16 Sistema visual de posicionamiento [26].

El sistema visual de posicionamiento funciona correctamente con una iluminación mayor a 300 lux y menor a 10,000 lux [26]. En la parte frontal, el Tello cuenta con una cámara que puede retransmitir vídeo a 720p a 30fps y captura fotografías de 5 megapíxeles (2592x1936) con estabilización electrónica que captura imágenes claras a pesar de estar volando; tiene un campo de visión de 82.6° (FOV), sin deformación en las líneas rectas, con apertura f/2.2 y focal 1.98 mm; transmite video HD en tiempo real por medio de su Wi-Fi 2.4 GHz 802.11n; puede lograr una distancia máxima de vuelo de 100m con una velocidad máxima de 10m/s; y cuenta con una batería desmontable de ion-litio 1100 mAh/ 3.8 V que le permite un tiempo de vuelo de 13 min [26].

Tello cuenta con una aplicación para controlar el drón desde el celular; éstos se comunican por medio de la conexión Wi-Fi que el drón genera. La aplicación puede mostrar algunos letreros para notificación del funcionamiento del drón.

A continuación, en la Figura 17, se muestran algunos de los mensajes que muestra la aplicación de advertencia como de peligro del control del drón o de su funcionamiento.

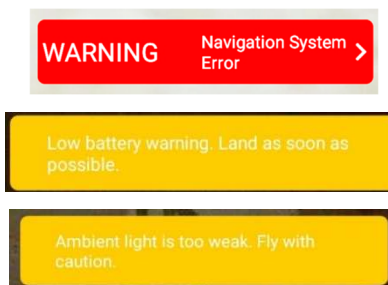


Figura 17 Mensaje alerta y advertencia de la aplicación Tello [27].

2.9 MARCADORES VISUALES

Los marcadores visuales son códigos bidimensionales o simplemente etiquetas (tags) que funcionan como un método de representar información en forma visual legible por las cámaras digitales. Los marcadores visuales cuentan con un patrón consistente, se colocan en entornos físicos para varias aplicaciones como proporcionar seguimiento, alineación, identificación de objetos y localización [28].

Los marcadores visuales son populares y útiles en múltiples campos como el de la robótica, realidad aumentada y otros. Los marcadores son aplicados para calibrar cámaras, aplicaciones industriales, interacción social humano-robot y humanoides, SLAM, robótica de rescate, colaboración de robots, control de enjambres, entre otros.

El diseño de la etiqueta depende directamente de su aplicación. Por ejemplo, en la Figura 18 se muestran diferentes formas de marcadores visuales; una cuadrícula 2D de cuadrados blancos y negros, un conjunto de líneas espaciadas, matrices de puntos o circunferencias con segmentos blancos y negros pueden comunicar información digital [19].

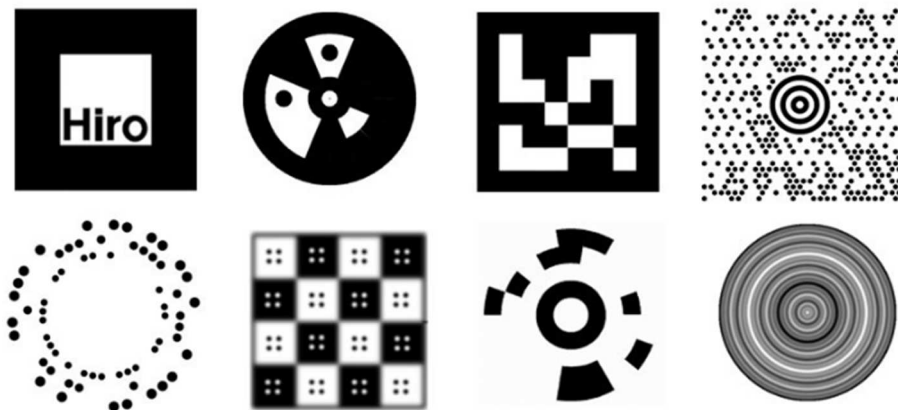


Figura 18 Ejemplo de marcadores visuales de izquierda a derecha primera fila: ARToolkit, Circular Data Matriz, ARToolkit Plus y Maxicode, segunda fila de izquierda a derecha: RuneTag, BlurTag, CanTag y Fourier Tag [29].

La mayoría de los marcadores se utilizan en aplicaciones de posicionamiento o realidad aumentada para almacenar información. Los marcadores cuentan con al menos cuatro puntos de características que ayudan a determinar la posición y la orientación a las cámaras en un espacio tridimensional [30]. A continuación, en la Tabla 3, se dan ejemplos de los distintos marcadores visuales, así como usos y características.

Tabla 3 Atributos principales marcadores visuales.

<p>Data Matrix</p> 	<p>Se utiliza para codificar gran cantidad de caracteres de datos; ideal para marcar envases muy pequeños. Este marcador es utilizado por la industria farmacéutica para dosis unitarias y marcado de productos; fue desarrollado originalmente para el Programa del Transbordador Espacial, donde se deben rastrear millones de piezas. Es un código matricial bidimensional que utiliza una representación visual del código binario (0 o 1) y cuenta con corrección de error y 1556 ASCII (8 bits), 2335 caracteres alfanuméricos o 3116 caracteres numéricos [31].</p>
<p>MaxiCode</p> 	<p>Fue desarrollado y utilizado por UPS para clasificar y abordar paquetes en todo el mundo. Este marcador está formado por filas desplazadas de módulos hexagonales dispuestos alrededor de un patrón de buscador único; cuenta con corrección de error y 93 ASCII [31].</p>
<p>QR Code</p> 	<p>Fue desarrollado en 1994 por DENSO para la industria automotriz en Japón. Este marcador se utiliza comúnmente en aplicaciones móviles y codifica gran cantidad de caracteres de datos; admite también caracteres Kana, binarios y Kanji. Cuenta con corrección de error y 4296 ASCII, 7089 caracteres alfanuméricos. [31].</p>
<p>Aztec Code</p> 	<p>Se usa en una variedad de aplicaciones de seguridad de pacientes, incluidas pulseras de identificación y etiquetas para medicamentos; puede codificar desde pequeñas hasta grandes cantidades de datos, con porcentajes de corrección de errores seleccionados por el usuario. El tamaño del símbolo se ajusta automáticamente según la cantidad de datos ingresados [31].</p>
<p>ARTag</p> 	<p>Utiliza un procesamiento de datos complejo para aumentar la confiabilidad y minimizar el efecto de las condiciones de iluminación. Se puede utilizar para facilitar la aparición de objetos virtuales, juegos y animaciones en el mundo real; permite capacidades de seguimiento de video que calcula la posición y orientación de una cámara en relación con los marcadores físicos en tiempo real [29].</p>
<p>Aruco</p> 	<p>Es fácilmente configurable. OpenCV tiene un total de 25 diccionarios predefinidos de marcadores. Todos los marcadores en un diccionario contienen el mismo número de bloques o bits (4×4, 5×5, 6×6 o 7×7), y cada diccionario contiene un número fijo de marcadores (50, 100, 250 o 1000) [31].</p>
<p>AprilTag</p> 	<p>Usado por la NASA. Estos marcadores son detectados a largas distancias, requieren pocos parámetros de configuración y requieren poco poder de cómputo necesario para su detección y decodificación [30].</p>

2.9.1 AprilTags

Es un marcador visual creado por los Laboratorios de April Robotics de la Universidad de Michigan en 2011 [19]. Los AprilTags pueden almacenar información en una matriz de puntos entre 4 y 12 bits. Esto hace que sean detectadas con mayor robustez y precisión [19]. Estos marcadores son usados en una gran variedad de tareas como realidad aumentada, robótica, o para la calibración de cámaras. Las etiquetas se imprimen desde archivos PDF o PNG [30] en una hoja blanca lo que las hace económicas; se debe tener cuidado de no hacer ninguna interpolación, de modo que se conserven los bordes afilados. Las AprilTags son detectadas por software de detección computacional con gran precisión de la posición en tres dimensiones, así como su orientación y se puede identificar la posición relativa de la cámara [20]. Las librerías de las AprilTags están implementadas en el lenguaje de programación C, sin dependencias externas [19]. Estos marcadores fueron diseñados para incluirlos en otras aplicaciones, son fácilmente portables en dispositivos embebidos y pueden detectarse en tiempo real [29]. Existen varias familias de AprilTag, que son diferenciadas por dos parámetros: el número de bits codificado y la distancia de Hamming mínima de diferencia entre cada código; por ejemplo, el nombre 36h11 indica que estas son etiquetas de 36 bits (matriz 6x6) con una distancia de Hamming mínima de 11 bits entre códigos, son 518 etiquetas de esta familia; la familia 16h5 se refiere a 16 bits (matriz de 4x4) con una distancia de Hamming mínima de 5 bits entre códigos como se muestra en la Tabla 4. El sistema en comparación con otros sistemas se caracteriza por un mayor número de códigos diferentes, un gran número de bits para detectar errores y poder corregirse lo que resulta en un menor número de falsos positivos, evita la confusión entre etiquetas y tiene diseño robusto para identificación del tag a gran distancia [19].

Tabla 4 Familia AprilTag ID etiquetas [19].

Familia AprilTag	ID etiquetas
16h5	30
25h7	242
25h9	35
36h10	2320
36h11	587

La estructura de un AprilTag está rodeada de un marco blanco y un marco negro; después contiene la matriz de codificación, según la familia del tag. La estructura de un AprilTag es llamada módulos de píxeles ya que debe respetar las dimensiones de un píxel, como se muestra en la Figura 19.

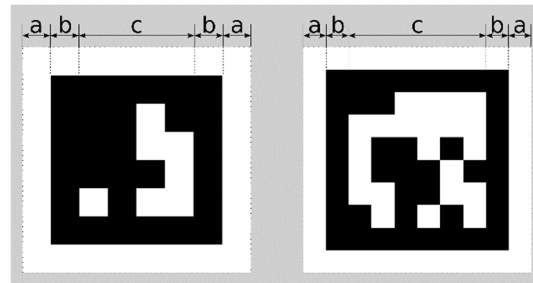


Figura 19 Contorno módulos píxel [20].

En esta tesis se utilizan las etiquetas de la familia AprilTag 36h11 debido a su versatilidad de ser detectadas a distancias mayores de 2 metros.

2.9.1.1 Codificación y decodificación de AprilTag 36h11 Tag 0

La generación de una etiqueta AprilTag parte de un número mágico o semilla preestablecida en hexadecimal, según sea la familia es la longitud del número hexadecimal; para la familia 16h5 con una matriz de 4x4, es un hexadecimal de 4 dígitos; para la familia 25h7 con matriz de 5x5 es un hexadecimal de 7 dígitos; y para la familia 36h11 que utiliza un hexadecimal de 9 dígitos, la matriz es de 6x6.

Para mostrar la codificación y descodificación se usa como ejemplo un AprilTag de la familia 36h11.

Primero usando la semilla 0xd5d628584 y convirtiendo los dígitos hexadecimales de derecha a izquierda en números binarios de 4 bits se obtiene una cifra de 36 dígitos, como se observa en la Figura 20 (a). Si este dígito lo colocamos de izquierda a derecha y de arriba hacia abajo, en la matriz de 6x6 se genera el marcador visual del AprilTag como se observa en la Figura 20 (b).

El paso siguiente sería llenar la matriz de los dígitos binarios, agregando el marco blanco exterior y otro marco negro cubriendo la matriz de datos, como se muestra en la Figura 20 (c). Cambiando los valores ceros por negro y unos por blanco queda como resultado el AprilTag 36h11 con Tag_ID 0 como se presenta en la Figura 20 (d) [28].

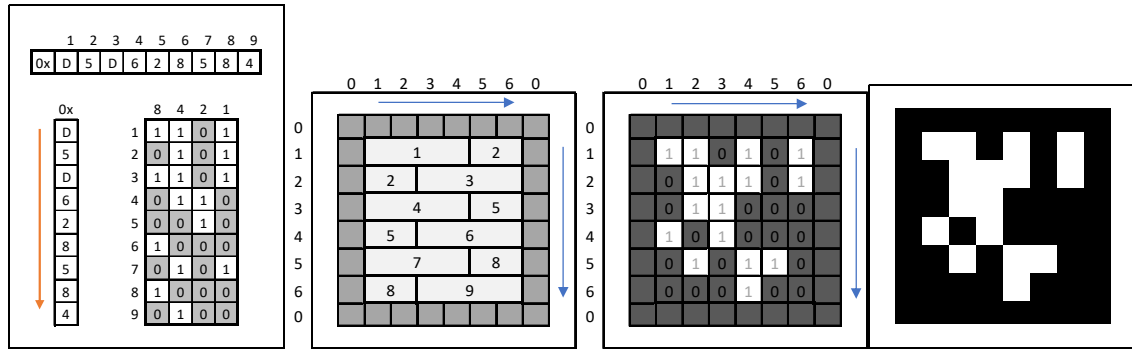


Figura 20 Generación de AprilTag 36h11 a) Descomposición número hexadecimal en número binario 8bits, b) Matriz de llenado de hexadecimal a binario, c) Colocación de 9 dígitos hexadecimal a binario en matriz, d) AprilTag 36h11 ID 0.

Fuente: elaboración propia.

En la Figura 21 se muestra nueve tags de diferentes familias de AprilTag; se puede observar lo diferente que son entre ellas.

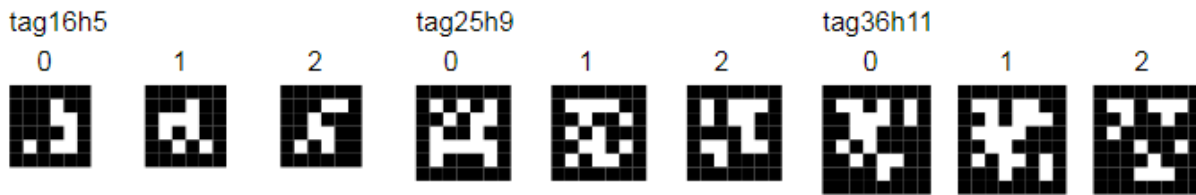


Figura 21 Familias de AprilTags Tag16h5, Tag25h9 y Tag36h11 [19].

2.9.1.2 Estimación de pose de AprilTag

Para cada etiqueta detectada, se estima la pose de esta etiqueta en el marco de coordenadas de la cámara. Un requerimiento para la estimación de pose es que una etiqueta sea completamente visible en el campo visual de la cámara [20]. El marco de coordenadas de la etiqueta debe estar alineado, como se muestra en la Figura 22.

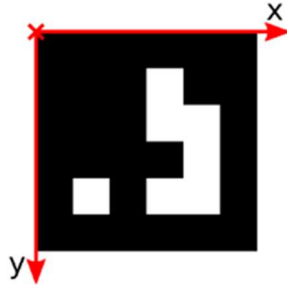


Figura 22 Marco de coordenadas de AprilTag [19].

El eje z apunta hacia la etiqueta. El origen del sistema de coordenadas se coloca exactamente en la transición del borde blanco al negro. Como las AprilTags no tiene una orientación obvia, el origen se define como la esquina superior izquierda en la orientación en la que se generan previamente. Durante la estimación de pose, también se estima el tamaño de la etiqueta, mientras se supone que la etiqueta es cuadrada. Para las AprilTags, el tamaño cubre solo la parte negra de la etiqueta, ignorando el borde blanco más externo. Cada etiqueta tiene una identificación; para AprilTags es la identificación de la familia más la el ID de la etiqueta [32].

2.9.2 Funcionamiento del algoritmo de las AprilTag

El mecanismo de detección de AprilTag se resume en nueve pasos, los cuales se describen a continuación.

2.9.2.1.1 Escala de grises y desenfoque

La detección de AprilTag comienza convirtiendo la imagen en escala de grises de punto flotante y aplicando un desenfoque gaussiano. La escala de grises de punto flotante es una escala de grises que toma todos los valores de píxeles y los establece en un valor entre 0.0 y 1.0 [20].

Esto se puede hacer de varias maneras dependiendo de la aplicación. La aplicación del desenfoque gaussiano se realiza mediante una distribución gaussiana. Cada valor es el peso que el píxel dará al valor final del píxel central. Esto se hace para suavizar la

imagen, lo que ayuda a eliminar el ruido cuando se calcula el gradiente local, como se muestra en la Figura 23 [7].

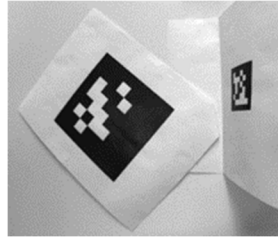


Figura 23 Imagen de referencia para procesar [19].

2.9.2.1.2 Gradiente local

El gradiente local es un cálculo de la dirección o gradiente en cada píxel. Esto se hace tomando las matrices presentadas en la Ecuación 1 y multiplicándolas por los valores de píxeles. El centro de la matriz es el píxel que se calcula. Esto se conoce como la derivada de la imagen [20].

$$L_z = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} L \text{ y } L_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} L.$$

Ecuación 1 Matriz cálculo de gradiente (kernels para el Filtro Sobel) [20]

La magnitud final del gradiente se calcula luego mediante la Ecuación 2 y la dirección se calcula mediante la Ecuación 3.

$$|\nabla L| = \sqrt{L_z^2 + L_y^2}$$

Ecuación 2 Magnitud del gradiente = M [20]

$$\theta = aTan2(L_y, L_z)$$

Ecuación 3 Dirección del gradiente = D [20]

Esto le da a cada píxel una magnitud y dirección de cambio. En la Figura 24 se muestra un ejemplo de magnitud a la izquierda y dirección a la derecha [7].

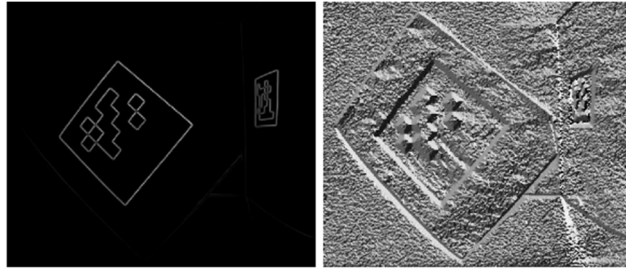


Figura 24 (izquierda) Magnitud de gradiente, (derecha) Dirección del gradiente. [19]

2.9.2.1.3 Detección de bordes

Para prepararse para la detección de bordes, los píxeles se filtran por un umbral de magnitud. Se utilizó un valor umbral de 0,8 [20]. Si la magnitud del píxel no supera el umbral, se ignora y se establece en un valor de 0. Los píxeles que quedan se agrupan por dirección, lo que da las áreas de mayor cambio en la imagen. Esto deja un conjunto de esquemas para cada cambio que alcanzó el valor umbral. Genera una lista de bordes, agrupando píxeles conectados con direcciones similares. Un borde está presente si la magnitud del gradiente para ambos píxeles está significativamente por encima de cero [7].

2.9.2.1.4 Ponderado de bordes

Una vez que se han detectado estas áreas de cambio, deben refinarse en bordes de un solo píxel. Esto se hace mediante la creación de grupos de píxeles. Los grupos se encuentran agrupando píxeles de la misma dirección a lo largo del borde del gradiente. Luego, cada grupo recibe un peso (color) basado en su gradiente [19].

Esto proporciona un conjunto definido de bordes para el marco. Un ejemplo de esto se muestra en la Figura 25.

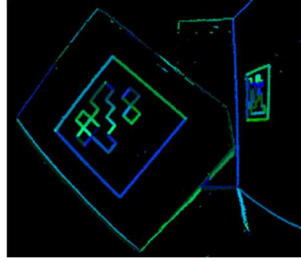


Figura 25 Ponderado de bordes [19].

2.9.2.1.5 Segmentos de línea

Los grupos se agrupan para formar segmentos de línea calculando si debieran agruparse. Esto se hace usando las fórmulas que se muestran en la ecuación 4 y 5 [20].

$$D(n \cup m) \leq \min(D(n), D(m)) + \frac{K_D}{|n \cup m|}$$

Ecuación 4 Dirección del gradiente.

$$M(n \cup m) \leq \min(M(n), M(m)) + \frac{K_M}{|n \cup m|}$$

Ecuación 5 Magnitud del gradiente.

En términos simples, se busca si las variaciones entre cada grupo son tan bajas como las variaciones dentro de cada grupo. Esto permite que una curva muy leve esté presente en un borde y aun así se detecte. Esto es muy importante porque la etiqueta podría no ser perfectamente plana. Luego, cada grupo se ajusta para formar segmentos de línea mediante el uso de un procedimiento de mínimos cuadrados, que es una forma de ajuste lineal. Luego, la dirección del gradiente se marca con una muesca para mostrar la dirección del cambio. Esto permitirá detectar cajas en pasos posteriores. En la Figura 26 el resultado en la imagen de prueba a la derecha. Los puntos rojos son los puntos de inicio y fin de cada línea [20].



Figura 26 Segmentación lineal [19].

2.9.2.1.6 Líneas conectadas

Los segmentos de línea resultantes se conectan. Los segmentos permiten una diferencia en el punto de detección de una línea y el punto de inicio de la siguiente para que alcancen su longitud más 5 píxeles. Esto da como resultados más falsos positivos, pero la clave es que elimina los falsos negativos y puede tenerse en cuenta en el octavo paso. Esto se hace para cada punto rojo en el marco y debe eliminar los segmentos que están en el mismo lado de una figura. Si los segmentos en el mismo lado no pudieran agruparse, no sería posible buscar bucles de 4 en el paso 7 (22).

2.9.2.1.7 Detección de Quads

En los segmentos de línea se buscan bucles de longitud 4. Aquí es donde se une todo el procesamiento. Las líneas primero deben estar todas conectadas. Las muescas deben seguir en una rotación en el sentido de las agujas del reloj. Los segmentos de línea resultantes que quedan se llaman Quads. En la Figura 27 se muestra un ejemplo con las dos etiquetas detectadas del original con dos adicionales como falsos positivos (22).

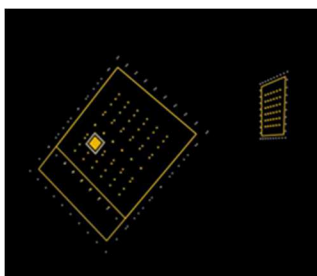


Figura 27 Detección de Quads con adquisición de carga útil [19].

Los falsos positivos estaban entre el borde del papel y el borde de AprilTag, y dentro de la carga útil de la etiqueta más grande. Esto muestra qué errores pueden ocurrir con su algoritmo de conexión de segmento de línea más liberal [19]. Sin embargo, los beneficios superan con creces los negativos. Si este algoritmo no se usara, las etiquetas tendrían que estar dentro de los 30 grados de ser planos, no los 80 grados que se pueden usar actualmente.

2.9.2.1.8 Quads decodificados

Los falsos positivos se eliminan al verificar cada Quad para datos de AprilTag. El área interna de un quad se denomina campo de carga útil. Los datos consisten en variaciones entre el blanco y el negro dentro de este campo de carga útil. Cada variación de blanco y negro tiene un código AprilTag diferente según la familia de la que forma parte y si una familia objetivo designada ha sido preseleccionada. Si se ha seleccionado una familia objetivo, los AprilTags que no estén en la familia seleccionada no se utilizarán para el cálculo de datos. Si no hay datos presentes, el Quad se descarta como falso positivo, lo que puede ocurrir para cualquier etiqueta que no esté en una familia seleccionada (22).

2.9.2.1.9 Refinar detecciones

Es necesario verificar la precisión de los Quads que quedan. Esto se hace primero eliminando cualquier Quad con datos que estén dentro de otros Quads con datos. Esto podría ser el resultado de que algunos de los AprilTags tengan otro código AprilTag dentro de su propia estructura de código. Luego se realiza una comparación con todas las etiquetas dentro de la familia designada utilizando un cálculo de distancia de Hamming (22). La distancia de Hamming es la cantidad de bits que deben corregirse para que coincida con una etiqueta en la biblioteca. La misma etiqueta podría dar múltiples resultados, por lo que la distancia de Hamming más baja es el código que se guarda. Se puede calcular una distancia relativa, guiñada, cabeceo y balanceo a partir de la etiqueta detectada, dada la ID o número de identificación del objeto.

CAPÍTULO 3 TRABAJO PROPUESTO

En este capítulo se describe la configuración del algoritmo de asistencia de navegación en el desplazamiento de un drón, se detalla el ambiente de pruebas y cómo se realiza la detección de las AprilTags; también, se describen las pruebas realizadas para configurar las acciones y los movimientos del drón.

3.1 ALGORITMO DE ASISTENCIA DE NAVEGACIÓN EN EL DESPLAZAMIENTO DE UN DRÓN

En esta tesis se propone un algoritmo de asistencia de navegación en el desplazamiento de un drón. En la Figura 28 se muestran las etapas principales del algoritmo.

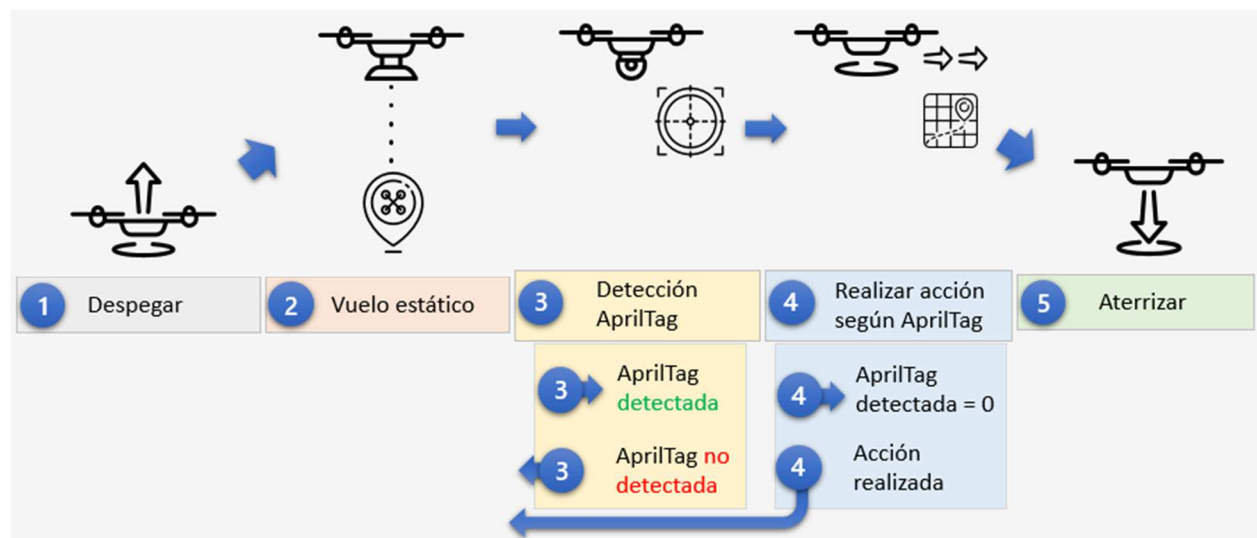


Figura 28 Etapas de funcionamiento del algoritmo.

Fuente: elaboración propia.

El algoritmo propuesto, presentado en la Figura 28, funciona de la siguiente manera:

- La primera etapa se acciona cuando el drón detecta la AprilTag para mandar la acción de despegue.
- En la segunda etapa, el drón realiza un vuelo estático después del despegue o después de realizar algún movimiento; se mantiene volando a una altura

aproximada de 80 cm del suelo. Esta etapa es la primera parte del ciclo anidado a donde regresa el drón después de realizar una acción.

- La etapa tres también forma parte del ciclo anidado. En esta etapa el sistema de visión manda imágenes al sistema de procesamiento (PC) para la detección de AprilTags; en el caso de no detección del AprilTag el drón regresa a la etapa dos (vuelo estático).
- Cuando el algoritmo detecta una AprilTag se lleva a cabo la etapa cuatro, donde el drón realiza la acción o movimiento según el AprilTag. Para los primeros nueve dígitos que están codificados en los AprilTags se tienen asignadas acciones o movimientos; el dígito cero está reservado para identificar la acción de aterrizar. Esta etapa también forma parte del ciclo anidado.
- En la etapa cinco es cuando el algoritmo detecta el dígito cero en la AprilTag, cuando esto sucede el drón inicia el aterrizaje.

El ciclo anidado de detección de AprilTags funciona como una máquina de estados para la realización de acciones o movimientos y el regreso al vuelo estático. Este ciclo comprende las etapas dos, tres y cuatro.

3.2 COMPONENTES DEL SISTEMA

A continuación, se describe la intervención de drón con el sistema de control de tierra, características del campo de vuelo y la familia de las AprilTags utilizadas.

3.2.1 Cuadricóptero Tello

El cuadricóptero utilizado en esta tesis es un Tello de la marca Ryze, que puede ser programado a través del SDK (software development kit) oficial con funciones limitadas y programación por bloques por el lenguaje Scratch [26], Cabe mencionar que la comunidad de software ha creado códigos de compatibilidad con el drón con los lenguajes C#, Java, Golang y Python.

Tello cuenta con reducidas dimensiones y peso liviano. Esto le permite el vuelo en interiores con facilidad, no cuenta con GPS; los sensores infrarrojos permiten el despegue y aterrizaje suave. Otras de las características con las que cuenta son los sensores de flujo óptico (optical flow) que lo ayudan a mantener una posición estática mientras vuela escaneando el suelo y manteniendo su posición; para escanear de forma correcta el campo de vuelo, éste debe estar iluminado de forma adecuada entre 300 a 10,000 lux para un correcto vuelo en el ambiente. Tello cuenta con un módulo Wi-Fi el cual le permite conectarse con un celular y ser controlado por medio de la aplicación o hacer una conexión a un ordenador o laptop. Tello genera una red Wi-Fi con dirección 192.168.10.1 IP estática; esto hace rápido el intercambio de información entre la laptop y el drón. Tello también cuenta con una cámara frontal de 5 Megapíxeles con estabilización electrónica por el procesador Intel Movidius Myriad 2 VPU [26].

3.2.2 Sistema de control de tierra

El equipo usado como sistema de control de tierra es una laptop Toshiba con procesador Intel i3, con tarjeta Wi-Fi necesaria para comunicarse con el drón; se utiliza el lenguaje de programación *Python 2.7* ya que cuenta con módulos para el control del drón Tello, además de otros módulos adicionales como OpenCV para la manipulación de las imágenes y un módulo de detección de AprilTags [33]. Se usa el sistema operativo Linux con la versión de Ubuntu 18.04, un sistema de código abierto que tiene la facilidad de instalar módulos de *Python 2.7* por medio de comandos desde la consola.

3.2.3 Marcadores Visuales AprilTag

Los AprilTags son una alternativa económica y versátil de posicionamiento basado en marcadores visuales. Los AprilTags son usados en este documento en varios escenarios para cubrir las pruebas, supliendo la necesidad de laboratorios de odometría visual o postes transmisores de radio que son infraestructuras costosas. Las AprilTags brindan al drón versatilidad en varios posibles escenarios de vuelo; solo se deben colocar las etiquetas en el trayecto para ofrecer autonomía al drón, no se requiere de infraestructura adicional.

En este trabajo se utilizaron etiquetas AprilTags de la familia 36h11 por sus atributos de eliminación de falsos positivos y los robustos atributos para la detección a larga distancia; las AprilTags se obtienen de forma económica imprimiéndolas etiquetas en una hoja blanca tamaño carta, ajustándolas a superficies rígidas para evitar una mala lectura del marcador visual. Las AprilTag se colocarán en ángulo recto con respecto al piso a una altura de 8cm en cajas de cartón o a 1cm del piso para la AprilTag nueve para el despegue.

3.2.4 Campo de vuelo

El campo de vuelo que se utilizó para este trabajo de investigación fue un espacio interior de 3.5m x 2.9m x 3m; se colocaron focos adicionales cada 110 cm para tener una iluminación adecuada entre 300 a 10,000 lux para un correcto vuelo en el ambiente. Las AprilTags se colocaron según el escenario de prueba, los cuales se describen en la Sección 3.3.6 Vuelo con detección de AprilTag.

3.3 ENTORNO DE DESARROLLO

A continuación, se describen los pasos para la configuración del sistema de control de tierra:

- El sistema que recibe la información del drón, la procesa y manda los comandos de vuelta al drón. Una vez instalado Ubuntu se deben ejecutar los siguientes comandos en la terminal para actualizar el sistema:

```
sudo apt-get update  
sudo apt-get upgrade
```

- Luego, se debe ejecutar el comando para descargar e instalar las herramientas de red y sus comandos para poder realizar conexiones:

```
sudo apt-get install net-tools
```

- Con las herramientas de red instaladas podremos hacer un ping al drón para confirmar que tenemos una conexión bidireccional; eso se hace prendiendo el drón y mandando un comando seguido de la dirección IP del drón:


```
ping 192.168.10.1
```

- De no lograr una conexión se debe revisar los drivers del Wi-Fi, en el caso afirmativo debe mostrar el tiempo que tarda en dar respuesta el drón a los datos enviados.
- Posteriormente se ejecutan una serie de comandos que son necesarios para manipular matrices, crear ventanas, desplegar imágenes, cambios de formatos entre imágenes, librería para diferentes transformaciones de información y que el código funcione correctamente con el algoritmo del trabajo de investigación [34]:

```
sudo apt install build-essential -y
sudo apt install python2.7 python-pip -y
sudo pip install cmake
sudo update-alternatives --install /usr/bin/python python /usr/bin/python2
150
sudo apt install aptitude -y
sudo aptitude install libboost-all-dev
sudo apt install python-matplotlib -y
sudo apt install python-imaging-tk -y
sudo apt install python-dev python-numpy -y
sudo python -m pip install pyzbar
sudo apt-get install -y python-dev pkg-config
sudo apt-get install -y libavformat-dev libavcodec-dev libavdevice-dev
sudo apt-get install -y libavutil-dev libswscale-dev libavresample-dev
libavfilter-dev
sudo pip install av
sudo pip install image
```

- Para instalar OpenCV (Software para el procesamiento de las imágenes) se usan los siguientes comandos:

```
sudo pip install opencv-python
sudo pip install opencv-contrib-python
```

- Para instalar el módulo de detección de AprilTag se usa el siguiente comando:

```
sudo pip install apriltag
```

- Para poder clonar varios repositorios de códigos se necesita instalar git con el siguiente comando:

```
sudo apt install git
```

- En el repositorio github existen varios códigos escritos en el lenguaje Python de código abierto para el control del drón, con base en estos códigos se creó el código del trabajo de investigación. Con los siguientes comandos se hace una copia local de los repositorios:

```
git clone https://github.com/dji-sdk/Tello-Python.git  
git clone https://github.com/hanyazou/TelloPy.git
```

- Se recomienda la instalación de alguna IDE para Python y crear carpetas para almacenar los códigos de los repositorios. En este proyecto se utilizó el IDE de Atom.

3.3.1 Iluminación del campo de vuelo

Para determinar la correcta iluminación del campo de vuelo se realizaron vuelos controlando el drón con la aplicación del Tello desde el celular para ver si la aplicación manda alguna alerta del sistema de posición por visión. El sistema visual de posicionamiento es activado automáticamente cuando el drón es prendido. El sistema manda mensajes de advertencia al celular cuando la zona en la que está volando tiene una iluminación fuera de los rangos de operación del sistema de posicionamiento por visión como marca el manual (iluminado entre 300 a 10,000 lux); el sistema no reconoce los patrones en el suelo si la iluminación no es adecuada para mantener un vuelo estable.

Se acondicionó el campo de vuelo agregando fuentes de luz (focos led 60w) cada 110cm a 130cm aproximadamente como se muestra en la Figura 29. Se usó un campo de vuelo bajo techo con dimensiones de 3.5m de ancho por 4m de largo con una altura de 3m. Se realizaron mediciones de la iluminación del campo de vuelo llegando a tener un promedio de iluminación de 400 lux; la cual fue adecuada para vuelos estables con el drón por todo el campo de vuelo, sin advertencia alguna de la aplicación. En la Tabla 5 se muestran las medidas de iluminación obtenidas en la columna de Iluminación medida en pruebas de vuelo desde celular; estas mediciones se realizaron desde el

celular con una aplicación de nombre Tello [27]; las zonas de medición son respecto a la Figura 29.

En las pruebas de vuelo semiautónomo con la lectura de AprilTags se notó que el piso reflejaba luz al sensor en zonas muy cercanas a la fuente de luz y el vuelo era diferente. Se colocaron cubiertas a los focos para evitar el reflejo. Se realizó una segunda medición, como se muestra en la Tabla 5 en la columna Iluminación con cubierta para evitar reflejos en el piso para lograr vuelos semiautónomos a través de la detección de AprilTags.

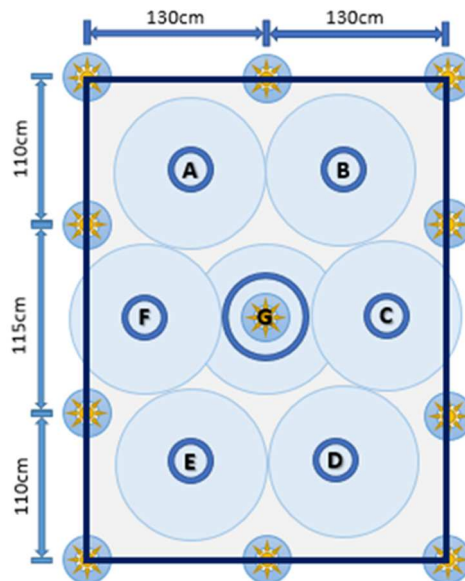


Figura 29 Distribución de fuentes de luz en el campo de vuelo y áreas de medición.

Fuente: elaboración propia.

Tabla 5 Iluminación medida de las áreas del campo de vuelo.

Fuente: elaboración propia

Área	Iluminación medida en pruebas de vuelo desde celular	Iluminación con cubierta para evitar reflejos en el piso para lograr vuelos semiautónomos con detección de AprilTag
A	415 lux	252 lux
B	415 lux	243 lux
C	410 lux	260 lux
D	375 lux	258 lux
E	375 lux	208 lux
F	415 lux	256 lux
G	415 lux	254lux

3.3.2 Vuelos del drón controlados desde sistema de control de tierra

Usando el código de *dji-sdk / Tello-Python* [35] se hicieron los primeros vuelos con el drón. Se mandaron comandos desde la laptop hacia el drón; en simultaneo se realizó la transmisión de imágenes con un retardo mínimo. Se observó que se envía una instrucción a la vez, el drón tiene respuesta hasta que acaba la instrucción anterior; las acciones tienen duración menor a 5 segundos; estas pueden ser movimiento hacia adelante, atrás, derecha, izquierda, ascenso, descenso del drón y giros. En este trabajo solo se consideraron movimientos con la orientación de la cámara al frente como la posición del despegue, sin giros para no perder la orientación de adelante, atrás y cambiar la referencia inicial.

3.3.3 Transmisión de imágenes del drón al sistema de control de tierra

Con base en el código de *hanyazou / TelloPy* [36] se realizaron transmisiones de imágenes sin volar el drón para hacer pruebas del funcionamiento de la cámara del drón; la cual cuenta con un campo de visión de 82.6° y transmisión de 30 fps, sin deformación en las líneas rectas (como se muestra en la Figura 30); la cámara del drón es útil para la detección de AprilTags sin necesidad de realizar calibraciones adicionales.

El estabilizador de imagen de Tello transmite imágenes nítidas con pocas interrupciones y sin deformación cuando la señal del Wi-Fi es buena, cuando la señal del Wi-Fi es mala las imágenes se muestran borrosas y llegan a perder resolución o perder píxeles de la imagen en la transmisión; solo se notó pérdida de la transmisión cuando la señal debía atravesar paredes, para el caso de las pruebas con detección de AprilTags, el campo de transmisión estará libre de obstáculos para contar con buena transmisión en toda la prueba.

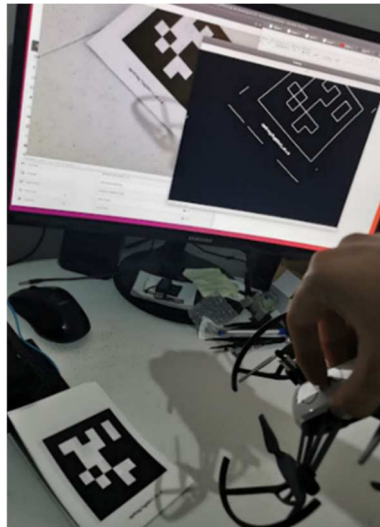


Figura 30 Pruebas transmisión de imagen.

Fuente: elaboración propia.

3.3.4 Detección de AprilTags de imágenes enviadas desde el drón al sistema de control de tierra

En la detección de AprilTags el drón manda las imágenes por Wi-Fi a la laptop. La laptop con la librería de Tellopy se conecta al drón, procesa las imágenes con el código en Python, decodifica la transmisión para obtener la transmisión de las imágenes que están en RGB de 24 bits cada canal 8 bits y las almacena en una matriz de dos dimensiones a escalas de grises. Con `cvtColor` (comando de OpenCV) se convierten las imágenes de color a imágenes en tono de grises (`cv2.COLOR_BGR2GRAY`). Luego, se utiliza el algoritmo de detección de las AprilTag para obtener las coordenadas de los centros del AprilTag, tanto en el eje X como en el eje Y. El origen de los ejes se encuentra en la parte

superior izquierda de la imagen. En la Figura 22, presentada en la Sección 2.9.1.2 Estimación de pose de AprilTag, se observa cómo funcionan las coordenadas en la detección de las AprilTags; con las coordenadas de los centros de los AprilTags detectados en la imagen se dibuja un círculo rojo en el centro del AprilTag detectado en la imagen.

La figura 31 muestra el funcionamiento de la detección de AprilTags, enviando imágenes a la laptop desde el drón (imagen A); en la imagen B se muestra cuando es detectada una AprilTag por laptop en las imágenes mandadas desde el drón.

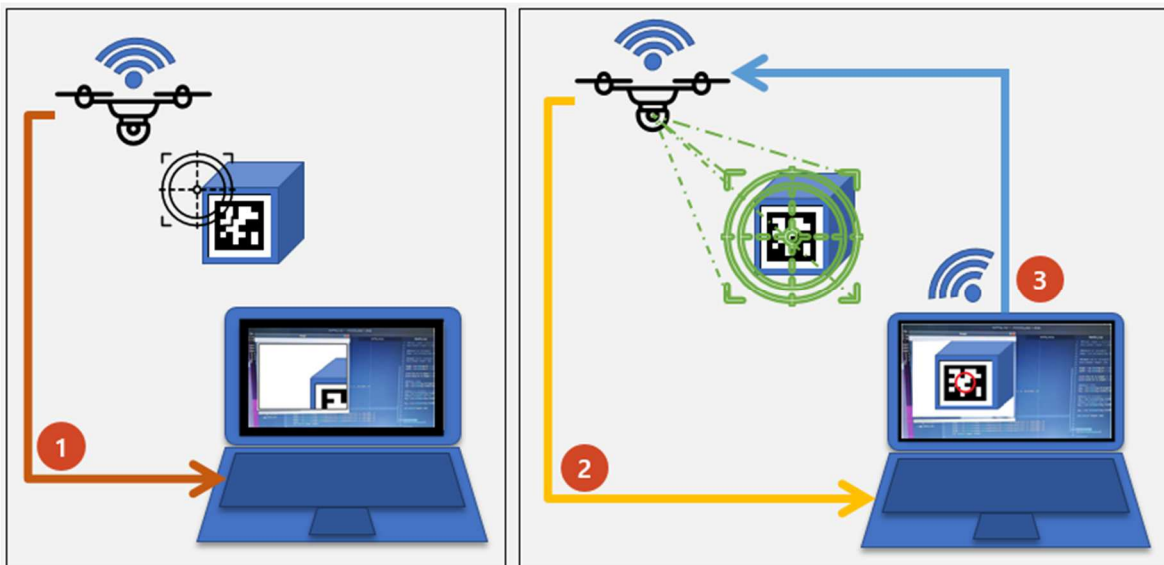


Figura 31 Transmisión de imágenes. A) transmisión de imágenes del drón a la laptop (1), B) transmisión de imagen del drón(2) respuesta de acción o movimiento a realizar por el drón cuando AprilTag es detectada (3).

Fuente: elaboración propia.

El paquete utilizado para la detección de AprilTags es el paquete: Apriltag 0.0.16 [33]. La función `detector = apriltag.Detector()` puede ser llamada ya que esté inicializada, pasando como parámetro la imagen en forma de matriz de dos dimensiones en tonos de grises. La llamada de la función regresa varios valores, si encuentra un AprilTag en la imagen. Los valores son los siguientes: Familia del Tag, Tag_ID, distancia de Hamming,

calidad de la detección, margen de decisión, homografía, centros del AprilTag y esquinas del AprilTag.

Una limitación encontrada en el paquete Apriltag 0.0.16 [33] es que solo detecta un AprilTag a la vez en la imagen y le da prioridad al menor número de Tag_ID. En las Figuras 32 y 33 se muestra el funcionamiento del algoritmo que coloca un círculo en el centro del AprilTag cuando es detectada; la Figura 32 contiene un Tag_ID 7 a la izquierda y Tag_ID 1 a la derecha, los dos AprilTags de dimensiones 8 x 8 cm a una distancia aproximada de 50 cm el algoritmo; en esta figura se observa que solo se dibuja el círculo que indica la detección en el centro del Tag_ID 1. En la Figura 33 el círculo se posiciona sobre el Tag_ID 0 de dimensiones 16.5 x 16.5 cm a una distancia de 250 cm que se encuentra al fondo de la imagen. Se puede observar como el algoritmo ignora los AprilTags más grandes en la imagen y busca el AprilTag con menor Tag_ID en la imagen; esta prioridad por el Tag_ID de menor tamaño se consideró para la asignación de acción o movimiento en el algoritmo de asistencia de navegación en el desplazamiento de un drón.

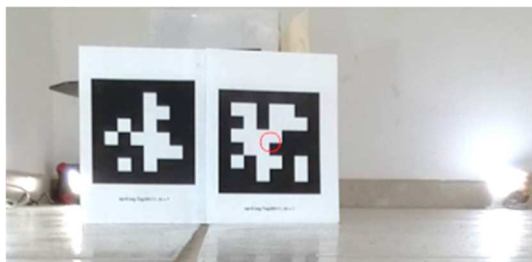


Figura 32 AprilTag Tag_ID 7 y Tag_ID 1 foto capturada desde Tello.

Fuente: elaboración propia.

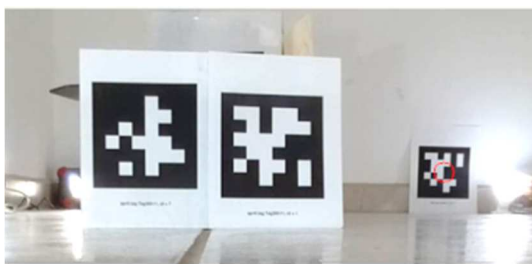


Figura 33 AprilTag Tag_ID 7, Tag_ID 1 y Tag_ID 0 foto capturada desde Tello.

Fuente: elaboración propia.

3.3.5 Angulo de detección de AprilTag

Para conocer el correcto ángulo de lectura y distancia de lectura del AprilTag se realizaron pruebas de detección del AprilTag con el drón. Las pruebas fueron realizadas a alturas de 0cm, 35cm y 90cm; el AprilTag se posicionó con un ángulo respecto a la cámara del drón de 0°, 45° y 60°, con distancias a la cámara de 50cm, 100cm, 150cm y 200cm. Se usaron dos tamaños de etiquetas de AprilTag de 8x8cm y de 16.5x16.5cm. En el transcurso de las pruebas se observó que para una correcta detección del Apriltag, todo el contorno de éste (tanto el marco blanco como el marco negro) deben de aparecer en la imagen; en caso contrario, el algoritmo no detecta el AprilTag. En la Figura 34 se muestra una simulación de los ángulos que se usaron en la prueba.

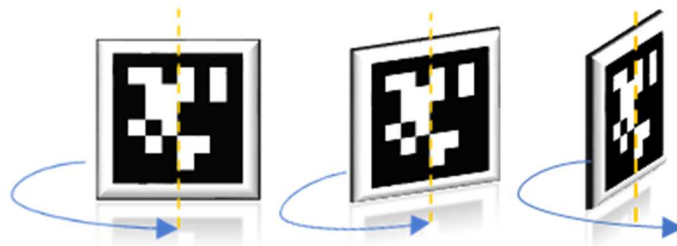


Figura 34 AprilTag con un ángulo respecto a la cámara del drón a 0°, 45°, 60°.

Fuente: elaboración propia.

A continuación, en la Tabla 6, se muestran los resultados obtenidos de las lecturas de los AprilTags con diferentes ángulos y a diferentes alturas de la cámara del drón.

Tabla 6 Resultados de detección de AprilTags 8x8cm y de 16.5x16.5cm.

Fuente: elaboracion propia

Detección de AprilTag a 50 cm de distancia de la cámara						
Tamaño	AprilTag de 8x8 cm			AprilTag de 16.5x16.5 cm		
Altura al suelo	0° respecto ángulo de la cámara	45° respecto ángulo de la cámara	60° respecto ángulo de la cámara	0° respecto ángulo de la cámara	45° respecto ángulo de la cámara	60° respecto ángulo de la cámara
0 cm	si	si	Si	no	no	no
35 cm	si	si	Si	si	si	si
90 cm	no	no	no	no	no	no

Detección de AprilTag a 100 cm de distancia de la cámara						
Tamaño	AprilTag de 8x8 cm			AprilTag de 16.5x16.5 cm		
Altura al suelo	0° respecto ángulo de la cámara	45° respecto ángulo de la cámara	60° respecto ángulo de la cámara	0° respecto ángulo de la cámara	45° respecto ángulo de la cámara	60° respecto ángulo de la cámara
0 cm	si	si	Si	si	si	si
35 cm	si	si	Si	si	si	si
90 cm	no	no	no	no	no	no

Detección de AprilTag a 150 cm de distancia de la cámara						
Tamaño	AprilTag de 8x8 cm			AprilTag de 16.5x16.5 cm		
Altura al suelo	0° respecto ángulo de la cámara	45° respecto ángulo de la cámara	60° respecto ángulo de la cámara	0° respecto ángulo de la cámara	45° respecto ángulo de la cámara	60° respecto ángulo de la cámara
0 cm	si	si	si	si	si	si
35 cm	si	si	si	si	si	si
90 cm	no	no	no	si	si	si

Detección de AprilTag a 200 cm de distancia de la cámara						
Tamaño	AprilTag de 8x8 cm			AprilTag de 16.5x16.5 cm		
Altura al suelo	0° respecto ángulo de la cámara	45° respecto ángulo de la cámara	60° respecto ángulo de la cámara	0° respecto ángulo de la cámara	45° respecto ángulo de la cámara	60° respecto ángulo de la cámara
0 cm	si	si	si	si	si	Si
35 cm	si	si	si	si	si	Si
90 cm	si	si	no	si	si	si

Con los resultados de detección de AprilTags obtenidos y para facilidad de las siguientes pruebas, el ángulo para colocar los AprilTags sobre el campo de vuelo será de 0° o cercano a éste con respecto a la cámara del drón; así como el AprilTag de 8x8cm solo será utilizado para el despegue, aterrizar o detecciones del Tag_ID cercanas a la pared por su detección en distancias pequeñas. Las etiquetas de 16.5x16.5cm se colocarán a mayor distancia para marcar la trayectoria del drón, debido que son detectados con diferentes alturas y distancias. El AprilTag de 16.5 x 16.5 cm colocado a una distancia de 150 cm fue detectado con un ángulo de 80°. Con lo observado anteriormente, se confirma que el algoritmo y las AprilTags son una herramienta estable y confiable para la detección.

3.3.6 Vuelo con detección de AprilTag

El algoritmo implementado para realizar los primeros vuelos con detección de AprilTag se desarrolló con base en el código de hanyazou/TelloPy/... /simple_takeoff.py [37] el funcionamiento del código utiliza parte de la lógica de los códigos que mandan las imágenes al sistema de control en tierra (laptop) por medio de Wi-Fi. El algoritmo de detección de AprilTag se lleva a cabo en el procesador de la laptop.

En la Figura 35, el ciclo anidado de detección de AprilTags de las etapas dos, tres y cuatro funciona como una máquina de estados, iniciando después del despegue y considerando la etapa dos como punto de retorno y de inicio del ciclo (vuelo estático). Posteriormente, en la etapa dos del vuelo estático se inicia la transmisión de imágenes a la laptop, esperando detectar AprilTags; esto inicia la etapa tres (detección de AprilTag); cuando una AprilTag es detectada por el procesamiento realizado en la laptop, se inicia la etapa cuatro (realizar acción según AprilTag). Cuando se finaliza la acción o movimiento generado en la etapa cuatro se regresa a la etapa dos (vuelo estático).

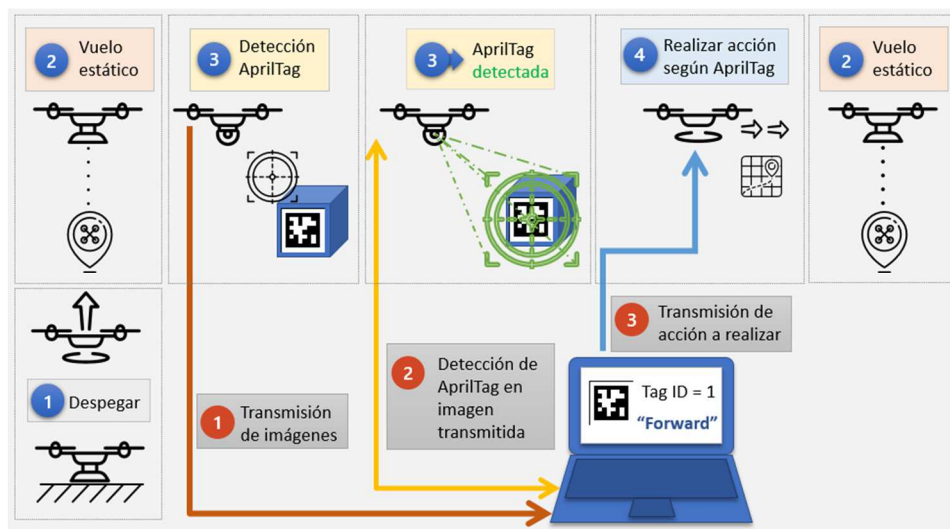


Figura 35 Detección de AprilTag en vuelo. Ciclo anidado de detección de AprilTags (Etapa 2, 3 y 4).

Fuente: elaboración propia.

En el algoritmo de asistencia de navegación en el desplazamiento de un dron se agrega la asignación de movimiento o instrucción a los primeros diez Tag_ID. En la asignación de instrucciones se consideró la importancia de la instrucción o movimiento, debido a que el algoritmo de detección de AprilTags da prioridad a los Tag_ID más pequeños; por lo que la acción más importante es el aterrizaje con la Tag_ID 0. Por cuestiones de seguridad, con solo mostrar este AprilTag al sistema de visión del dron, éste debe de aterrizarlo (con un mínimo retardo de 0.5 segundos). La acción que se realiza solo una vez al inicio del vuelo es la de despegue. A esta acción se le asignó el Tag_ID 9; esta sería la primera etiqueta que debería de ver el dron. Esta AprilTag no se muestra después de que el dron esté volando. Las demás acciones, movimientos o instrucciones se asignaron de acuerdo con lo que se presenta en la Tabla 7. En la realización de varias pruebas se observó que el dron después de realizar algún movimiento conservaba inercia del movimiento previo; para evitar este movimiento se agregaron retardos entre cada instrucción y se agregaron instrucciones adicionales para restar la inercia del movimiento y dejar al dron lo más estable después de haber realizado la acción para estar estático para el próximo movimiento. Estas acciones adicionales tienen menor duración y se agregaron al final de la acción principal para evitar afectar la acción o movimiento. Por ejemplo, cuando se realiza un movimiento a la derecha, después de

un retardo se manda la acción de movimiento a la derecha nulo para así frenar el movimiento.

Tabla 7 Tag_ID asignada a cada movimiento o instrucción.

Fuente: elaboracion propia

Tag_ID	Retardo en ejecutar la acción	Movimiento o Instrucción	Tiempo de ejecución de acción	Movimiento (M) o Instrucción (I)
0	0.5 seg	Aterrizaje (Land)	0	I
1	1 seg	Adelante (Forward)	4 seg.	M
2	1 seg	Atrás (Backward)	4 seg	M
3	1 seg	Derecha (Right)	4 seg	M
4	1 seg	Izquierda (Left)	4 seg	M
5	1 seg	Arriba (Up)	4 seg	M
6	1 seg	Abajo (Down)	4 seg	M
7	1 seg	Giro horario (Clockwise)	4 seg	M
8	1 seg	Giro contra horario (Counter clockwise)	4 seg	M
9	2 seg	Despegue (Takeoff)	2 seg	I

En la Figura 36 se muestra la secuencia del algoritmo de asistencia de navegación en el desplazamiento de un drón por detección de AprilTags. Para que el algoritmo se pueda ejecutar se debe prender el drón, conectar la laptop a la red Wi-Fi generada por el drón y correr el algoritmo.

El programa inicia en el estado de IDLE o periodo de inactividad. El programa se encuentra en ejecución sin realizar movimiento hasta que el sistema de visión detecta el Tag_ID 9, el cual tiene asignada la instrucción de despegar el drón. Una vez en el aire el drón tiene un retardo para eliminar la posible inercia del despegue; después de ese tiempo se mantiene en el aire a una altura aproximada de 80 cm respecto al piso donde el sistema de visión se mantiene en la espera de detectar algún AprilTag. Después del despegue el algoritmo se posiciona en el proceso de Mantenerse volando. En la lógica del algoritmo de detección de AprilTags, ésta sería la primera etapa del ciclo anidad

(vuelo estático) a donde regresaría la secuencia del programa después de realizar algún movimiento. Cuando se detecta una AprilTag, el programa deja de transmitir las imágenes y realiza el movimiento o instrucción según el Tag_ID detectado. La detección del Tag_ID 0 genera la instrucción de aterrizaje del drón.

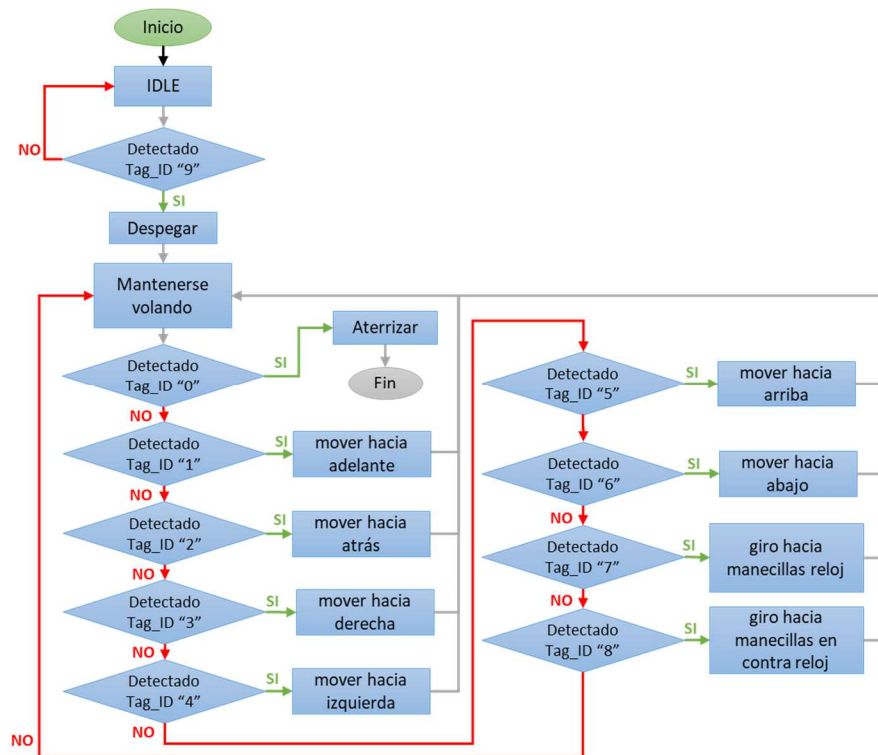


Figura 36 Lógica del algoritmo de detección de AprilTags.

Fuente: elaboracion propia.

Al correr el algoritmo se abre una ventana de consola que muestra información del drón como el nivel de batería, la intensidad de la señal y cuando es detectado un AprilTag; en una ventana adicional se despliegan las imágenes en tiempo real que está mandando el drón a la laptop, como se muestra en la Figura 37.

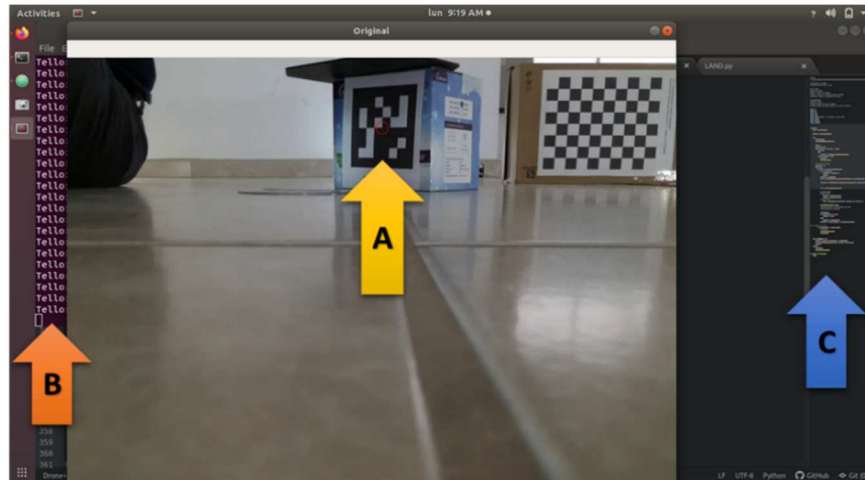


Figura 37 Ventana de detección de AprilTag (A), Consola de ejecución del programa (B), Ventana de IDE de programación (C).

Fuente: elaboración propia.

3.3.7 Capas del sistema y su estructura

El sistema de control de tierra (laptop) tiene el sistema operativo Linux distribución de Ubuntu sobre este sistema se corre el algoritmo de asistencia de navegación en el desplazamiento de un drón. En la Figura 38 se muestran los componentes de software y la interacción de la laptop con el drón por medio de la señal de Wi-Fi. El programa usa Python como lenguaje de programación, se usan los paquetes de h.264 Decoder para la decodificación de las imágenes, el paquete de OpenCV para procesamiento de las imágenes y el paquete de AprilTag para la detección de las AprilTags en las imágenes. El algoritmo de asistencia de navegación en el desplazamiento de un drón utiliza todos estos paquetes y el paquete de TelloPy para controlar el drón.

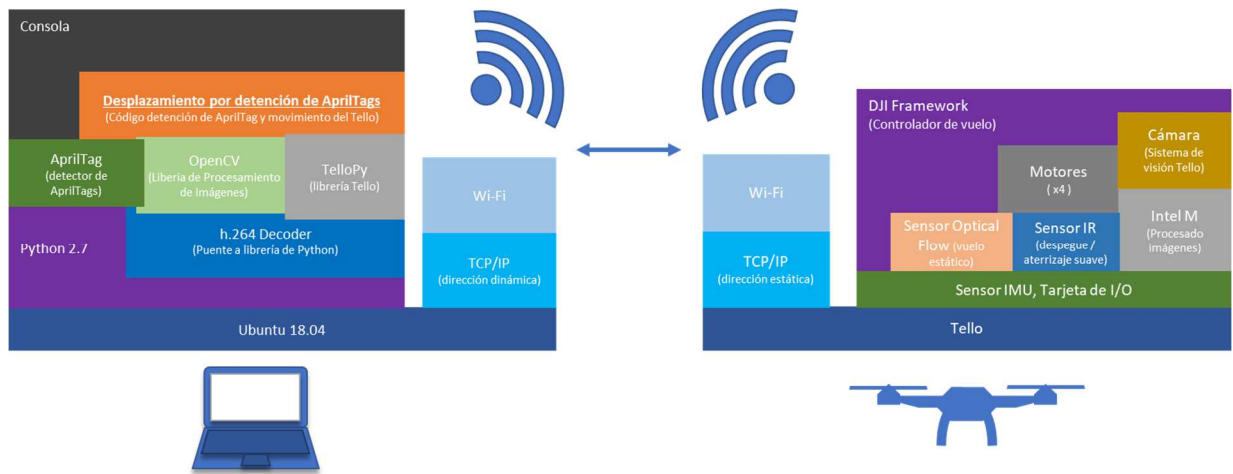


Figura 38 Esquema de componentes de software y comunicación entre dispositivos.

Fuente: elaboración propia.

3.3.8 Desplazamientos medidos después de detección de AprilTag

Se realizaron pruebas para conocer la distancia recorrida del drón con respecto a cada movimiento y observar la repetibilidad con cuatro escenarios.

- Tag_ID 1: movimiento hacia adelante
- Tag_ID 2: movimiento hacia atrás
- Tag_ID 3: movimiento hacia derecha
- Tag_ID 4: movimiento hacia izquierda

Se colocó la AprilTag con el Tag_ID 9 con etiqueta tamaño 8 x 8 cm a una distancia del drón de 60 cm al punto de despegue; después se colocaron las AprilTags con Tag_ID 1 de tamaño 16.5 x 16.5 cm a una distancia del 120 cm del punto de despegue. Esta prueba se realizó también para los AprilTag Tag_ID 2, Tag_ID 3 y Tag_ID 4; al finalizar cada movimiento se le mostró al drón el Tag_ID 0 para hacer el aterrizaje lo más cercano posible a donde quedó su movimiento final para medir la distancia desplazada desde el punto de despegue al punto de aterrizaje. Cada escenario se realizó por separado; los resultados se presentan en las Tablas 8, 9, 10 y 11. En las Figuras 39, 40, 41 y 42 se muestran gráficamente las distancias de las pruebas y una línea de tendencia.

Tabla 8 Resultados movimiento adelante.

Fuente: elaboración propia

Prueba Tag_ID 1	Número de prueba	Distancia al punto de despegue en m.
Secuencia de acciones: 1.- Lectura Tag_ID 9 – Despegue 2.- Lectura Tag_ID 1 – Adelante 3.- Lectura Tag_ID 0 – Aterrizar	1	2.208 m
	2	2.726 m
	3	2.528 m
	4	2.431 m
	5	2.196 m
	Promedio	2.417 m

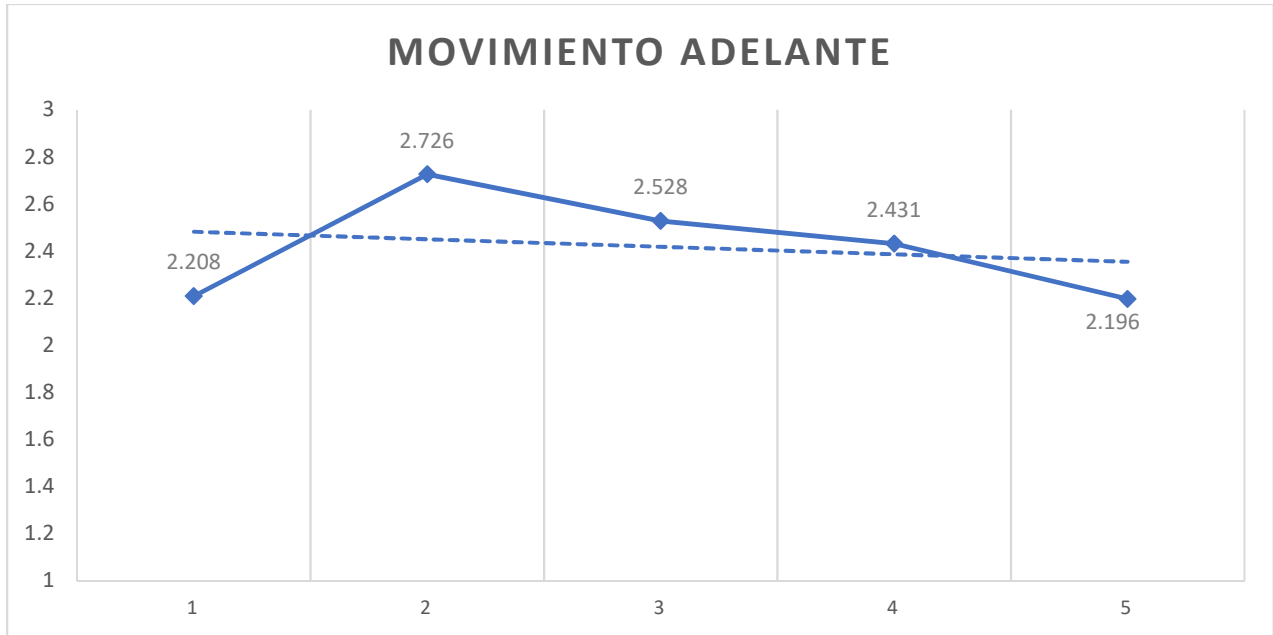


Figura 39 Resultados movimiento adelante (distancia en metros).

Fuente: elaboración propia

Tabla 9 Resultados movimiento atrás.

Fuente: elaboración propia

Prueba Tag_ID 2	Número de prueba	Distancia al punto de despegue en m.
Secuencia de acciones: 1.- Lectura Tag_ID 9 – Despegue 2.- Lectura Tag_ID 2 – Atrás 3.-Lectura Tag_ID 0 – Aterrizaje	1	1.914 m
	2	1.244 m
	3	1.799 m
	4	1.882 m
	5	1.978 m
	Promedio	1.763 m

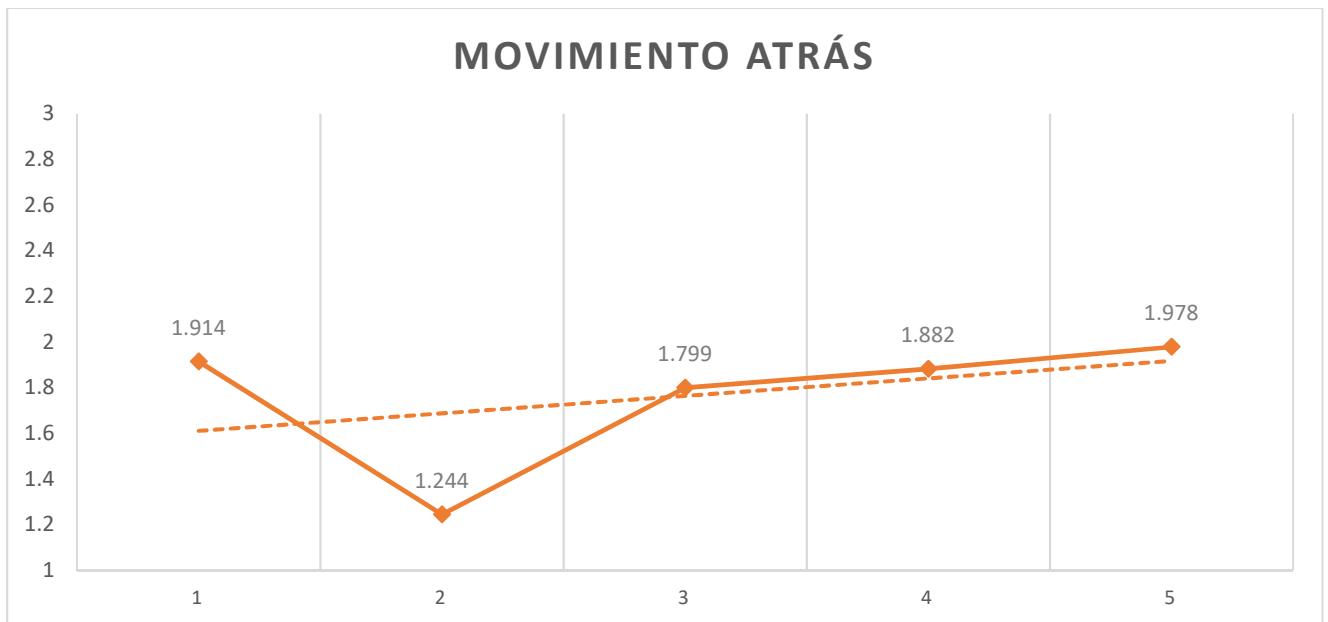


Figura 40 Resultados movimiento atrás (distancia en metros).

Fuente: elaboración propia

Tabla 10 Resultados movimiento derecha.

Fuente: elaboración propia

Prueba Tag_ID 3	
Secuencia de acciones:	
1.- Lectura Tag_ID 9 – Despegue	
2.- Lectura Tag_ID 3 – Derecha	
3.-Lectura Tag_ID 0 – Aterrizaje	

Número de prueba	Distancia al punto de despegue en m.
1	1.727 m
2	2.099 m
3	1.594 m
4	1.902 m
5	2.392 m
Promedio	1.942 m

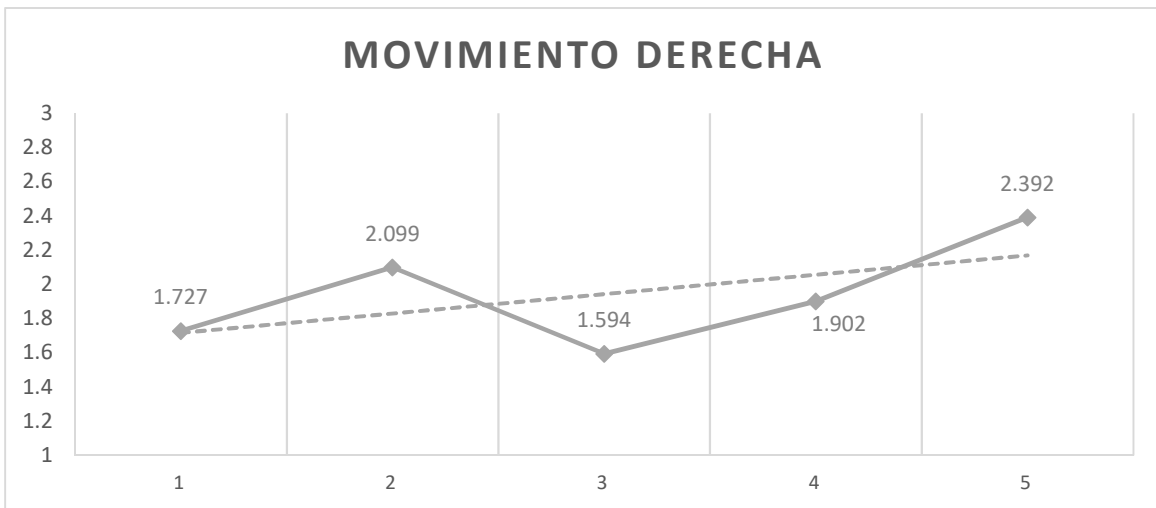


Figura 41 Resultados movimiento derecha (distancia en metros).

Fuente: elaboración propia

Tabla 11 Resultados movimiento izquierda.

Fuente: elaboración propia

Prueba Tag_ID 4	
Secuencia de acciones:	
1.- Lectura Tag_ID 9 –	
Despegue	
2.- Lectura Tag_ID 4 –	
Izquierda	
3.-Lectura Tag_ID 0 –	
Aterrizaje	

Número de prueba	Distancia al punto de despegue en m.
1	2.156 m
2	2.172 m
3	1.316 m
4	1.937 m
5	1.747 m
Promedio	1.865 m

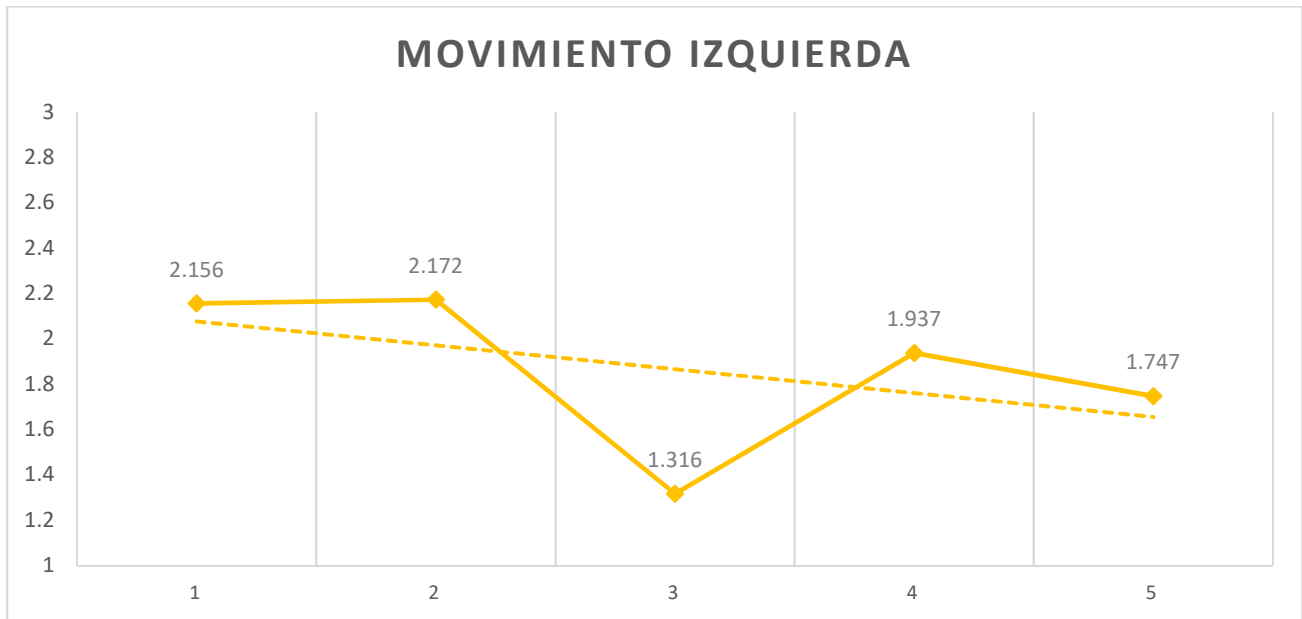


Figura 42 Resultados movimiento izquierda (distancia en metros).

Fuente: elaboración propia

Para los Tag_ID 5, 6, 7 y 8, que corresponden a arriba, abajo, giro en sentido del reloj y giro en contra de las manecillas del reloj, no se tomaron medidas, pero los comandos funcionan cuando el AprilTag es detectado de forma correcta.

Durante la realización de las distintas pruebas se probaron con diferentes fuentes de luz; se observó que el piso reflejaba la luz de las lámparas, éstas se cubrieron para evitar un reflejo en el piso; bajo esta iluminación los vuelos del drón fueron más estables. Las nuevas medidas de iluminación se mostraron en la columna 2 de la Tabla 5 Iluminación medida de las áreas del campo de vuelo, presentada en la Sección 3.3.1.

El resultado de las distancias recorridas por el drón después de la detección de la AprilTag se muestra en la Tabla 12. Conforme a los resultados presentados en esta tabla, se observa que el drón se desplaza en promedio una distancia de $2\text{m} \pm 500\text{cm}$ en la dirección según el AprilTag identificado; el Tag_ID 1 de desplazamiento hacia adelante logra el mayor desplazamiento y el Tag_ID 2 de desplazamiento hacia atrás cuenta con el menor desplazamiento. Las distancias promedio de cada movimiento se usarán como base para calcular la distancia teórica y compara con los resultados finales.

Tabla 12 Resultados desplazamiento AprilTag Tag_ID.

Fuente: elaboración propia

Resultados del muestro	Ventana de medidas	Promedio desplazamiento
Desplazamiento adelante, Tag_ID 1	Mínimo	2.196 m
	Promedio	2.417 m
	Máximo	2.726 m
Desplazamiento atrás, Tag_ID 2	Mínimo	1.244 m
	Promedio	1.763 m
	Máximo	1.978 m
Desplazamiento derecha, Tag_ID 3	Mínimo	1.594 m
	Promedio	1.942 m
	Máximo	2.392 m
Desplazamiento izquierda, Tag_ID 4	Mínimo	1.316 m
	Promedio	1.865 m
	Máximo	2.172 m

3.4 ESCENARIOS CON COMBINACIÓN DE MOVIMIENTOS CON APRILTAG.

Se crearon dos escenarios para que el drón complete un trayecto detectando AprilTags en el campo de vuelo, para probar la versatilidad y repetibilidad con la que cuenta el algoritmo.

El primer escenario es un trayecto en forma de “_|”, con desplazamiento hacia adelante y movimiento a la derecha. Los AprilTags son colocados en el siguiente orden:

1. Tag_ID 9 despegar
2. Tag_ID 1 avanzar hacia adelante
3. Tag_ID 3 moverse hacia la derecha
4. Tag_ID 0 aterrizar.

En el segundo escenario el trayecto tiene forma de “L”, con desplazamiento hacia delante y movimiento a la izquierda. Los AprilTags son colocados en el siguiente orden:

1. Tag_ID 9 despegar
2. Tag_ID 1 avanzar hacia adelante
3. Tag_ID 4 moverse hacia la izquierda
4. Tag_ID 0 aterrizar

Los trayectos se observan en las Figuras 43 y 44; las elipses grises con número muestran la actividad que realiza el drón en el campo de vuelo; los rectángulos biselados morados o amarillos muestran la ubicación de los AprilTags en el campo de vuelo con su Tag_ID; el círculo morado o círculo amarillo con el signo de más (+) muestra la colocación inicial del drón al momento de despegar; el círculo morado o amarillo con el signo de menos (-) muestra la posición esperada de aterrizaje del drón; la elipse gris 1 es el despegue de del drón; la elipse gris 2 es el desplazamiento hacia delante de la detección del Tag_ID 1; la elipse gris 3 es la detección de Tag_ID 3 para la Figura 43 con movimiento a la derecha, y en la Figura 44 es la detección del Tag_ID 4 con movimiento a la izquierda; la elipse gris 4 es la detección del Tag_ID 0 con la instrucción del aterrizaje.

Se midió la distancia del punto de despegue al punto de aterrizaje; el experimento se realizó diez veces para ver la repetibilidad del experimento en los escenarios. Debido a que el algoritmo cuenta con la limitante de detectar un AprilTag a la vez, se usó un sistema de cortinas para tapar los AprilTags con Tag_ID 3, 4 y 0; estos AprilTags se hacían visibles cuando el movimiento anterior había finalizado para que el drón pudiera realizar el siguiente movimiento.

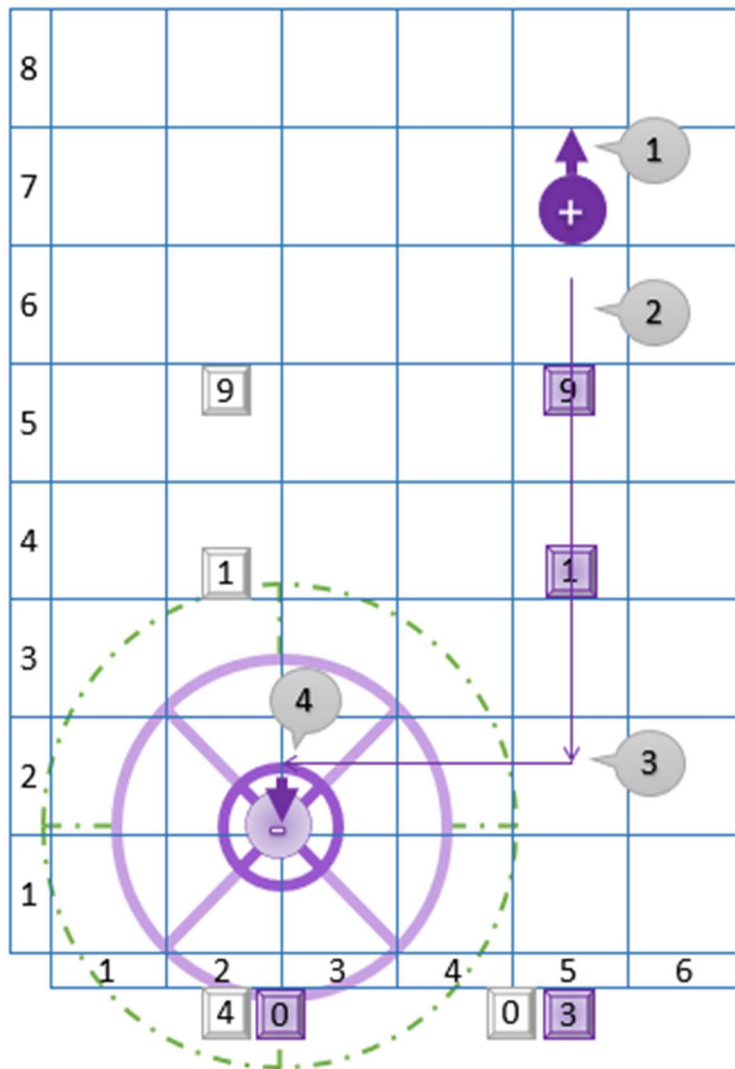


Figura 43 Escenario movimiento derecha: Tag_ID 9, 1, 3, 0.

Fuente: elaboración propia

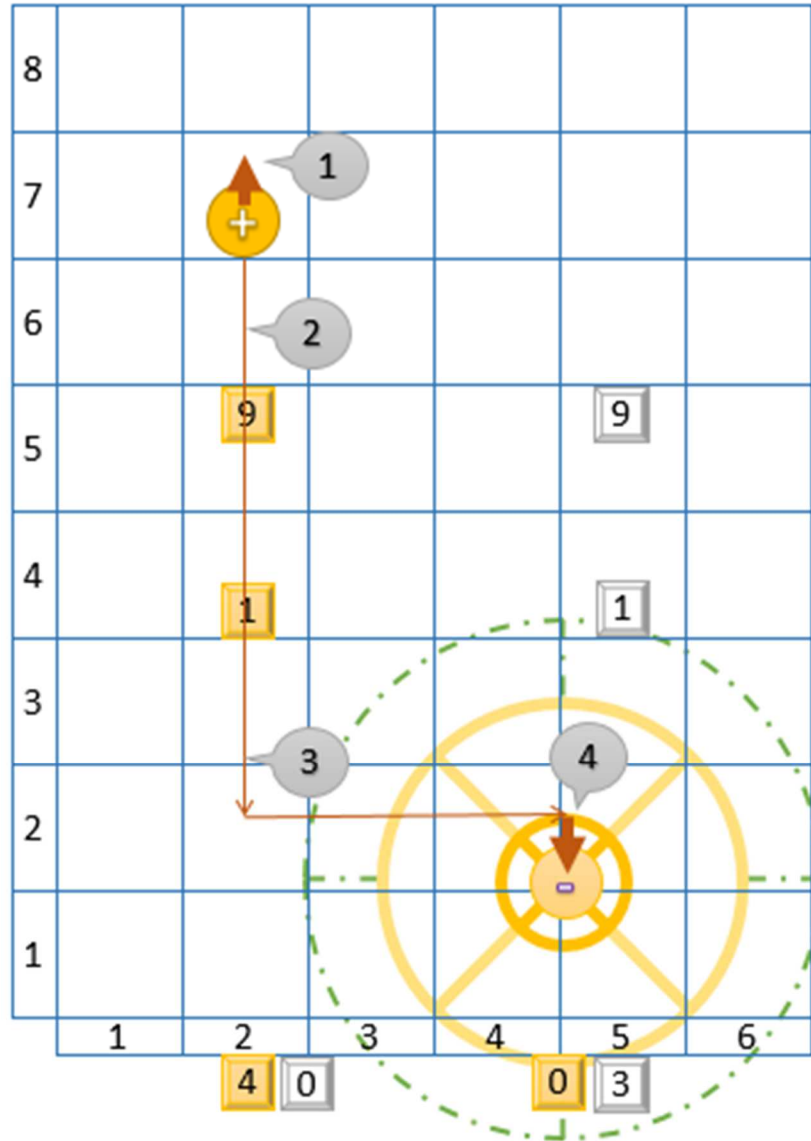


Figura 44 Escenario movimiento izquierda: Tag_ID 9, 1, 4, 0.

Fuente: elaboración propia

3.5 DESTINO FINAL TEÓRICO PARA ESCENARIOS CON COMBINACIÓN DE MOVIMIENTOS CON APRILTAG

Considerando las medidas de los desplazamientos promedio de la Tabla 13, se estima la distancia final que completará el drón al cumplir una trayectoria, combinando varios AprilTags en el campo de vuelo.

Tabla 13 Desplazamientos promedios de las AprilTag.

Fuente: elaboración propia

Resultados del muestro	Ventana de medidas	Promedio desplazamiento
Desplazamiento adelante, Tag_ID 1	Promedio	2.41 m
Desplazamiento atrás, Tag_ID 2	Promedio	1.76 m
Desplazamiento derecha, Tag_ID 3	Promedio	1.94 m
Desplazamiento izquierda, Tag_ID 4	Promedio	1.86 m

En la Figura 45 se muestra los puntos de despegue y los puntos finales tomando como base las distancias promedio de los experimentos. La distancia final teórica en los escenarios se marca como la suma del desplazamiento hacia adelante (2.4m) y la distancia de desplazamiento para la derecha (1.9m) en el escenario uno y desplazamiento para la izquierda (1.8m) en el escenario dos. La distancia final teórica se resalta con líneas azules. La trayectoria del drón se denota con líneas que conectan el punto de despegue con el área de aterrizaje, en el escenario uno se marca con líneas moradas y en el escenario dos se marca con líneas amarillas. En la Figura 45, la zona de aterrizaje se indica con una X dentro de un círculo; la zona de aterrizaje morada es para el escenario uno y la X dentro de un círculo amarillo es para el escenario dos; las flechas muestran los movimientos a realizar en las trayectorias.

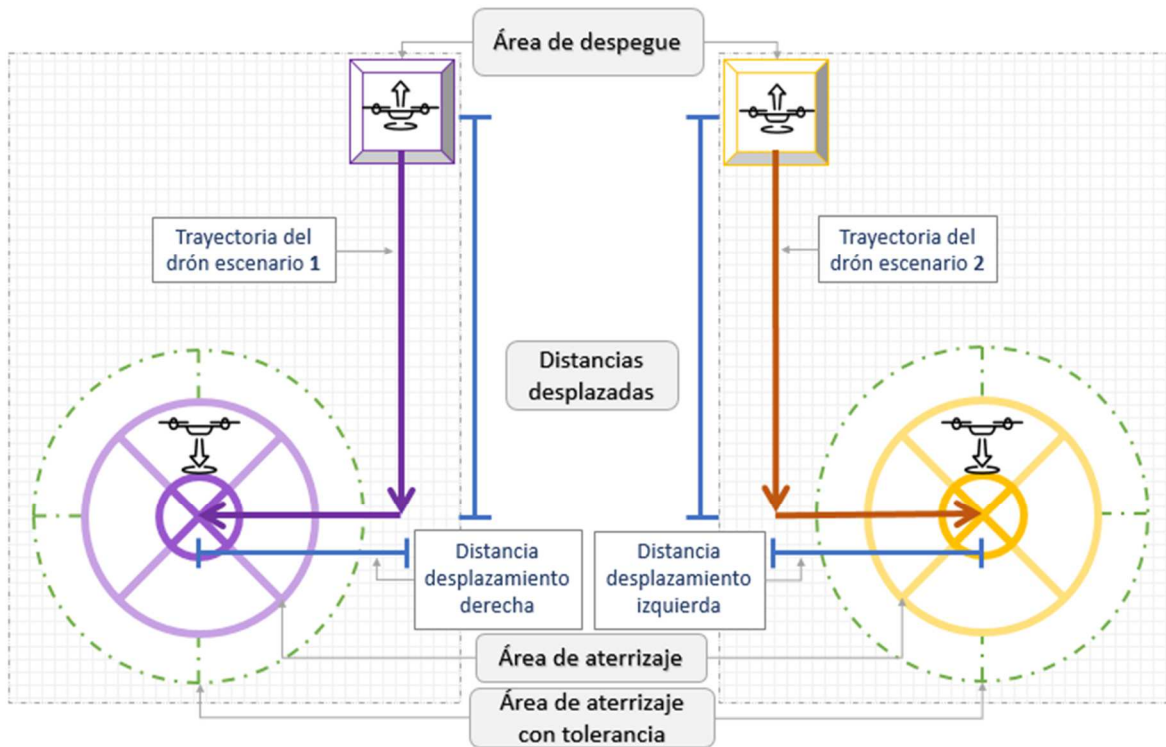


Figura 45 Distancia final teórica en escenario uno y dos.

Fuente: elaboración propia

CAPÍTULO 4 RESULTADOS

En este capítulo se explica cómo se realizaron las mediciones de los escenarios con combinación de movimientos de AprilTags; se muestran los resultados obtenidos en los escenarios uno y dos; se contrastan los resultados medidos contra el destino final teórico esperado.

4.1 MEDICIÓN DE DESPLAZAMIENTO DE LOS ESCENARIOS

Los resultados de los dos escenarios con varias AprilTags en el campo de vuelo fueron obtenidos a partir de 10 muestras de vuelos. Se colocó una marca en la posición de despegue del escenario uno y otra en el escenario dos, para iniciar desde la misma posición todos los vuelos. Con una marca en el piso, se identificó dónde aterrizó el drón al final de cada trayectoria; esto para tener un registro de la distribución de todos los vuelos y poder realizar la medición del aterrizaje de las 10 muestras al punto de despegue.

En la Figura 46 se muestra la forma en la que se mide la distancia del punto de despegue al punto de aterrizaje, denotado con una línea azul. En la Figura 46, los triángulos con número representan el lugar del aterrizaje y el número en ellos representa el número de la muestra; la cuadrícula en las Figura 46 representan el campo de vuelo.

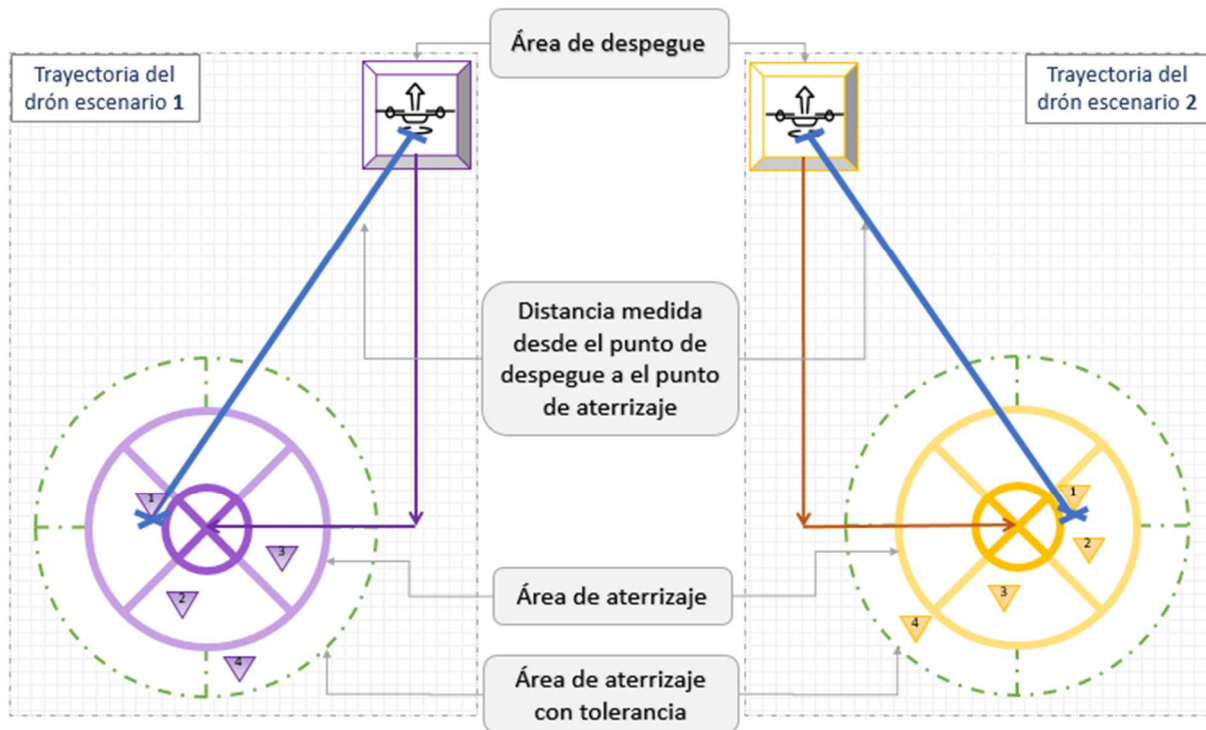


Figura 46 Medición de desplazamiento del punto de despegue al punto de aterrizaje.

Los triángulos no muestran aterrizajes reales, solo son demostrativos.

Fuente: elaboración propia

4.1.1 Resultados de escenario uno: avanzar y desplazarse a la derecha

El escenario uno consiste en que el drón realice las siguientes instrucciones o movimientos: despegue, movimiento hacia delante, movimiento a la derecha y aterrizaje.

Los resultados de diez muestreos son presentados en la Tabla 14A; se obtuvo un promedio de distancia recorrida de 2.881 m, con un mínimo de 2.671 m y un máximo de 3.339 m. En la Tabla 14B se indica cuáles aterrizajes quedaron dentro del área de aterrizaje (considerando una circunferencia de 35cm de radio), o fuera (como es el caso de las pruebas 4 y 5); también, se indican los aterrizajes dentro del área de aterrizaje con 50cm de circunferencia o fuera (como es el caso de la prueba 4, mostrado gráficamente en la Figura 47.

En la Figura 47 se presentan graficados los resultados de la Tabla 14A. Se usa una línea punteada para representar el promedio de distancias; se puede observar en esta figura que los muestreos 4 y 5 están alejados de la línea promedio.

Tabla 14 Distancias escenario uno. A) Distancia del punto de despegue al punto de aterrizaje. B) Relación de aterrizajes dentro de áreas de aterrizaje con circunferencia de 35cm y de 50cm de radio.

Fuente: elaboración propia

Número de prueba	Distancia al punto de despegue en m.
1	2.790 m
2	2.671 m
3	2.937 m
4	3.339 m
5	2.426 m
6	3.113 m
7	2.956 m
8	2.972 m
9	2.771 m
10	2.837 m
D. Mínimo	2.426 m
D. Promedio	2.881 m
D. Máximo	3.339 m

Número de prueba	Dentro del área de aterrizaje (circunferencia de 35cm de radio).	Dentro del área de aterrizaje con tolerancia (circunferencia de 50cm radio).
1	Dentro	Dentro
2	Dentro	Dentro
3	Dentro	Dentro
4	Fuera	Fuera
5	Fuera	Dentro
6	Dentro	Dentro
7	Dentro	Dentro
8	Dentro	Dentro
9	Dentro	Dentro
10	Dentro	Dentro

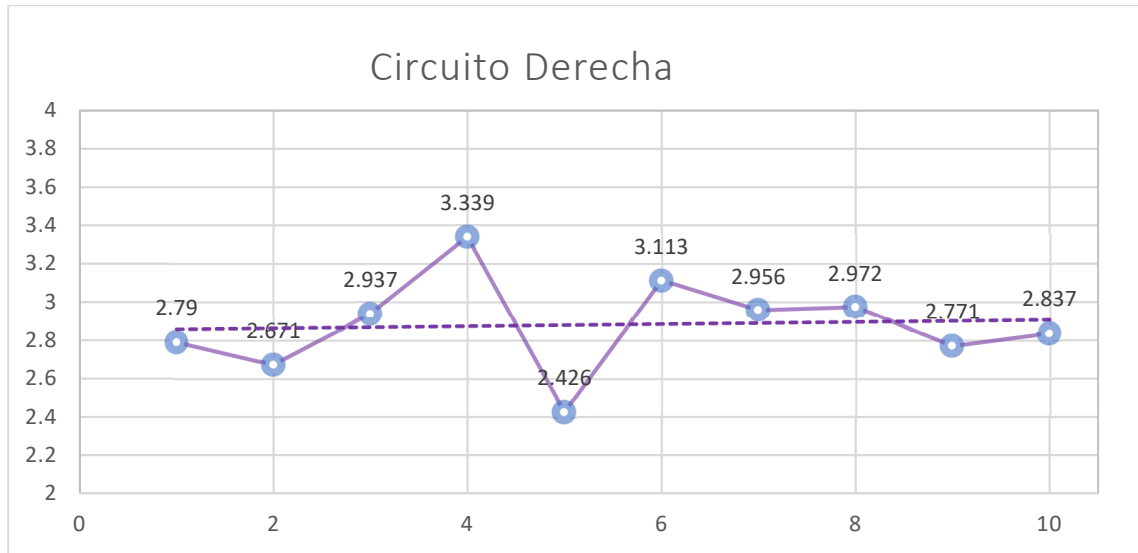


Figura 47 Gráfica de resultados desplazamiento escenario movimiento derecha Tag_ID 9, 1, 3, 0.

Fuente: elaboración propia

4.1.2 Resultados de escenario dos: avanzar y desplazarse a la izquierda

El escenario dos consiste en que el drón realice las siguientes instrucciones o movimientos: despegue, movimiento hacia delante, movimiento a la izquierda y aterrizaje.

Los resultados de diez muestreos son presentados en la Tabla 15A; se obtuvo un promedio de distancia recorrida de 2.933 m, con un mínimo de 2.530 m y un máximo de 3.219 m. En la Tabla 15B se indica cuáles aterrizajes quedaron dentro del área de aterrizaje (considerando una circunferencia de 35cm de radio), o fuera (como es el caso de las pruebas 3, 4, 6, 7, 8 y 9); también, se indican los aterrizajes dentro del área de aterrizaje con 50cm de circunferencia o fuera (como es el caso de las pruebas 8 y 9, mostradas gráficamente en la Figura 48).

En la Figura 48 se presentan graficados los resultados de la Tabla 15A. Se usa una línea punteada para representar el promedio de distancias; se puede observar en esta figura que los muestreos 4 y 5 están alejados de la línea promedio.

Tabla 15 Distancias escenario dos. A) Distancia del punto de despegue al punto de aterrizaje. B) Relación de aterrizajes dentro de áreas de aterrizajes con circunferencia de 35cm y de 50cm de radio.

Fuente: elaboración propia

Número de prueba	Distancia al punto de despegue en m.	Número de prueba	Dentro del área de aterrizaje (circunferencia de 35cm de radio).	Dentro del área de aterrizaje con tolerancia (circunferencia de 50cm radio).
1	3.028 m	1	Dentro	Dentro
2	3.088 m	2	Dentro	Dentro
3	2.530 m	3	Fuera	Dentro
4	2.972 m	4	Fuera	Dentro
5	2.988 m	5	Dentro	Dentro
6	3.011 m	6	Fuera	Dentro
7	2.231 m	7	Fuera	Dentro
8	3.180 m	8	Fuera	Fuera
9	3.219 m	9	Fuera	Fuera
10	3.085 m	10	Dentro	Dentro
D. Mínimo	2.231 m			
D. Promedio	2.933 m			
D. Máximo	3.219 m			

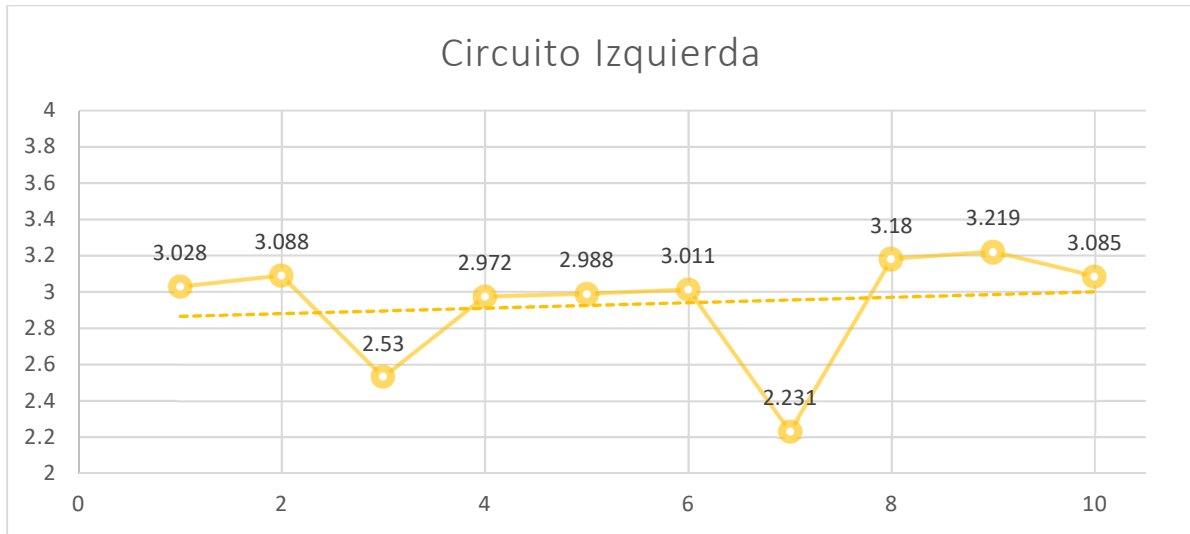


Figura 48 Gráfica de resultados desplazamiento escenario movimiento izquierda Tag_ID 9, 1, 4, 0.

Fuente: elaboración propia

4.2 POSICIONES DE LOS ATERRIZAJES DEL DRÓN DE LOS ESCENARIOS 1 Y 2

En la Figura 49 se muestran las posiciones de los aterrizajes del drón después de los desplazamientos de los escenarios 1 y 2; los triángulos en color morado corresponden a la posición de aterrizaje del escenario uno; los triángulos en color amarillo corresponden a la posición de aterrizaje del escenario dos.

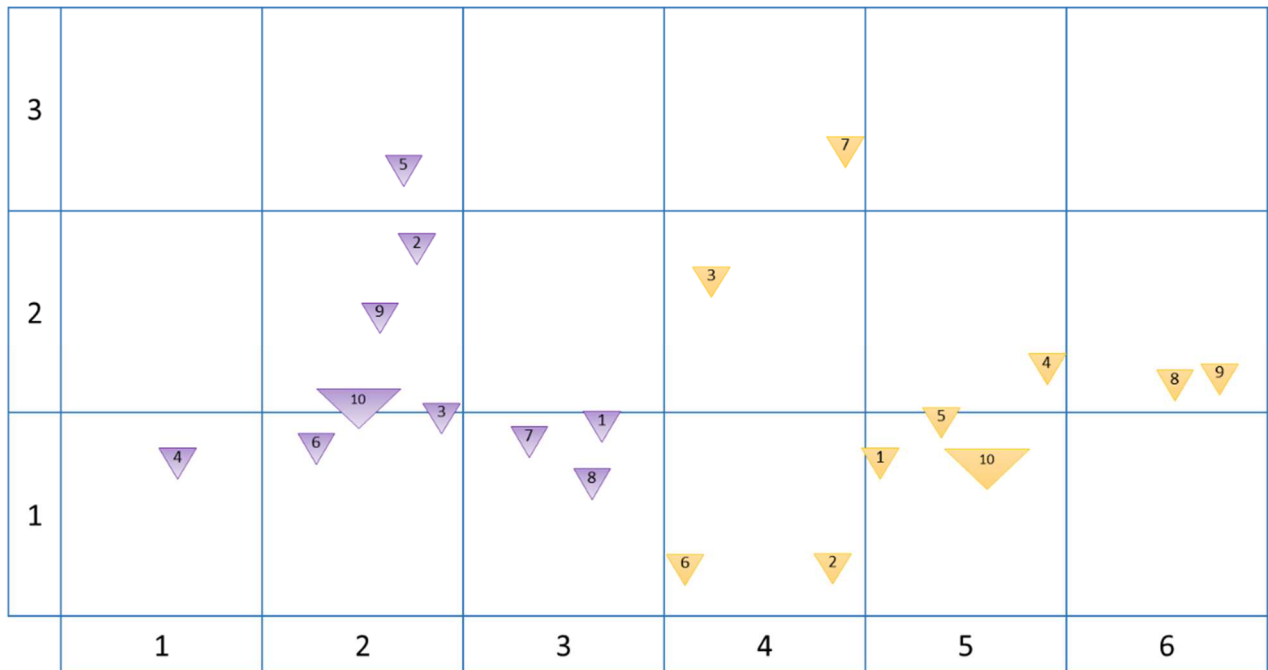


Figura 49 Mapa de resultados. Escenario uno (lado izquierdo con triángulos en color morado) y escenario dos (lado derecho con triángulos en color amarillo).

Fuente: elaboración propia

En la Figura 50 se muestran las posiciones de los aterrizajes del drón con áreas de aterrizaje con circunferencias con 35cm de radio y con 50 cm de radio; también, se marca la silueta del perímetro de los aterrizajes y se marcan con una X los aterrizajes que quedaron fuera del área de aterrizaje con tolerancia.

La cuadrícula representa cuadrados de 45 cm que se tomó como referencia para la reproducción grafica de los aterrizajes presentados en la Figura 50.

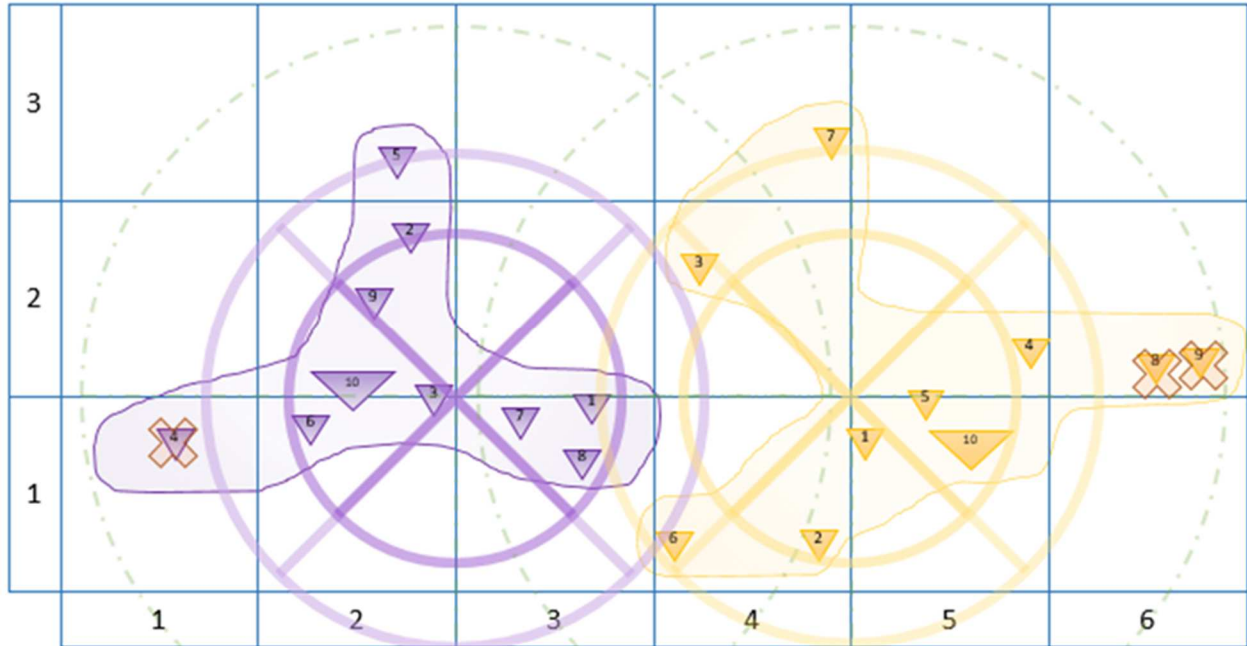


Figura 50 Mapa de resultados con áreas de aterrizaje (circunferencias con 35cm de radio y 50 cm de radio). Escenario uno (lado izquierdo con triángulos en color morado) y escenario dos (lado derecho con triángulos en color amarillo).

Fuente: elaboración propia

4.3 CÁLCULO DE DESPLAZAMIENTOS TEÓRICOS ESPERADOS

Para los cálculos de los resultados teóricos esperados se consideraron las medidas de distancia promedio obtenidas en el experimento presentado en la Sección 3.5 Destino final teórico para escenarios con combinación de movimientos con AprilTag. Las distancias consideradas para los movimientos del drón según el AprilTag Tag_ID son las siguientes:

- 2.41m para desplazamiento hacia adelante (Tag_ID 1)
- 1.94m para desplazamiento a la derecha (Tag_ID 3)
- 1.86m para desplazamiento a la izquierda (Tag_ID 4)

Tomando como base estas medidas, se calcula la distancia final para los escenarios uno y dos. La distancia calculada desde el punto de despegue hasta el punto de destino del escenario uno es de 3.09 m en diagonal a la derecha; y la distancia calculada desde

el punto de despegue hasta el punto de destino del escenario dos es de 3.04 m en diagonal a la izquierda. Para ambos casos, se tomó como referencia el frente del drón, como se muestra en la Figura 51.

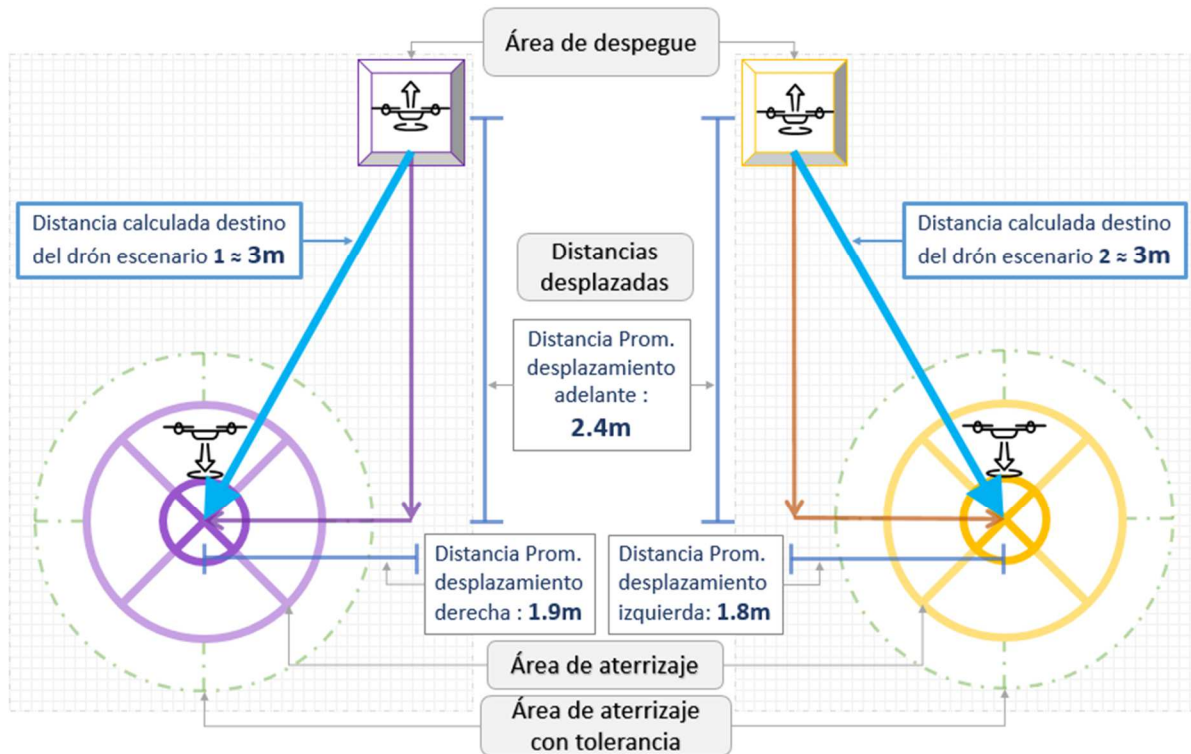


Figura 51 Destino calculado final del drón.

Fuente: elaboración propia

Para el cálculo de la distancia entre los puntos de despegue y aterrizaje se usó el Teorema de Pitágoras; se calculó la hipotenusa considerando como cateto adyacente al desplazamiento hacia adelante y como cateto opuesto a los movimientos de desplazamiento a la derecha o a la izquierda, dependiendo del escenario de prueba.

En la Tabla 16 se muestran los resultados de los cálculos de la distancia recorrida teórica. Se realizó el cálculo con los valores del experimento presentado en la Sección 3.5 Destino final teórico para escenarios con combinación de movimientos con AprilTag. En las columnas DMDAT1 y DMDAT2 (desplazamientos medidos después de detección de AprilTag) de la tabla están expresados los datos de los valores de las distancias de los

movimientos hacia adelante, a la derecha y a la izquierda para el desplazamiento mínimo, promedio y máximo (D. min, D. Prom y D. max). En las columnas DistCalE1 y DistCalE2 (distancia calculada desde el punto despegue al punto de aterrizaje) de la tabla se presentan los valores teóricos de la distancia donde debería de aterrizar el drón con respecto al punto de despegue; la distancia es de 3.09 m cuando se realizan los movimientos adelante y derecha (DistCalE1) y de 3.04 m cuando se realizan los movimientos adelante e izquierda (DistCalE2).

Tabla 16 Cálculo de destino final de los escenarios uno (movimientos adelante y a la derecha Tag_ID 9, 1, 3, 0) y dos (movimientos adelante y a la izquierda Tag_ID 9,1, 4, 0).

Fuente: elaboración propia

Escenario	Desplazamientos medidos después de detección de AprilTag: (DMDAT1) Adelante y derecha.	Escenario	Desplazamientos medidos después de detección de AprilTag: (DMDAT2) Adelante e izquierda.
1	Adelante	2	Adelante
D. min	2.19 m	D. min	2.19 m
D. Prom	2.41 m	D. Prom	2.41 m
D. max	2.72 m	D. max	2.72 m
	Derecha		Izquierda
D. min	1.59 m	D. min	1.31 m
D. Prom	1.94 m	D. Prom	1.86 m
D. max	2.39 m	D. max	2.17 m
Estimación teórica escenario 1	Distancia calculada desde el punto despegue al punto de aterrizaje. (DistCalE1) $DistCalE1$ $= \sqrt{Adelante^2 + Derecha^2}$	Estimación teórica escenario 2	Distancia calculada desde el punto despegue al punto de aterrizaje. (DistCalE2) $DistCalE2$ $= \sqrt{Adelante^2 + Izquierda^2}$
D. min	= 2.70 m	D. min	2.55 m
D. Prom	3.09 m	D. Prom	3.04 m
D. max	3.62 m	D. max	3.48 m

4.4 COMPARACIÓN DEL DESTINO FINAL TEÓRICO Y LA DISTANCIA AL PUNTO DE DESPEGUE, ERROR ABSOLUTO Y ERROR RELATIVO DE LOS RESULTADOS ESCENARIO UNO Y DOS

Utilizando los valores de los resultados de las distancias obtenidas de los experimentos entre los puntos de origen y destino (es decir, las distancias DMDAT1, DMDAT2, DistCalE1 y DistCalE2) de los escenarios 1 y 2, se realizó la comparación contra los valores reales (VR) para obtener el error absoluto y el error relativo, como se muestra en las Tablas 17 y 18. El error absoluto es la diferencia entre el valor real y el valor tomado como exacto (es decir, la distancia calculada desde el punto de despegue al punto de aterrizaje, DistCalE). El error relativo es el cociente entre el error absoluto y el valor considerado exacto después de varias iteraciones del experimento.

Para el escenario uno, de acuerdo con la Tabla 17, todos los valores en la columna del error relativo están dentro del rango de aceptación de 30% al punto de aterrizaje; siendo la prueba 5 la que tiene mayor error absoluto (0.66 m) y la prueba 4 la que tiene el mayor valor negativo en el error absoluto (-0.24 m).

Tabla 17 Distancia del escenario uno comparado contra valor teórico, error absoluto y error relativo, a partir de resultados tomados de la Tabla 14.

Fuente: elaboración propia

Número de prueba	Distancia al punto de despegue en m. Valor Real (VR)	Distancia calculada desde el punto de despegue al punto de aterrizaje (DistCalE1)	Error absoluto VR - DistCalE1	Error relativo (%) E. abs / DistCalE
1	2.79 m	3.09 m	$2.79 - 3.09 = 0.30$	$0.30 / 3.09 = 0.09 \rightarrow$
2	2.67 m	3.09 m	$2.67 - 3.09 = 0.41$	(9%)
3	2.93 m	3.09 m	$2.93 - 3.09 = 0.15$	$0.41 / 3.09 = 0.13 \rightarrow$
4	3.33 m	3.09 m	$3.33 - 3.09 = -$	(13%)
5	2.42 m	3.09 m	0.24	$0.15 / 3.09 = 0.05 \rightarrow$
6	3.11 m	3.09 m	$2.42 - 3.09 = 0.66$	(5%)
7	2.95 m	3.09 m	$3.11 - 3.09 = -$	$-0.24 / 3.09 = -0.08 \rightarrow (-$
8	2.97 m	3.09 m	0.02	8%)
9	2.77 m	3.09 m	$2.95 - 3.09 = 0.13$	$0.66 / 3.09 = 0.21 \rightarrow$
10	2.83 m	3.09 m	$2.97 - 3.09 = 0.11$	(21%)
			$2.77 - 3.09 = 0.31$	$-0.02 / 3.09 = -0.01 \rightarrow (-$
			$2.83 - 3.09 = 0.25$	1%)
				$0.13 / 3.09 = 0.04 \rightarrow$
				(4%)
				$0.11 / 3.09 = 0.03 \rightarrow$
				(3%)
				$0.31 / 3.09 = 0.10 \rightarrow$
				(10%)
				$0.25 / 3.09 = 0.08 \rightarrow$
				(8%)

Para el escenario uno se logró obtener un error absoluto promedio del 0.20 m y un error relativo promedio del 6%, como se muestra en la Tabla 18. Esto se encuentra dentro del rango de aceptación planteado en la tesis (30%).

Tabla 18 Error absoluto y error relativo de distancia promedio del escenario uno.

Fuente: elaboración propia

	Distancia al punto de despegue en m. Valor Real (VR)	Distancia calculada desde el punto despegue a punto de aterrizaje. (DistCalE1)	Error absoluto VR - DistCalE1	Error relativo (%) E. abs / DistCalE1
D. Mínimo	2.42 m	3.09 m	$2.42 - 3.09 = 0.66$	$0.66 / 3.09 = 0.21 \rightarrow$ (21%)
D. Promedio	2.88 m	3.09 m	$2.88 - 3.09 = 0.20$	$0.2 / 3.09 = 0.06 \rightarrow$ (6%)
D. Máximo	3.33 m	3.09 m	$3.33 - 3.09 = -$ 0.24	$-0.24 / 3.09 = -0.08 \rightarrow$ (- 8%)

Para el escenario dos, de acuerdo con la Tabla 19, todos los valores en la columna del error relativo están dentro del rango de aceptación de 30% al punto de aterrizaje; siendo la prueba 7 la que tiene mayor error absoluto (0.80m) y la prueba 9 la que tiene el mayor valor negativo en el error absoluto (-0.17 m).

Tabla 19 Distancia escenario dos comparado contra valor teórico, error absoluto y error relativo, a partir de resultados tomados de la Tabla 15.

Fuente: elaboración propia

Número de prueba	Distancia al punto de despegue en m. Valor Real (VR)	Distancia calculada desde el punto de despegue al punto de aterrizaje (DistCalE2)	Error absoluto VR - DistCalE2	Error relativo (%) E. abs / DistCalE2
1	3.02 m	3.04 m	3.02 - 3.04 = 0.01	0.01 / 3.04 = 0.003 ->
2	3.08 m	3.04 m	3.08 - 3.04 = -	(0%)
3	2.53 m	3.04 m	0.04	-0.04 / 3.04 = -0.01 -> (-
4	2.97 m	3.04 m	2.53 - 3.04 = 0.51	1%)
5	2.98 m	3.04 m	2.97 - 3.04 = 0.06	0.51 / 3.04 = 0.16 ->
6	3.01 m	3.04 m	2.98 - 3.04 = 0.05	(16%)
7	2.23 m	3.04 m	3.01 - 3.04 = 0.02	0.06 / 3.04 = 0.02 -> (2%)
8	3.18 m	3.04 m	2.23 - 3.04 = 0.80	0.05 / 3.04 = 0.01 -> (1%)
9	3.21 m	3.04 m	3.18 - 3.04 = -	0.02 / 3.04 = 0.009 ->
10	3.08 m	3.04 m	0.14	(0%)
			3.21 - 3.04 = -	0.80 / 3.04 = 0.26 ->
			0.17	(26%)
			3.08 - 3.04 = -	-0.14 / 3.04 = -0.04 -> (-
			0.04	4%)
				-0.17 / 3.04 = -0.05 -> (-
				5%)
				-0.04 / 3.04 = -0.01 -> (-
				1%)

Para el escenario dos se logró obtener un error absoluto promedio del 0.10 m y un error relativo promedio del 3%, como se muestra en la Tabla 20, lo cual se encuentra dentro del rango de aceptación planteado en la tesis.

Tabla 20 Error absoluto de distancia promedio y error relativo de distancia promedio del escenario dos.

Fuente: elaboración propia

	Distancia al punto de despegue en m. Valor Real (VR)	Distancia calculada desde el punto de despegue al punto de aterrizaje (DistCalE2)	Error absoluto VR - DistCalE2	Error relativo (%) E. abs / DistCalE2
D. Mínimo	2.23 m	3.04 m	$2.23 - 3.04 = 0.80$	$0.80 / 3.04 = 0.26 \rightarrow$ (26%)
D. Promedio	2.93 m	3.04 m	$2.93 - 3.04 = 0.10$	$0.10 / 3.04 = 0.03 \rightarrow$ (3%)
D. Máximo	3.21 m	3.04 m	$3.21 - 3.04 = -$ 0.17	$-0.17 / 3.04 = -0.05 \rightarrow$ (- 5%)

CAPÍTULO 5 CONCLUSIONES

En este capítulo se presentan las conclusiones de la tesis, se enlistan las contribuciones, se exponen las aportaciones y se plantea el trabajo futuro para el trabajo de investigación.

El problema estudiado en esta tesis es la navegación para un drón en interiores de forma autónoma o semiautónoma sin la necesidad de infraestructura adicional, para lo cual se propuso un sistema para la asistencia de navegación en interiores por medio del sistema visual del drón y de la detección de AprilTags, de manera que, el drón con un sistema de visión solo necesitara un campo de vuelo iluminado y AprilTags en el trayecto del drón.

Se crearon dos escenarios para que el drón complete un trayecto detectando AprilTags en el campo de vuelo, para probar la versatilidad y repetibilidad con la que cuenta el algoritmo. Los resultados del escenario uno y dos contra las distancias calculadas desde el punto despegue a punto de aterrizaje, así como también los errores absolutos y relativos, y la implementación y experimentación del algoritmo fueron satisfactorios, superando las expectativas; ya que se esperaban más aterrizajes fuera del área de las zonas de 35 y 50 cm. Se estimaban errores relativos cercanos al 35% o 40% con las pruebas iniciales, pero la estimación de tiempo de recorridos y el movimiento adicional al inicial para restar la inercia ayudó a mejorar el correcto desplazamiento y aterrizaje del drón. El algoritmo de detección de las AprilTag a distancia es un factor favorable para la navegación en interiores.

Con este proyecto, se pudo confirmar que el algoritmo de asistencia de navegación en el desplazamiento de un drón no es un sistema que dote al drón de completa autonomía, pero si es un sistema semiautónomo, debido a que el área de vuelo debe de contar con la iluminación adecuada y la colocación de los AprilTags. Después de varias pruebas el drón pudo realizar las trayectorias de vuelo de una forma adecuada;

pero sin duda es un sistema funcional que podría adaptar varias nuevas áreas de aplicación después de optimizar el código.

Se puede concluir que los AprilTags son una herramienta estable y útil para dotar al drón de una navegación y posicionamiento basado en marcadores visuales, superando el ángulo de los experimentos planeados y logrando una detección con un ángulo de visión de 80 grados respecto al sistema de visión del drón.

También, se puede resaltar que, si se cumplieron los objetivos planteados en esta tesis, pues se creó un algoritmo que con el sistema de visión realizó una navegación de desplazamiento en interiores siendo guiado de las AprilTags detectadas en el campo de vuelo completando la trayectoria.

TRABAJOS A FUTURO

El tema de sistemas de navegación semiautomáticos para drones en interiores por medio de sistema de visión puede derivar en nuevos proyectos de investigación, siguiendo con la aplicación de este trabajo se proponen las siguientes líneas:

1. Adaptar el algoritmo actual para centrar el AprilTag respecto al drón.
2. Modificar el algoritmo para precargar un plan de vuelo (lista de movimientos) y guiarse en orden del AprilTag, Tag_ID colocados en el campo de vuelo de forma ascendente para guiar y completar el trayecto.
3. Optimizar e integrar los algoritmos para ubicar su posición en el transcurso de su movimiento (SLAM).

APORTACIÓN DE LA TESIS

Desarrollo de un algoritmo sencillo de implementar y extender para ejemplificar una forma de utilizar un posicionamiento visual por marcadores en interiores, aplicando dos tecnologías: la detección de AprilTags y el vuelo de un drón de tamaño reducido para vuelo en interiores. Desarrollo de un algoritmo de asistencia de navegación en el desplazamiento para un drón Tello. Mediante el algoritmo desarrollado y nuevas mejoras se podrán cubrir actividades para los drones en interiores que ayudarán a realizar tareas repetitivas o peligrosas para el humano.

APORTACIÓN SOCIAL DE LA TESIS

Mediante el algoritmo desarrollado y nuevas mejoras se podrán cubrir actividades para los drones en interiores que ayudarán a realizar tareas repetitivas o peligrosas para el humano. Todos los sistemas dinámicos que cuenten con un sistema de visión que requieran orientación, navegación y control en interiores se pueden ver fácilmente beneficiados de este algoritmo para el desplazamiento por medio de la detección de las AprilTags.

REFERENCIAS BIBLIOGRÁFICAS

1. **España, Martín.** *Sistemas de Navegación Integrada con Aplicaciones.* s.l. : CONAE, 2017.
2. **Ferrer Mínguez, G.** *Integración Kalman de sensores inerciales INS con GPS en un UAV.* 2009.
3. **Lawrence, A.** *Modern inertial technology: navigation, guidance, and control.* . s.l. : Springer Science & Business Media., 2012.
4. *An Evaluation of Indoor Location Determination Technologies.* **Kevin Curran, Eoghan Furey, Tom Lunney, Jose Santos, Derek Woods and Aiden Mc Caughey.** 2, s.l. : Journal of Location Based Services, 2011, Vol. 5. 1748-9725.
5. *Drone Business Development Guide, The comprehensive guide for a successful start into UAV operation.* **Kay Wackwitz, Hendrik Boedecker, Andreas Froehlich.** Hamburg, Germany : Drone Industry Insights | Guide Droneii.com, 2016.
6. **Tech, Ryze.** ryzerobotics Manual del usuario v1.0 2018.05.
<https://www.ryzerobotics.com/es/tello/downloads>. [En línea] 2018. [Citado el: 21 de Enero de 2020.] https://dl-cdn.ryzerobotics.com/downloads/Tello/201806mul/Tello%20User%20Manual%20V1.0_ES.pdf.
7. **Jeff Deeds, Zach Engstrom, Caleb Gill, and Zack Woods.** *Vision Based Autonomous Control of a Quadcopter.* s.l. : Bradley University, 2017.
8. **Kawai, Ryuji.** github.com. [En línea] [Citado el: 6 de mayo de 2019.]
<https://github.com/r9kawai/ToyDroneWithAutopilotBarcodeReader/blob/master/ENVIRONMENT.TXT>.
9. **Winkler, Bernd.** <https://pypi.org/>. Python Software Foundation. [En línea] [Citado el: 7 de diciembre de 2018.] <https://pypi.org/project/apriltag/>.
10. **GmbH., Roboception.** Roboception rc_visard User Manual. Roboception GmbH. [En línea] Roboception GmbH., 2018. [Citado el: 28 de septiembre de 2019.] <https://doc.rc-visard.com/v1.2/en/tagdetect.html#sect-tag-re-identification>.
11. **Cuenca Sáenz, L. E., Cando, L., & Alfonso, J.** *Implementación de un sistema de navegación inercial, para mejorar la precisión de posicionamiento de un prototipo GPS en una trayectoria dentro de la ESPOCH.* Escuela Superior Politécnica de Chimborazo. Riobamba : TESIS, 2017.
12. *An Open Source and Open Hardware Embedded Metric Optical Flow CMOS Camera for Indoor and Outdoor Applications.* **Dominik Honegger, Lorenz Meier, Petri**

Tanskanen and Marc Pollefeys. Zurich, Switzerland : IEEE, 2013.
10.1109/ICRA.2013.6630805.

13. **Rodríguez Martín, Elías.** *Sistema de posicionamiento para un dron.* 2015.

14. **Paredes, Rubén Amezcua y Abimael, Pineda Salgado G.** *Tesina “Sistema de referencia inercia: Análisis de funcionamiento, fundamentos y evolución”.* México : s.n., 2012.

15. **González García, Nestor.** *Control de un cuadricóptero para navegación en interiores usando un sensor de flujo óptico.* Madrid : Universidad Pontificia Comillas Madrid, 2016.

16. *AprilTag: A robust and flexible visual fiducial system.* **Olson, Edwin.** Michigan : IEEE International Conference on Robotics and Automation, 2011, Vols. IEEE, 2011. 3400-3407.

17. *ARtag, AprilTag and CALtag fiducial systems comparison in a presence of partial rotation: manual and automated approaches.* **Shabalina, Ksenia & Sagitov, Artur & Sabirova, Leysan & Li, Hongbing & Magid, Evgeni.** s.l. : Springer, Cham, 2020.
10.1007/978-3-030-11292-9_27..

18. **Lefeber, A. A. J.** *Controlling of an single drone, Hovering the drone during flight modes.* Eindhoven : Technical Report TU/E Eindhoven, Department of Mechanical Engineering, 2015.

19. *Experimental Evaluation of UWB Indoor Positioning for Sport Postures.* **Ridolfi, Matteo & Vandermeeren, Stef & Defraye, Jense & Steendam, Heidi & Gerlo, Joeri & Clercq, Dirk & Hoebeke, Jeroen & De Poorter, Eli.** 1, s.l. : Sensors, 2018, Vol. 18. 168.

20. **Roger Miranda, Ruben Garrido, Luis Aguilar, Jose Herrero.** *Drones modelado y control de cuadrotores.* Mexico : Alfaomega Grupo Editorial, 2018. ISBN: 978-607-538-314-9.

21. **DJI.** dji cdn. [En línea] [Citado el: 1 de junio de 2020.]
https://dl.djicdn.com/downloads/a2/20170425/A2_User_Manual_V1.26_en.pdf.

22. **Master, Blob.** <https://github.com>. [En línea] 22 de septiembre de 2019.
https://github.com/dji-sdk/Tello-Python/blob/master/Tello_Video/tello.py.

23. **Umich.edu.** AprilTag. *umich.edu*. [En línea] [Citado el: 10 de noviembre de 2019.]
<https://april.eecs.umich.edu/software/apriltag.html>.

24. **Sira, Kr.** luxmeter. [En línea] luxmeter. [Citado el: 9 de julio de 2020.]
<https://play.google.com/store/apps/details?id=kr.sira.luxmeter&hl=es>.

25. **Dynamics, Boston.** bostondynamics. [En línea] [Citado el: 1 de junio de 2020.] <https://www.bostondynamics.com/atlas>.
26. **Aero, Catec.** catec.aero. [En línea] [Citado el: 1 de junio de 2020.] <http://www.catec.aero/es/aviónica-y-sistemas/equipamiento/banco-de-pruebas-o-testbed-en-interiores>.
27. *An indoor-outdoor positioning system based on the combination of GPS and UWB sensors.* **Belakbir, Abdelghani, Amghar, Mustapha y Sbiti, Nawal** . 1, s.l. : Journal of Theoretical and Applied Information Technology, 2014, Vol. 61. ISSN: 1992-8645.
28. **Net, Data.** data-net. [En línea] [Citado el: 8 de junio de 2020.] <http://www.data-net.com/education/barcodes.html>.
29. **Olson, Edwin.** AprilTag. *april robotics laboratory*. [En línea] The APRIL Robotics Laboratory at the University of Michigan investigates Autonomy, Perception, Robotics, Interfaces, and Learning, and is part of the Computer Science and Engineering department. Copyright. [Citado el: 08 de 07 de 2019.] <https://april.eecs.umich.edu/software/apriltag.html>.
30. **Bentham, Jeremy P.** What is a fiducial? [En línea] 2019. [Citado el: 18 de febrero de 2020.] <https://iosoft.blog/2019/09/02/raspberry-pi-position-detection-fiducial-tags/>.
31. **Wikipedia.** Inertial measurement unit. [En línea] [Citado el: 1 de junio de 2020.] https://es.qwe.wiki/wiki/Inertial_measurement_unit.
32. **Hanyazou.** github.com. [En línea] [Citado el: 10 de abril de 2019.] https://github.com/hanyazou/TelloPy/blob/develop-0.7.0/tellopy/examples/video_effect.py.
33. **Hanyazou.** github. [En línea] [Citado el: 11 de abril de 2019.] https://github.com/hanyazou/TelloPy/blob/develop-0.7.0/tellopy/examples/simple_takeoff.py.
34. **Wikipedia.** Triangulación. [En línea] [Citado el: 20 de junio de 2019.] <https://es.wikipedia.org/wiki/Triangulación>.
35. **Wikipedia.** Robot Navigation. [En línea] [Citado el: 15 de enero de 2020.] https://en.wikipedia.org/wiki/Mobile_robot_navigation.

ANEXO CÓDIGO

```
##Fuente
## https://github.com/hanyazou/TelloPy/blob/develop-0.7.0/tellopy/examples/video\_effect.py
#base tellopy base tello.py
##Descripcion
# funcionan algoritmo de asistencia de navegación en el desplazamiento de un drón
import av
import sys
import numpy
import tellopy
import apriltag
import time
from time import sleep
import traceback
import cv2.cv2 as cv2 # for avoidance of pylint error
def handler(event, sender, data, **args):
    drone = sender
    if event is drone.EVENT_FLIGHT_DATA:
        print(data)
def main():
    drone = tellopy.Tello()
    detector = apriltag.Detector()

    try:
        drone.subscribe(drone.EVENT_FLIGHT_DATA, handler)
        drone.connect()
        drone.wait_for_connection(40.0)
        retry = 3
        container = None
        while container is None and 0 < retry:
            retry -= 1
            try:
                container = av.open(drone.get_video_stream())
            except av.AVError as ave:
                print(ave)
                print('retry...')
#Variables control AprilTag
drone_on_ground = True
frame_skip = 300
for frame in container.decode(video=0):
    if 0 < frame_skip:
        frame_skip = frame_skip - 1
        continue
    start_time = time.time()
    frame_gray = cv2.cvtColor(numpy.array(frame.to_image()), cv2.COLOR_BGR2GRAY)
    dets = detector.detect(frame_gray)
    for det in dets:
        #obtener centro
        center_x = int(dets[0][6][0])
        center_y = int(dets[0][6][1])
        #obtener ID
        Tag_ID = dets[0][1]
        #dibujar centro
        frame_gray = cv2.circle(frame_gray,(center_x, center_y), 10, (0,0,255), 1)
##AprilTag, acciones o movimientos segun el Tag_ID
# 9 drone.takeoff()
    if Tag_ID == 9 and drone_on_ground == True:
        print("AprilTag ID ----- :"),Tag_ID
        drone.takeoff()
        drone_on_ground = False
        sleep(2)
        drone.down(20)
        sleep(0.5)
# 0 drone.land()
    elif Tag_ID == 0 and drone_on_ground == False:
        print("AprilTag ID ----- :"),Tag_ID
        sleep(0.5)
        drone.land()
        drone_on_ground = True
# 1 drone.forward
    elif Tag_ID == 1 and drone_on_ground == False:
        print("AprilTag ID ----- :"),Tag_ID
        sleep(0.5)
        drone.forward(50)
```

```

        sleep(4)
        drone.forward(0)
# 2 drone.backward
    elif Tag_ID == 2 and drone_on_ground == False:
        print("AprilTag ID -----: "),Tag_ID
        sleep(0.5)
        drone.backward(50)
        sleep(4)
        drone.backward(0)
# 3 drone.right
    elif Tag_ID == 3 and drone_on_ground == False:
        print("AprilTag ID -----: "),Tag_ID
        sleep(1)
        drone.right(40)
        sleep(4)
        drone.right(0)
# 4 drone.left
    elif Tag_ID == 4 and drone_on_ground == False:
        print("AprilTag ID -----: "),Tag_ID
        sleep(1)
        drone.left(40)
        sleep(4)
        drone.left(0)
# 5 drone.up
    elif Tag_ID == 5 and drone_on_ground == False:
        print("AprilTag ID -----: "),Tag_ID
        sleep(0.5)
        drone.up(20)
        sleep(3)
        drone.up(0)
# 6 drone.down
    elif Tag_ID == 6 and drone_on_ground == False:
        print("AprilTag ID -----: "),Tag_ID
        sleep(0.5)
        drone.down(20)
        sleep(3)
        drone.down(0)
# 7 drone.clockwise
    elif Tag_ID == 7:
        print("AprilTag ID -----: "),Tag_ID
        sleep(0.5)
        drone.clockwise(30)
        sleep(6)
        drone.clockwise(0)
# 8 drone.counter_clockwise
    elif Tag_ID == 8:
        print("AprilTag ID -----: "),Tag_ID
        sleep(0.5)
        drone.counter_clockwise(30)
        sleep(6)
        drone.counter_clockwise(0)
    else:
        pass
    cv2.imshow('Gray', frame_gray)
    cv2.waitKey(1)
    if frame.time_base < 1.0/60:
        time_base = 1.0/60
    else:
        time_base = frame.time_base
    frame_skip = int((time.time() - start_time)/time_base)
    if cv2.waitKey(1) & 0xFF== ord('q'):
        cv2.destroyAllWindows()
        cam.release()
except Exception as ex:
    exc_type, exc_value, exc_traceback = sys.exc_info()
    traceback.print_exception(exc_type, exc_value, exc_traceback)
    print(ex)
finally:
    drone.quit()
    cv2.destroyAllWindows()
if __name__ == '__main__':
    main()

```