



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS  
AVANZADOS DEL INSTITUTO POLITÉCNICO  
NACIONAL**

UNIDAD ZACATENCO

Programa de Sistemas Autónomos en Navegación Aérea y Submarina

# **“Establecimiento de conectividad en redes de comunicaciones mediante un vehículo aéreo autónomo”**

Tesis que presenta

**Ing. Carlos David Espinosa Ramírez**

Para obtener el grado de

**Maestro en Ciencias**

en

**Sistemas Autónomos de Navegación Aérea y Submarina**

Directores de Tesis:

Dr. Moises Bonilla Estrada

Dr. Aldo Gustavo Orozco Lugo



# Dedicatoria

---

A mis padres:

**Carlos Espinosa Morales y María Aracelly Ramírez Cervera**

Por siempre apoyarme incondicionalmente en todas las decisiones que he tomado en mi vida y siempre estar ahí cuando los necesito. Así como brindarme consejos que me han permitido ser una persona de bien con metas claras en la vida.

A mi madrina y mi tío Memo:

**Rosa Guadalupe Espinosa Morales y Guillermo Medina Camarillo**

Por ser mi guía los últimos años, un ejemplo de vida y siempre brindarme consejos que me han permitido salir adelante ante las adversidades. Por apoyarme siempre que los necesito y brindarme todo su amor y cariño.

A mis hermanos:

**Diana Isabel, Vladimir Antonio, Israel, Guillermo y Roy**

Por siempre alentarme a seguir adelante y motivarme a poder lograr mis metas.

A mis compañeros de grado:

**Sam, Alex, Moises, Ulises y Esteban**

Por apoyarme durante todo el proceso y brindarme tanto su amistad como su conocimiento, así como la motivación constante para lograrlo.

Carlos David Espinosa Ramírez



# Agradecimientos

---

Al CONAHCYT ya que sin su apoyo no habría sido posible lograr este trabajo de tesis, así como mi formación académica.

Al CINVESTAV, por proporcionarme las herramientas necesarias para obtener el grado, así como los conocimientos para desenvolverme en el ámbito laboral y académico.

A mis asesores, el Dr. Moisés Bonilla Estrada y al Dr. Aldo Gustavo Orozco Lugo, que me guiaron y proporcionaron los medios para desarrollar dicho trabajo, por brindarme sus conocimientos, su tiempo e infinita paciencia; y siempre estar pendientes del trabajo de tesis.

A mi madrina, Rosa Guadalupe Espinosa Morales, que me proporcionó su cariño, el apoyo y los medios para continuar mis estudios.

Carlos David Espinosa Ramírez



# Resumen

---

La tesis titulada “Establecimiento de conectividad en redes de comunicaciones mediante un vehículo aéreo autónomo” aborda la adaptación tanto de un prototipo de un vehículo aéreo autónomo como de un protocolo de comunicación para el establecimiento de comunicaciones en áreas remotas o de difícil acceso.

El objetivo principal de la investigación es comunicar dos estaciones base utilizando una memoria portátil en un vehículo aéreo como transporte de la información, haciendo uso de radios con el protocolo IEEE 802.11 para transferir los datos entre el vehículo aéreo y las estaciones base.

El trabajo comienza con una revisión del estado del arte en sistemas de comunicación mediante vehículos aéreos autónomos, así como de las tecnologías de comunicación inalámbrica empleando módulos de radio a computadora portátil con el protocolo IEEE 802.11.

Posteriormente, se diseñan y llevan a cabo una serie de pruebas y simulaciones para evaluar el rendimiento del vehículo aéreo autónomo adaptado para el escenario de comunicación que se plantea de portería a portería en una cancha de fútbol. Se adapta el equipo de comunicaciones para la capacidad de transmisión de datos y la estabilidad de la conexión en situaciones realistas.

A continuación, se propone una arquitectura de tramas para la comunicación a través del vehículo aéreo autónomo, considerando aspectos de seguridad, privacidad (con un código de redundancia cíclica) y capacidad de adaptación a condiciones cambiantes del entorno.

Finalmente, se presenta una serie de conclusiones y recomendaciones basadas en los resultados obtenidos durante la investigación. Se discuten las ventajas y desafíos de la propuesta, así como las posibles aplicaciones futuras en diversas áreas como el establecimiento de comunicación entre dos pueblos.



# Abstract

---

The thesis entitled “Establishment of Connectivity in communication networks by autonomous aerial vehicle” deals with the adaptation of both a prototype of an autonomous aerial vehicle and a communication protocol for the establishment of communications in remote or difficult-to-access areas.

The main objective of the research is to communicate two base stations using a portable memory in an aerial vehicle as information transport, making use of IEEE 802.11 protocol radios to transfer data between the aerial vehicle and the base stations.

The writing begins with a review of the state of the art in communication systems using autonomous aerial vehicles, as well as wireless communication technologies using IEEE 802.11 protocol with radio-PC modules.

Subsequently, a series of tests and simulations are designed and carried out to evaluate the performance of the autonomous aerial vehicle adapted for the goal-to-goal communication scenario on a soccer field. The communication equipment is adapted for data transmission capacity and connection stability in realistic situations.

Next, a frame architecture for communication through the autonomous aerial vehicle is proposed, considering aspects of security, privacy, (with **CRC**) and adaptability to changing environmental conditions.

Finally, a series of conclusions and recommendations based on the results obtained during the research are presented. The advantages and challenges of the proposal are discussed, as well as possible future applications in various areas such as the establishment of communication between two villages.

# Índice general

<b>Dedicatoria</b>	<b>I</b>
<b>Agradecimientos</b>	<b>III</b>
<b>Resumen</b>	<b>V</b>
<b>Abstract</b>	<b>VII</b>
<b>Índice de tablas</b>	<b>XI</b>
<b>Índice de figuras</b>	<b>XII</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Objetivos de la tesis . . . . .	2
1.1.1 Objetivo General . . . . .	2
1.1.2 Objetivos Particulares . . . . .	2
1.2 Justificación de la investigación . . . . .	2
1.3 Estructura de la tesis . . . . .	3
<b>2 Planteamiento del problema</b>	<b>5</b>
2.1 Problema . . . . .	5
2.2 Aspectos básicos de comunicación . . . . .	6
2.2.1 Modos de transmisión . . . . .	6
2.2.2 Envío de información . . . . .	8
2.2.3 Redes de comunicación . . . . .	8
2.2.4 Radio frecuencia . . . . .	9
2.2.5 Protocolos de comunicación . . . . .	10
<b>3 Vehículo aéreo</b>	<b>11</b>
3.1 Prototipo de un vehículo aéreo . . . . .	11
3.1.1 Estructura del prototipo . . . . .	11
3.2 Adaptaciones del prototipo . . . . .	15
3.2.1 Construcción de piezas . . . . .	17
3.3 Sistemas de transporte y comunicación . . . . .	21
3.3.1 Sistema de transporte . . . . .	22
3.3.2 Sistema de comunicaciones . . . . .	22
3.4 Modos de vuelo . . . . .	23

3.4.1	Modo <i>Autotune</i> . . . . .	23
3.4.2	Modo <i>Manual</i> . . . . .	25
3.4.3	Modo <i>Hover</i> . . . . .	25
<b>4</b>	<b>Radio comunicación</b> . . . . .	<b>27</b>
4.1	Comunicación extremo a extremo . . . . .	27
4.2	Comunicación de Radio a PC . . . . .	29
4.2.1	Interconexión Radio a PC . . . . .	30
4.2.2	Arreglo de bits . . . . .	31
4.3	Comunicación de PC a radio . . . . .	32
4.3.1	Mantenerme en la red . . . . .	33
4.4	RSSI . . . . .	33
4.4.1	Definición . . . . .	33
4.4.2	Monitoreo del <i>RSSI</i> . . . . .	33
4.5	Reconocimiento de tramas . . . . .	35
4.5.1	Identificar una trama . . . . .	36
4.5.2	Identificando el tipo de trama . . . . .	40
4.6	Creación de tramas . . . . .	41
4.6.1	Trama de datos . . . . .	43
<b>5</b>	<b>Solución del Problema</b> . . . . .	<b>45</b>
5.1	Planteamiento de la solución . . . . .	45
5.2	Plan de vuelo . . . . .	48
5.3	Programación de la misión . . . . .	53
5.3.1	Programación de la ruta de vuelo . . . . .	53
5.3.2	Programación del protocolo de comunicación . . . . .	54
5.4	Protocolo y pruebas de campo . . . . .	56
5.4.1	Modificaciones al campo <b>M</b> . . . . .	56
5.4.2	Transferencia del documento . . . . .	58
5.4.3	Comparación del RSSI entre A y B . . . . .	68
5.5	Comentarios de las pruebas de campo . . . . .	69
5.6	Resultados . . . . .	69
<b>6</b>	<b>Conclusiones</b> . . . . .	<b>71</b>
6.1	Resumen de los Objetivos de Investigación . . . . .	71
6.1.1	Objetivo General . . . . .	71
6.1.2	Objetivos Particulares . . . . .	71
6.2	Cumplimiento de los Objetivos . . . . .	72
6.3	Interpretación de los Resultados . . . . .	72
6.4	Limitaciones del Estudio . . . . .	72
6.5	Recomendaciones para Futuras Investigaciones . . . . .	73
6.6	Implicaciones Prácticas y Teóricas . . . . .	73
6.7	Conclusión General . . . . .	74
<b>A</b>	<b>Anexos</b> . . . . .	<b>75</b>

A.1	Cuadricóptero . . . . .	75
A.1.1	Modelo dinámico del Cuadricóptero . . . . .	75
A.1.2	ArduPilot . . . . .	76
A.1.3	Pixhawk . . . . .	76
A.1.4	Mission Planner . . . . .	76
A.2	Radios TTR - MRAPC . . . . .	77
A.2.1	Tabla de estructura de trama . . . . .	77
A.2.2	Tabla de instrucciones PC a MRAPC . . . . .	78
A.2.3	Tabla de instrucciones MRAPC a PC . . . . .	79
A.3	Protocolo <i>RQ Idle</i> . . . . .	80
A.3.1	RQ Idle: Solicitud explícita . . . . .	80
A.3.2	Poema enviado . . . . .	81
	<b>Referencias</b>	<b>83</b>

# Índice de tablas

4.1	Parámetros de configuración del puerto serial. . . . .	30
4.2	Formato de tramas de comunicación. . . . .	35
4.3	ID de dispositivo PC y radio. . . . .	36
4.4	Trama enviada por el radio. . . . .	36
4.5	Descripción de los campos de la trama enviada del radio a la PC. . . . .	37
4.6	Posición de campos de la trama de Comunicación. . . . .	40
4.7	Descripción de las instrucciones enviadas del radio a la PC [10]. . . . .	40
4.8	Trama enviada por la PC. . . . .	41
4.9	Descripción de los campos de la trama enviada de la PC al radio. . . . .	41
4.10	Descripción de las instrucciones enviadas de la PC al radio. . . . .	42
4.11	Estructura del campo <b>M</b> . . . . .	43
4.12	Descripción de las instrucciones enviadas de la PC al radio. . . . .	44
5.1	Estaciones Base y sus identificadores. . . . .	47
5.2	Formato de tramas de comunicación. . . . .	56
5.3	Estructura del campo <b>M</b> . . . . .	57
5.4	Bytes de cada campo de la subtrama del campo de Mensaje. . . . .	57
5.5	Campo ID de mensaje. . . . .	57
5.6	Posición de campos de la trama de comunicación enviada de la Base al <i>Drone</i> . . . . .	64
5.7	Nuevos identificadores en el campo <b>M</b> . . . . .	65
5.8	Tipos de mensaje. . . . .	65
A.1	Descripción de los campos de la trama enviada de la PC al radio. Parte 1 . . . . .	77
A.2	Descripción de los campos de la trama enviada de la PC al radio. Parte 2 . . . . .	78
A.3	Instrucciones de la PC al MRAPC. Parte 1 . . . . .	78
A.4	Instrucciones de la PC al MRAPC. Parte 2 . . . . .	79
A.5	Instrucciones del MRAPC a la PC. . . . .	79

# Índice de figuras

2.1	Escenario de $A$ , $B$ y $C$ .	5
2.2	Tipos de transmisión en el canal.	7
2.3	Ejemplo de mensaje.	8
2.4	Modelos de estaciones.	9
2.5	Ejemplo de la estructura de una trama.	10
3.1	Primer prototipo.	11
3.2	Tren de fibra de carbono x2.	12
3.3	Base del motor completa.	12
3.4	Brazos del cuadricóptero x4.	13
3.5	Base del brazo x4.	13
3.6	Centro del cuerpo.	13
3.7	Cuatro Rotores.	14
3.8	Batería LiPo de 3 celdas.	14
3.9	Pixhawk 1 y Ardupilot.	15
3.10	Carcasa Radio v1.	16
3.11	Carcasa <i>Raspberry</i> v1.	16
3.12	Base para <i>Pixhawk</i> v1.	17
3.13	Componentes apilados al centro.	17
3.14	Tren de aterrizaje.	18
3.15	Compartimientos de batería y electrónica.	19
3.16	Base de aluminio.	19
3.17	Compartimiento de electrónica.	20
3.18	Diagrama de conexión del vehículo aéreo.	20
3.19	Peso total del <i>Drone</i> (1.862 kg).	21
3.20	<i>Pixhawk 1 3DR</i> .	22
3.21	<i>Raspberry</i> Pi 3 Modelo B.	23
3.22	Ganancias PID.	23
3.23	Gráfica de “ <i>Autotune</i> ” de datos generados en <i>Dataflash log</i> .	24
4.1	Flujo de la información.	27
4.2	Partes del flujo de la información.	28
4.3	Comunicación PC a PC.	29
4.4	Medio de transmisión entre nodos.	29
4.5	Comunicación por el puerto serial de una PC a una impresora.	31
4.6	Registrarse en la red.	34

4.7	RSSI según la distancia. . . . .	35
4.8	Algoritmo de reconocimiento de trama. . . . .	39
4.9	Comunicación PC 1 a PC 2. . . . .	42
5.1	Campo de fútbol. . . . .	46
5.2	Red tipo malla. . . . .	46
5.3	Estaciones Base y sus identificadores. . . . .	47
5.4	Etapa 1 del plan de vuelo. . . . .	48
5.5	Etapa 2 del plan de vuelo. . . . .	49
5.6	Etapa 3 del plan de vuelo. . . . .	49
5.7	Etapa 4 del plan de vuelo. . . . .	50
5.8	Etapa 5 del plan de vuelo. . . . .	51
5.9	Etapa 6 del plan de vuelo. . . . .	51
5.10	Etapa 7 del plan de vuelo. . . . .	52
5.11	Resumen del plan de vuelo. . . . .	52
5.12	Diseño de ruta de vuelo. . . . .	53
5.13	Protocolo del radio. . . . .	55
5.14	Autonomía vs peso del <i>drone</i> . [22] . . . . .	56
5.15	Tiempos de espera del protocolo RQ Inactivo. . . . .	58
5.16	Etapa 1: Iniciando el código. . . . .	59
5.17	Etapa 2: Búsqueda del <i>drone</i> . . . . .	60
5.18	Comunicación de las bases con el radio. . . . .	62
5.19	Gráfica del <i>RSSI</i> obtenido del <i>drone</i> hacia la Base <i>B</i> . . . . .	63
5.20	Inicio de la transmisión de Base <i>B</i> con el <i>drone</i> . . . . .	63
5.21	Etapa 3: Inicia la transmisión. . . . .	64
5.22	Etapa 4: Base <i>A</i> solicita <i>RSSI</i> al <i>drone</i> . . . . .	66
5.23	Gráficas del <i>RSSI</i> en aproximación a la Base <i>A</i> . . . . .	66
5.24	Etapa 5 del plan de vuelo. . . . .	67
5.25	Comparación de gráficas. . . . .	68
A.1	Protocolo <i>RQ Idle</i> . . . . .	80



---

# Introducción

---

En la era actual de la tecnología y las comunicaciones, el acceso rápido y confiable a la información es fundamental para la sociedad. Sin embargo, existen áreas geográficas remotas o de difícil acceso donde la infraestructura de comunicaciones tradicional no es suficiente para garantizar una conectividad adecuada. En este contexto, el desarrollo de tecnologías innovadoras [1] [2] se convierte en una búsqueda constante para ampliar los límites de la conectividad y superar los desafíos impuestos por el terreno y las condiciones adversas [3][4].

En respuesta a esta necesidad, los vehículos aéreos autónomos (VAA) emergen como una prometedora solución para establecer comunicaciones en áreas remotas o de difícil acceso. Estos sistemas no tripulados poseen la capacidad de volar de manera autónoma, lo que los convierte en herramientas versátiles y adaptables para diversas aplicaciones, incluyendo el establecimiento de redes de comunicación [5][6].

Otro problema que se presenta es la fiabilidad de la transmisión de información por lo que es necesario adaptar un protocolo de comunicación para hacer uso junto al vehículo aéreo autónomo y lograr una transmisión libre de errores [7].

El presente trabajo de investigación tiene como objetivo principal adaptar un vehículo aéreo autónomo así como su protocolo de comunicación para establecer las comunicaciones en zonas donde la conectividad tradicional es limitada o inexistente. Se abordará la problemática de cómo implementar de manera efectiva el uso de un vehículo aéreo autónomo para transmitir señales y datos en tiempo real, asegurando la confiabilidad y la estabilidad de la conexión [8].

### 1.1. Objetivos de la tesis

Los objetivos específicos de esta tesis son los siguientes:

#### 1.1.1. Objetivo General

Comunicar dos estaciones base utilizando una memoria portátil en un vehículo aéreo como transporte de la información, haciendo uso de radios con protocolo IEEE 802.11 para transferir los datos entre el vehículo aéreo y las estaciones base.

#### 1.1.2. Objetivos Particulares

- Adaptar un prototipo de vehículo aéreo autónomo para desempeñar las funciones de establecer la conectividad en redes de comunicación.
- Adaptar un protocolo de comunicación y sobreponerlo a un protocolo establecido de mayor esfuerzo para asegurar la fiabilidad de nuestra información.
- Establecer un protocolo de comunicación con el porcentaje mínimo de error en la recepción.
- Programar una ruta de vuelo en el prototipo de vehículo aéreo para que vuele de forma autónoma de portería a portería en el campo de fútbol en las instalaciones del CINVESTAV.
- Validar experimentalmente el uso del vehículo aéreo autónomo transmitiendo información con el protocolo de comunicación adaptado.

### 1.2. Justificación de la investigación

El desarrollo de un sistema de comunicación basado en vehículos aéreos autónomos tiene el potencial de llevar la conectividad a nuevas fronteras, abriendo oportunidades para el acceso a información vital en zonas donde la infraestructura de comunicaciones convencional es insuficiente. Esta investigación busca contribuir al avance de la tecnología de vehículos aéreos

autónomos y su aplicación en el campo de las comunicaciones, generando conocimiento que pueda beneficiar a comunidades aisladas y mejorar la eficiencia de las operaciones en áreas de difícil acceso. Como por ejemplo un entorno donde dos pueblos se encuentren geográficamente divididos por una montaña, cuyo caso en particular presentaría complicaciones si se quiere llevar a cabo la implementación de cables o antenas sobre la montaña o rodearla. En este caso particular el uso de un vehículo autónomo aéreo es destacable.

El procedimiento empleado en el presente trabajo de tesis consistió en dividir el problema en dos secciones: la primera sección considera los radios con la comunicación; y la segunda, el vehículo aéreo autónomo. En el caso de los radios seguimos con una subdivisión, ya que se aproximó el mismo problema para dos plataformas distintas, un programa del *software MatLab* usando una PC y otro usando el *software de python* en una *Raspberry*.

### **1.3. Estructura de la tesis**

La presente tesis se organiza de la siguiente manera: En el Capítulo 2 se presentará el problema que se plantea resolver, ya que el problema se vuelve complejo al abordar las adaptaciones del modelo del vehículo aéreo autónomo y las adaptaciones del protocolo de comunicación. En el Capítulo 3 se detallará la metodología utilizada para la adaptación del prototipo del vehículo aéreo autónomo empleado. En el Capítulo 4 se presentará y analizará el protocolo de comunicación de los radios, así como sus adaptaciones y la implementación de la misma. En el Capítulo 5 expondrá los resultados obtenidos en las pruebas realizadas en el trabajo de campo. Finalmente en el Capítulo 6, las conclusiones de esta investigación, junto con las recomendaciones para futuros trabajos en este campo.



---

# Planteamiento del problema

---

## 2.1. Problema

Partiremos de lo más general del problema desde un planteamiento, Figura (2.1), en donde  $A$  quiere comunicarse con  $B$ , pero la distancia  $d$  es tan grande que no permite comunicación directa, por lo cual incorporamos a  $C$  y lo colocamos en el punto medio entre  $A$  y  $B$  para retransmitir la información. Pero la distancia es tal que las comunicaciones de  $C$  no se pueden comunicar con  $A$  y  $B$  al mismo tiempo, por lo que es necesario que  $C$  se mueva entre  $A$  y  $B$  para poder comunicarlos, donde  $A$  y  $B$  son fijos.

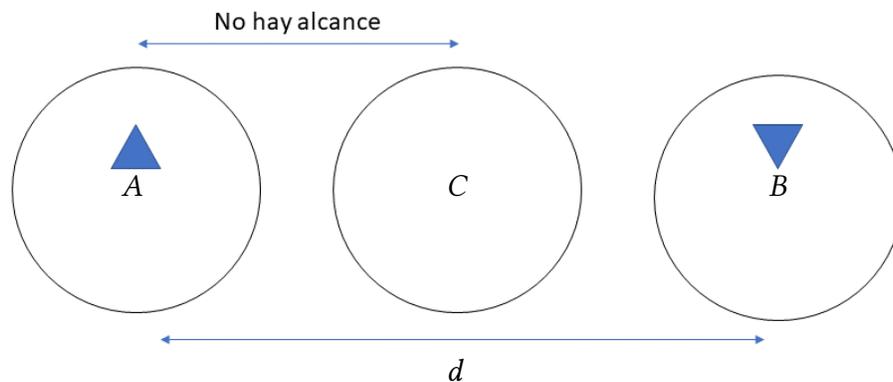


Figura 2.1: Escenario de  $A$ ,  $B$  y  $C$ .

Las características de  $C$  son tal que debe mantenerse en el alcance de  $A$  de tal manera que puedan comunicarse, y posteriormente moverse sin comunicación alguna hasta la posición de  $B$ , en donde podrá comunicarse de nuevo.

## 2.2. Aspectos básicos de comunicación

### 2.2.1. Modos de transmisión

¿Qué significa que  $A$  quiere “comunicarse” con  $B$ ? significa que  $A$  quiere emitir un mensaje para nuestro receptor  $B$ . Tenemos nuestros tres elementos esenciales para la comunicación,  $A$ , el emisor,  $B$ , el receptor y por último el mensaje que se quiere enviar. Cabe mencionar que si  $B$  quiere darle respuesta a  $A$ , entonces ahora el emisor es  $B$  y  $A$  pasa a ser el receptor, por lo que estos roles no son fijos en todos los casos.

#### Transmisión *simplex*

Se conoce como transmisión modo “*simplex*” (Figura 2.2a) en el caso que solo  $A$  (el chico) emita mensajes y  $B$  (la chica) solo los reciba, es decir,  $A$  permanece emisor en todo momento, así como  $B$  receptor. También denominada unidireccional, es una transmisión única, en una sola dirección.

#### Transmisión *half-duplex*

En la transmisión modo “*half-duplex*” (Figura 2.2b) se tiene que  $A$  (el chico) emita mensajes a  $B$  (la chica), y luego  $B$  después de recibirlos conteste y pase a ser el emisor y  $A$  el receptor, es decir  $A$  y  $B$  se alternan entre emisor y receptor. Los datos circulan en una sola dirección por vez. El canal de comunicaciones permite alternar la transmisión en dos direcciones, pero no en ambas direcciones simultáneamente.

#### Transmisión *full-duplex*

En la transmisión modo “*full-duplex*” (Figura 2.2c) se tiene que  $A$  (el chico) y  $B$  (la chica) emiten mensajes entre ellos simultáneamente, es decir tanto  $A$  como  $B$  son emisores y receptores en todo momento. A pesar de que los datos circulan en ambas direcciones a la vez, se hace uso del mismo canal, por lo que se considera que este modo de transmisión es el que mejor lo aprovecha en términos de rendimiento para protocolos de comunicación más complejos. [9]

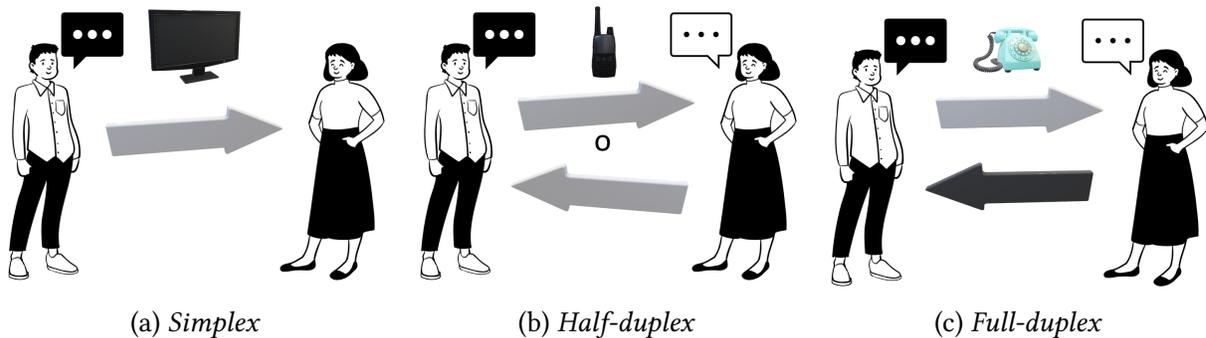
**Tipos de transmisión:**

Figura 2.2: Tipos de transmisión en el canal.

En la Figura 2.2 se muestran algunos ejemplos: Para el caso de la Figura 2.2a, un ejemplo es la televisión, ya que solo hay un canal y no podemos contestar a quién transmite. Para el caso de la Figura 2.2b, un ejemplo es un *Walkie-Talkie*, ya que es necesario liberar el canal para permitir una respuesta. Para el caso de la Figura 2.2c, un ejemplo es el teléfono, ya que se puede hablar al mismo tiempo sin necesidad de liberar un canal, aunque esto último no sería bien visto.

Sin embargo, en general, no hemos entrado en detalle sobre los mecanismos para transmitir información y de que el receptor detecte la presencia de cualquier error en la transmisión. Cuando un error en la transmisión es detectado (el error puede tratarse incluso de 1 bit), entonces todo el mensaje será descartado. Este tipo de esquema se conoce como **transmisión de mejor esfuerzo**. Detallaremos otras estrategias que tenemos para maximizar la probabilidad de que el mensaje llegue correctamente como se muestra en la Sección 5.4 y se proporcionará de forma general en el Anexo A.3.

Los modos de transmisión dependen del canal de comunicación, y a la vez del *hardware*. En nuestro caso, debido a los radios que usamos, la comunicación es “*half-duplex*”.

Esto limitará los protocolos de comunicación que se podrán emplear, sin embargo con *half-duplex* tenemos una comunicación fiable debido a los protocolos empleados, pero más lentas que las transmisiones *full-duplex*, aunque con menor requerimiento de procesamiento. Para compensar el tiempo de transmisión se usarán “banderas de transmisión” que se explicarán para la misión.

### 2.2.2. Envío de información

Aún no se ha definido “comunicación”. Se ha mencionado que hay tres elementos importantes que forman parte de la comunicación, el emisor, el receptor, y nos falta hablar del mensaje. ¿Qué es el mensaje? entendemos por mensaje, en el caso de redes de comunicación informática, que es un conjunto de símbolos enviados por el emisor que, al ser interpretados de forma apropiada por el receptor, éstos tienen un significado.

Estos símbolos pueden ser elementos de algún alfabeto o incluso números. Cabe aclarar que por “mensaje” no nos referimos a un “mensaje de texto”, ejemplo Figura 2.3, sino al conjunto de símbolos que tendrán que ser interpretados por el receptor. Un mensaje puede ser la instrucción para registrar un dispositivo a una red, solicitar una ubicación, o incluso mandar un mensaje de texto.

1 1 1 0 1 0 1 1

Figura 2.3: Ejemplo de mensaje.

Hasta ahora el problema que se presenta es la distancia  $d$  y el alcance de la transmisión. Es aquí donde entra  $C$ , el cual es nuestro sistema de comunicación móvil. Ya que, a diferencia de  $A$  y  $B$ , este puede desplazarse, lo que lo vuelve el portador del mensaje. Considerando que la distancia  $d$  puede ser cubierta por el desplazamiento de  $C$  un número suficiente de veces para transmitir los mensajes necesarios entre  $A$  y  $B$ , entonces el problema de la distancia  $d$  y el alcance de la transmisión queda resuelto, pero surgen otros problemas que se plantean a continuación.

### 2.2.3. Redes de comunicación

Hemos aclarado que  $A$  y  $B$  quieren comunicarse, y lo pueden hacer a través de la ayuda de  $C$ . Pero ¿qué son  $A$ ,  $B$  y  $C$ ? En nuestro caso son dispositivos de red conectados a una PC, en donde  $A$  y  $B$  comparten las mismas características ya que son fijos, y  $C$  debe ser ligero ya que requiere desplazarse por el aire. Al ser  $A$  y  $B$  fijos también se les puede llamar “Estaciones Base”, y a  $C$  “Estación Móvil” ya que se deslaza entre las estaciones base.

Se observa en la Figura 2.4, con el fin de ejemplificar las estaciones, un modelo represen-

tativo; más adelante en el trabajo de tesis se muestran imágenes de los modelos reales. Estos dispositivos están conectados a un radio, por lo que es posible conectarse a una red inalámbrica en la cual cada dispositivo se identifica y se registra, lo cual permite que las PC se conecten para intercambiar información y/o compartir hardware y software con los dispositivos de la misma red que tengan identificados (siempre y cuando se conozca la dirección).



Figura 2.4: Modelos de estaciones.

Una de las características que poseen estos dispositivos de radio, es que pueden formar una red de comunicaciones con tan solo estar encendidos [10]. Sin embargo es necesario registrarse en la red, y para ello es requerido que todos los dispositivos de la red se comuniquen de la misma manera. Esto es equivalente a dos personas hablando, en donde podrían hablar en dos idiomas distintos, pero eso no quiere decir que entiendan. Por otro lado, si ambas personas hablan el mismo idioma se pueden entender y comunicar apropiadamente. En el caso de los dispositivos de red a esto se le llama “protocolos de comunicación”.

Sin embargo, es importante recalcar que nuestros radios se comunican a través del espectro de radiofrecuencia, lo que puede generar errores que se pueden minimizar con nuestro protocolo de comunicación.

#### 2.2.4. Radio frecuencia

La radiofrecuencia (RF) es la tasa de oscilación de una corriente, de una tensión eléctrica alterna, de un campo magnético, eléctrico o electromagnético en el intervalo de frecuencias comprendido entre unos 20 kHz y unos 300 GHz [11]. Este intervalo se sitúa aproximadamen-

## 2. Planteamiento del problema

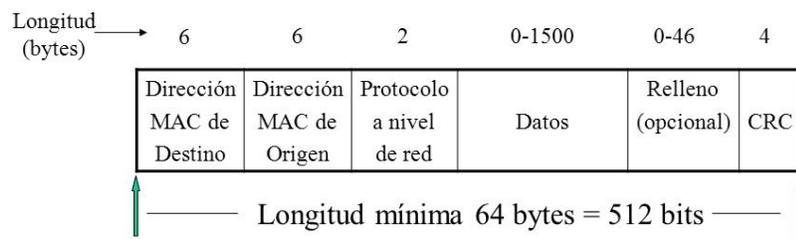
---

te entre el límite superior de las frecuencias de audio y el límite inferior de las frecuencias infrarrojas [12] [13], y engloba las frecuencias a las que la energía de una corriente oscilante puede irradiarse desde un conductor al espacio en forma de ondas de radio. Diferentes fuentes especifican diferentes límites superior e inferior de frecuencias de operación.

### 2.2.5. Protocolos de comunicación

Los protocolos de comunicación son descripciones formales de los formatos y reglas de los mensajes digitales (ver Figura 2.5). Son necesarios para intercambiar mensajes en o entre sistemas informáticos. Los protocolos de comunicación son importantes en los sistemas de telecomunicaciones y otros sistemas porque crean coherencia y universalidad para el envío y la recepción de mensajes.

#### Estructura de trama Ethernet DIX



MAC Destino-Origen: Direcciones de 6 bytes

Protocolo (Ethernet): Especifica protocolo a nivel de red, Ej. para IP X'0800'

Relleno: Garantiza que la trama nunca tenga menos de 64 bytes

CRC: Cyclic Redundancy Check. Detecta errores de transmisión

Figura 2.5: Ejemplo de la estructura de una trama.

Los protocolos de comunicación pueden abarcar la autenticación, la detección y corrección de errores y la señalización. También pueden describir la sintaxis, la semántica y la sincronización de las comunicaciones analógicas y digitales.

Existen muchos protocolos de comunicación que se utilizan en todas las comunicaciones analógicas y digitales. Las redes informáticas no pueden existir sin ellos. Para más información se puede consultar [9].

## 3.1. Prototipo de un vehículo aéreo

### 3.1.1. Estructura del prototipo

Antes de comenzar a hablar del prototipo, clasificaremos el tipo de vehículo aéreo que estamos empleando. Es un *UAV*<sup>1</sup> de ala rotativa de tamaño pequeño (alrededor de 60 cm de diámetro), que cuenta con cuatro motores eléctricos<sup>2</sup>. A este tipo de vehículos aéreos se les conoce como “Cuadricópteros”. También por su categoría y características de tamaño se les suele llamar “*Drones*”. Dicho esto, en el presente trabajo se usarán los términos *UAV*, vehículo aéreo, *drone* o cuadricóptero de forma indistinta, para referirnos al prototipo empleado, aunque este último será el más empleado en este capítulo.

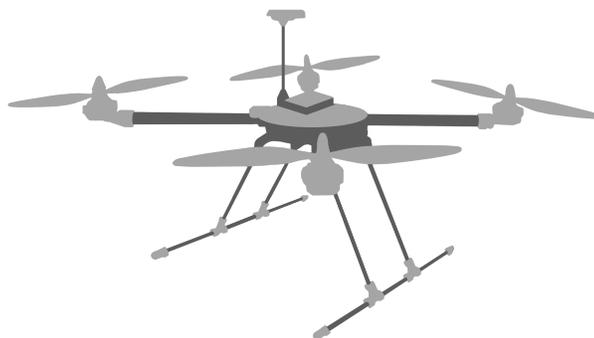


Figura 3.1: Primer prototipo.

---

<sup>1</sup>*UAV: Unmanned Aerial Vehicle*, por sus siglas en inglés para Vehículo Aéreo No-tripulado.

<sup>2</sup>No existe una clasificación oficial respecto a estos vehículos, ya que esta puede variar según el país o el uso del mismo.

Inicialmente se utilizó el cuadricóptero que se muestra en la Figura 3.1 que fue construido en [14] con otro propósito, por lo que el prototipo debe adaptarse a nuestra misión. El modelo dinámico del prototipo está en el Anexo A.1.

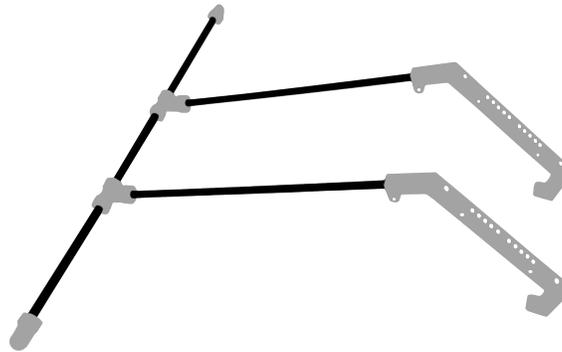
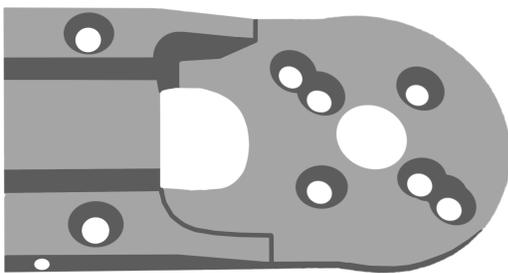


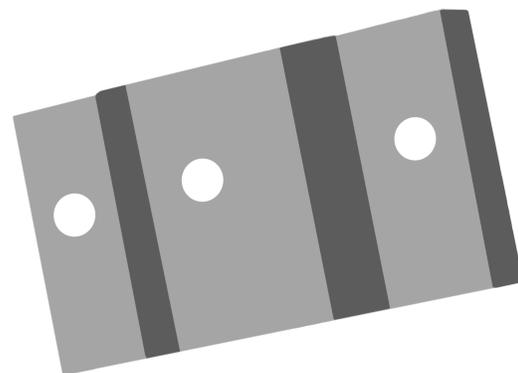
Figura 3.2: Tren de fibra de carbono x2.

El prototipo estaba constituido de la siguiente manera:

1. Tren de aterrizaje. Para mantener la batería lejos del piso, el prototipo contaba con un tren de aterrizaje largo de fibra de carbono (ver Figura 3.2).



(a) Base del motor x4.



(b) Sujetador de base x4.

Figura 3.3: Base del motor completa.

2. Soporte de motores. Los motores se fijaron en su base usando las piezas al revés para que el cable salga directo al centro del vehículo y se usaron los sujetadores para fijar las 4 bases de motor a los brazos, usando ambas piezas que se muestran en la Figura 3.3. Para tener un modelo más ligero se utilizó el tren de aterrizaje de la Figura 3.2. Con respecto al sujetador de base (ver Figura 3.3b), se emplea un tornillo ajustable para apretar los brazos a los que se sujeta.

Esta pieza es clave para mantener a los motores en posición, ya sea para evitar que estos se inclinen o que se separen del prototipo.

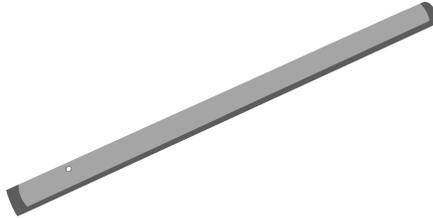


Figura 3.4: Brazos del cuadricóptero x4.

3. Brazos del cuadricóptero. Las bases de los motores se fijan a los brazos que llevan al centro del cuerpo del vehículo aéreo como se muestra en la Figura 3.4. Un extremo del brazo es sujetado por la base del motor y el otro va a la base amarilla que se ubica en el centro del vehículo aéreo (ver Figura 3.5)

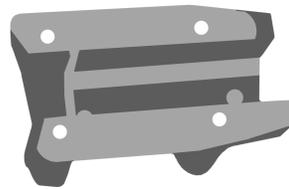
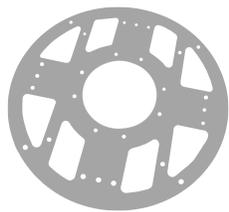
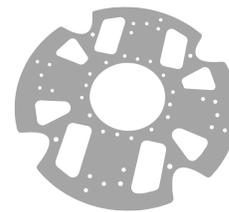


Figura 3.5: Base del brazo x4.

4. Cuerpo del cuadricóptero. La base de los brazos, que se muestra en la Figura 3.5, une el centro inferior y superior del cuadricóptero con los brazos.



(a) Centro superior.



(b) Centro inferior.

Figura 3.6: Centro del cuerpo.

Las dos piezas del centro (ver Figura 3.6) mantienen en su posición a las cuatro bases de los

### 3. Vehículo aéreo

---

brazos, las cuales son utilizadas para tener un centro rígido. Todas estas piezas son materiales de *3D Robotics*<sup>3</sup>.

5. Sistema de propulsión. El cuadricóptero es un vehículo aéreo con cuatro rotores, constituido por: (i) cuatro motores *brushless*, que brindan la potencia requerida, éstos son motores 4220-880 Kv de 16 polos (ver Figura 3.7a)<sup>4</sup> y (ii) cuatro hélices de 11x4.5 pulgadas, que generan la sustentación (ver Figura 3.7b).



(a) Motor *brushless* x4.



(b) Hélices.

Figura 3.7: Cuatro Rotores.

6. Sistema de alimentación. La energía de los rotores de la Figura 3.7, es suministrada por una batería LiPo con capacidad de 4000 mAh de tres celdas y velocidad de descarga de 45C (ver Figura 3.8). El amperaje de descarga de la batería es de 180A.



Figura 3.8: Batería LiPo de 3 celdas.

7. Sensores externos. Entre los sensores externos contamos con un GPS Ublox Neo-M8N de la marca 3DR.

8. Sistema de control. El sistema de control del cuadricóptero está dado por una *3DR Pixhawk 1 Flight Controller* (ver Figura 3.9)<sup>5</sup>. El 3DR Pixhawk® 1 es un controlador popular de vuelo de propósito general basado en el diseño de hardware abierto *Pixhawk-project*, esto combinado con

---

<sup>3</sup> *3D Robotics* es una empresa pionera en la creación de vehículos aéreos no tripulados (UAV). En 2017 ha dejado la manufactura de *drones* dedicándose sólo al desarrollo del software. Sitio web: <https://www.kittyhawk.aero/>

<sup>4</sup> Imagen tomada de: <https://hobbyking.com/>

<sup>5</sup> Imágenes tomadas de: <https://ardupilot.org/>

Ardupilot, un *Autopilot Software Suite* de código abierto para vehículos no tripulados, capaz de controlar de forma autónoma.



(a)



(b)

Figura 3.9: Pixhawk 1 (a) y Ardupilot *Software* (b).

El prototipo inicial era deficiente respecto a su estructura y balance, por lo que se realizaron algunas adaptaciones para resolverlas, que se muestran enseguida.

## 3.2. Adaptaciones del prototipo

Para adaptar al prototipo original, tenemos que tener en cuenta a la misión deseada. Se modifica al vehículo aéreo autónomo para llevar a cabo un establecimiento de conectividad en redes de comunicaciones. Además del enfoque de la misión, se aprovechan las modificaciones para tener un modelo con una estructura más robusta que el prototipo original.

Se comenzó por detectar las partes frágiles del prototipo, las cuales tras diferentes vuelos de prueba se determinó que en caso de un accidente, las partes que principalmente sufrían daños son el tren de aterrizaje, las hélices y el soporte del GPS. También se determinó que la estructura original tenía gran parte del centro ocupada por las bases de los brazos del cuadricóptero, lo que dificultaba ocultar en la zona central cualquier tipo de cable o equipo.

Dentro de las adaptaciones requeridas además de las modificaciones, se debe añadir una *Raspberry 3* modelo B, en la que se implementarán los módulos de software requeridos para lograr comunicación extremo a extremo, lo que se tratará más adelante en este capítulo, así como también un radio que permitirá al cuadricóptero comunicarse con las bases. Estas adaptaciones se realizaron evitando cambiar la configuración del vehículo.

Originalmente se diseñaron dos carcasas, una para el radio, Figura 3.10, y otra para la *Raspberry 3* Modelo B, Figura 3.11, en la parte inferior se encuentran ranuras para enfriamiento, los sujetadores en ambos casos se trataban de 4 puntos pero se descartó por el tipo de base a donde

### 3. Vehículo aéreo

---

debían sujetarse. Ambos cuentan con el logo del cinvestav en la parte superior. Este diseño se consideró para mantener a ambos dispositivos separados.

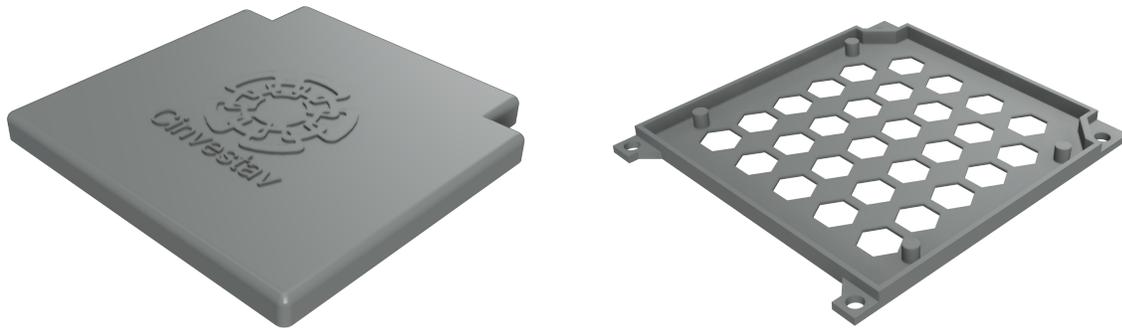


Figura 3.10: Carcasa superior radio v1 (a) y Carcasa inferior radio v1 (b).

En el caso de la carcasa de la *Raspberry 3*, Figura 3.11, no tenía compartimientos de salida para las conexiones que se necesitan realizar, y añadirlos comprometía la estructura. Por otro lado, el mecanismo era solo una pieza en la cual solamente era necesario deslizar la *Raspberry*, lo que hacía su funcionamiento poco práctico.

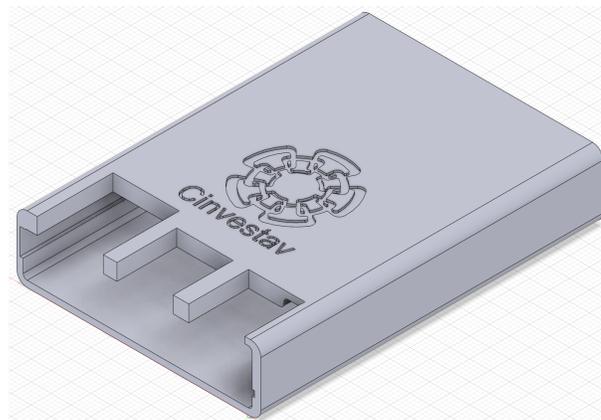


Figura 3.11: Carcasa *Raspberry* v1.

Posteriormente, se diseñó una base para colocar el *Pixhawk 1*, Figura 3.12, por encima de estas dos carcasas, esto se pondría con cinta doble cara para mantenerlo sujeto a la superficie.

Éstas últimas tres piezas resultaban en una estructura robusta, se tomaron como referencia de otros prototipos<sup>6</sup> pero al llevar acabo otro tipo de tareas, no resultaron tan convenientes para la misión a realizar, por lo que se realizó un rediseño de la estructura principal.

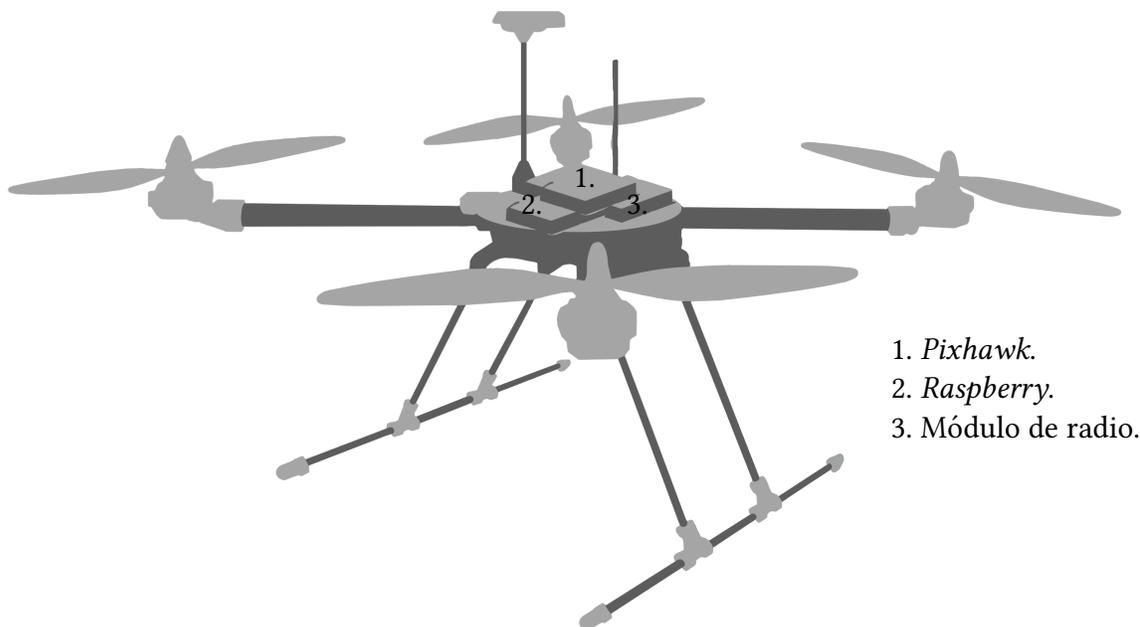
---

<sup>6</sup>Carcasas tomadas de: <https://sketchfab.com/features/free-3d-models>

Figura 3.12: Base para *Pixhawk* v1.

### 3.2.1. Construcción de piezas

Para la construcción de piezas se tomaron en cuenta las adaptaciones, una de las más importantes gira en torno a la batería. Se busca crear compartimientos en el centro del vehículo para albergar tanto la *Raspberry* como el radio. Al aumentar el peso con los nuevos componentes también aumentamos el impacto que recibe el tren de aterrizaje el cual ya era frágil. Al tener varios componentes escalados de forma vertical en el vehículo, también se genera un momento que afecta la maniobrabilidad del mismo como se muestra en la Figura 3.13.



1. *Pixhawk*.
2. *Raspberry*.
3. Módulo de radio.

Figura 3.13: Componentes apilados al centro.

### 3. Vehículo aéreo

---

Este peso adicional combinado con la fragilidad del tren de aterrizaje expone la batería debajo del centro del cuerpo del vehículo, ya que al romperse el tren de aterrizaje lo siguiente en impactar será la batería. Tomando en cuenta estos puntos, se cambió el tren de aterrizaje a uno más resiliente. Este nuevo tren de aterrizaje baja el centro de gravedad y por lo tanto deja aún más expuesta la batería, por lo que se opta diseñar un nuevo centro del vehículo en torno a ella.

El marco del *UAV* está conformado de tres partes importantes: tren de aterrizaje, compartimiento de batería y compartimiento de electrónica; todos estos elementos están organizados dentro del marco del cuadricóptero con distribución “X”.

#### **Tren de aterrizaje y compartimiento para motores**

En la Figura 3.14 tenemos una vista del cuadricóptero donde se aprecia tanto el tren de aterrizaje como los compartimientos para motor que forman parte de la misma estructura del tren. Al tener una composición en “X”, estas partes se encuentran al final de cada extremidad por lo que tenemos asignados en estas áreas los motores 1, 2, 3 y 4.



Figura 3.14: Tren de aterrizaje.

#### **Compartimiento de batería y compartimiento de electrónica**

Esta base se diseñó con el propósito de proteger la batería ante cualquier impacto (ver Figura 3.15), así como agregar cierta flexibilidad para amortiguar los aterrizajes. Todo esto se logra sin perder la rigidez por el diseño especial de la base, la cual cuenta con un refuerzo a lo largo de la estructura. La pieza que va por encima de la batería va separada de los brazos de los motores para evitar los momentos que se producen por los mismos, pero esto se compensa con piezas de fibra de carbono, las cuales sirven de base para los controladores de velocidad de cada motor.

Esto nos da el compartimiento de la batería, y espacio para montar la electrónica protegidas con piezas de material PLA.

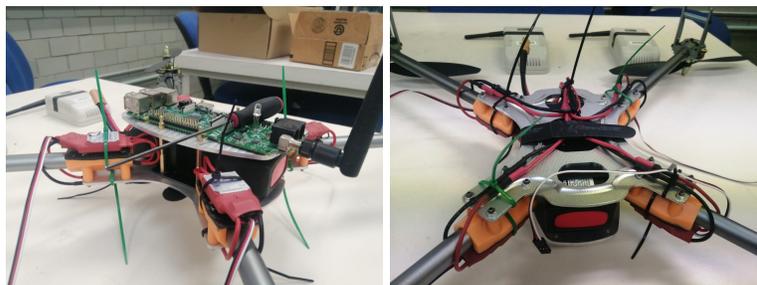


Figura 3.15: Compartimientos de batería y electrónica.

Para el compartimiento de electrónica se diseñaron dos piezas, una base donde va montada la *Raspberry* Modelo B y una tapa donde va montado el radio. Se optó por colocar el radio de cabeza para ahorrar espacio en el nuevo centro. Antes de colocar el compartimiento de electrónica analizaremos el esquema general de todos los componentes que ya posee. Además de incorporar la *Raspberry* y el radio, se tiene también el *Pixhawk* y los componentes del GPS, así como el manejo de cables.

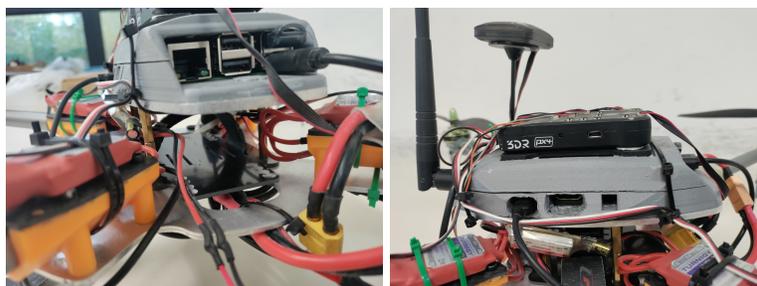


Figura 3.16: Base de aluminio.

En el diseño de estas dos piezas (ver Figura 3.17) se consideró el espacio para enfriamiento y el manejo de cables; esta nueva pieza sigue el diseño de la base donde se encuentra ubicada la batería. Esta pieza es un elemento muy importante junto con sus piezas similares de aluminio, ya que permite a los elementos ensamblados tener una distribución adecuada del peso, permitiendo al cuadricóptero tener un centro de masa ubicado cerca del centro geométrico, por lo que los momentos que genera al moverse en cualquier dirección son similares, lo que ayuda a su estabilidad.

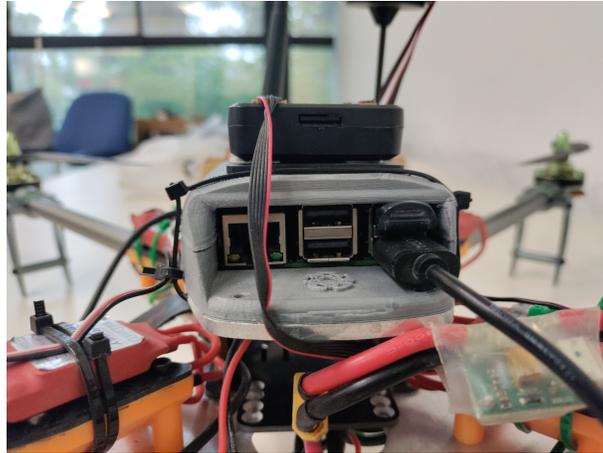


Figura 3.17: Compartimiento de electrónica.

El esquema general de conexión del cuadricóptero se muestra en la Figura 3.18.

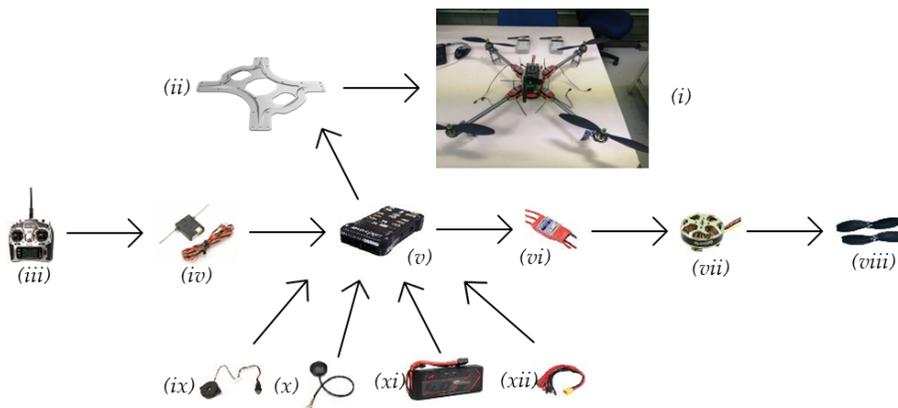


Figura 3.18: Diagrama de conexión del vehículo aéreo. (i) Cuadricóptero, (ii) base de estructura de aluminio, (iii) transmisor DX7s, (iv) receptor DSMX, (v) Pixhawk, (vi) controladores electrónicos de velocidad ( $\times 4$ ), (vii) motores 4220-880Kv de 16 polos ( $\times 4$ ), (viii) hélices 11x4.5 ( $\times 4$ ), (ix) zumbador/interruptor de seguridad, (x) GPS Ublox Neo-M8N, (xi) batería LiPo 4000 mAh 3S, (xii) arnés de distribución.

El peso del vehículo aéreo (ver Figura 3.19) es importante para el recorrido ya que a mayor peso incrementará la potencia de los motores, por lo que el consumo de energía de la batería será mayor y por lo tanto tendremos menos tiempo de vuelo, el cual es crucial para la misión por lo tiempos de espera de envíos de paquetes como se muestra más adelante en el Capítulo 5.



(a) Peso del *Drone* sin batería (1.487 kg).



(b) Peso de la batería (0.375 kg).

Figura 3.19: Peso total del *Drone* (1.862 kg).

Como se puede ver en la Figura 3.19, el peso total del *Drone* está por debajo de los 2 kg por lo que no requiere autorización ni permiso de la Dirección General de Aeronáutica Civil en México. [15]

### 3.3. Sistemas de transporte y comunicación

El capítulo anterior habla sobre la misión que se debe llevar a cabo, en donde se mencionan tres elementos. La estación base *A*, la estación base *B* y el cuadricóptero mencionado como el elemento *C*. En este capítulo se habla sobre el cuadricóptero, el cual es el prototipo adaptado que se emplea para llevar a cabo la misión. Este prototipo, a diferencia de las estaciones base, realiza dos funciones al mismo tiempo: volar y transmitir información.

A pesar de ser un solo vehículo que lleva a cabo la misión, el prototipo se divide en dos sistemas independientes: el sistema de transporte y el sistema de comunicaciones, los cuales se presentan a continuación.

### 3.3.1. Sistema de transporte

El sistema de transporte consiste en el uso del cuadricóptero como vehículo de transporte para el sistema de comunicaciones. Dada su independencia, el cerebro del sistema de transporte, el *Pixhawk 1* de la Figura 3.20, es aquel que se encarga de determinar qué modos de vuelo (Sección 3.4) entran en función. A grandes rasgos el prototipo tiene dos requerimientos esenciales: transferir información y moverse. El sistema de transporte se encarga del movimiento.



Figura 3.20: *Pixhawk 1 3DR*

Se consideran sistemas independientes entre sí porque no hay una relación directa entre ellos, y ambos pueden realizar su función correctamente sin el otro. Sin embargo, para los requerimientos de la misión del prototipo es requerido combinar las funciones de ambos sistemas, ya que uno se encarga del movimiento pero es necesario transferir información. El sistema de comunicaciones se explica a continuación.

### 3.3.2. Sistema de comunicaciones

El sistema de comunicaciones es el que determina cuando se completa la misión. Ambos sistemas van montados en el cuadricóptero, por lo que es equivalente a que éste vuele con dos cerebros, uno enfocado en los modos de vuelo y la estabilidad (*Pixhawk 1*); y el otro en la transferencia de información y su proximidad con las estaciones base, este último es la Raspberry 3 Modelo B, Figura 3.21<sup>7</sup>.

---

<sup>7</sup>Imágenes tomadas de: <https://www.raspberrypi.com/>

Figura 3.21: *Raspberry Pi 3 Modelo B.*

## 3.4. Modos de vuelo

Los modos de vuelo del cuadricóptero dependen del *software* instalado en el *Pixhawk*. Dos de las opciones más populares son *Ardupilot* y *PX4*. La primera es la opción que se eligió debido a un modo de vuelo llamado “*Autotune*”. Este modo de vuelo sirve para obtener las ganancias PID del control de vuelo de forma automática. Se aplica al primer vuelo del cuadricóptero mientras se encuentra descalibrado.

### 3.4.1. Modo *Autotune*

El trabajo de tesis considera que tenemos un vehículo aéreo apto para volar. Sin embargo, al realizar cambios en el fuselaje del cuadricóptero es necesario volver a calibrar las ganancias PID debido a un cambio en la masa y forma geométrica del vehículo (ver Figura 3.22).

Las ganancias PID son las constantes del control Proporcional, Integral y Derivativo; estas ganancias nos permiten tener un vuelo con la respuesta esperada a la entrada de control.

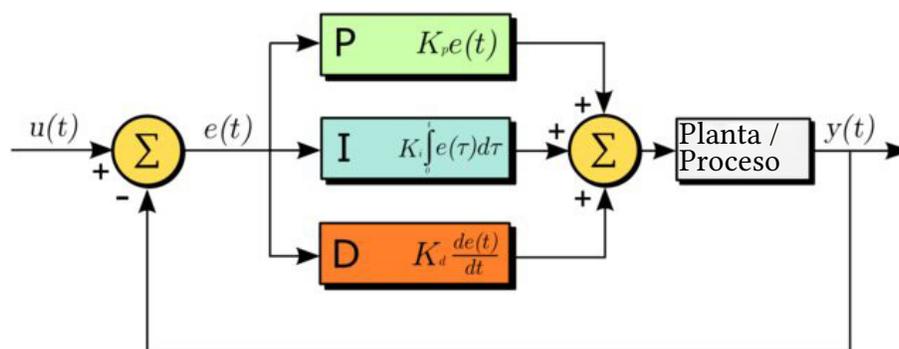


Figura 3.22: Ganancias PID.

El modo de vuelo “Autotune” intenta ajustar automáticamente la estabilización P, la velocidad I y D, y las aceleraciones rotacionales máximas para proporcionar la respuesta más alta sin un sobreimpulso significativo. El cuadricóptero debe ser “básicamente” volable en modo *AltHold* antes de intentar utilizar *Autotune*, ya que la función debe ser capaz de “mover” el cuadricóptero en los ejes de alabeo y cabeceo [16].

El progreso del *autotune* se registra en el log del *dataflash* (ver Figura 3.23<sup>8</sup>):

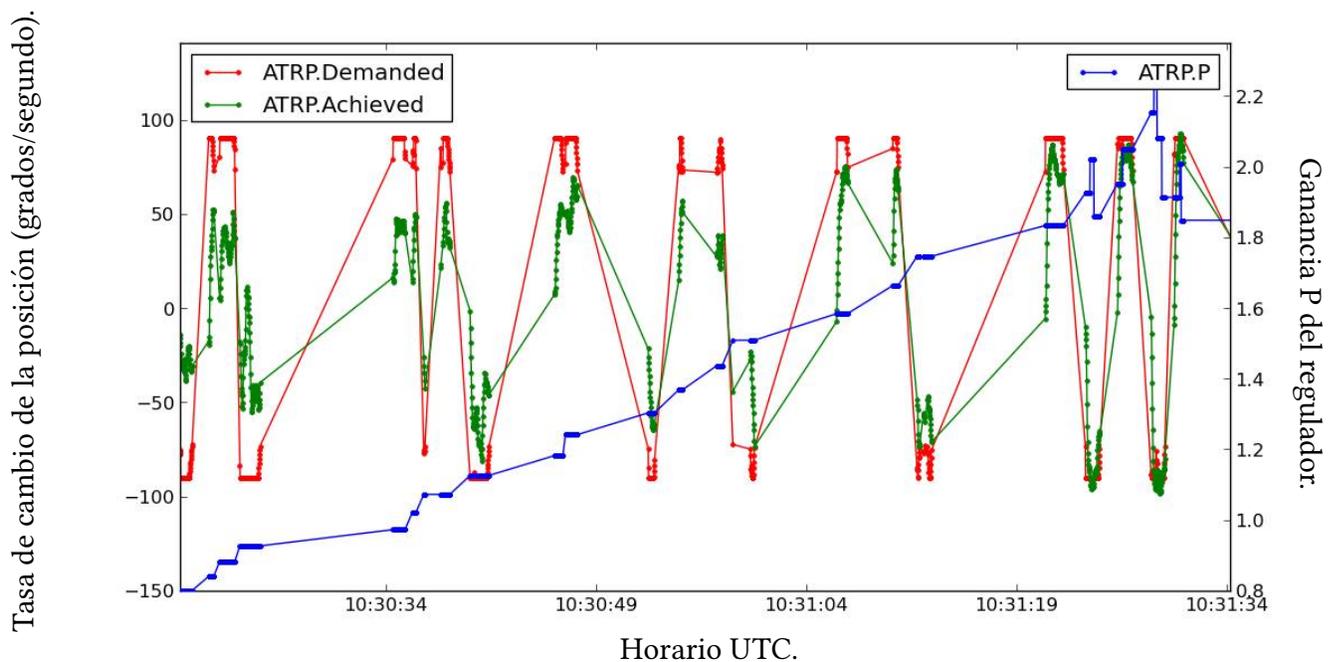


Figura 3.23: Gráfica de “Autotune” de datos generados en *Dataflash log*.

El mensaje ATRP tiene un campo “Tipo” que muestra qué tipo de autoajuste se está grabando. Un valor de Tipo de 0 es para sintonización de balanceo, y un valor de 1 es para sintonización de tono. Cuando grafique los resultados de ATRP debe elegir datos con un Tipo de 0 o 1 (no ambos).

El campo ATRP.Demanded es la tasa demandada de cambio de actitud (tasa de balanceo o tasa de cabeceo) en grados por segundo. El campo ATRP.Achieved es lo que la aeronave realmente alcanzó en tasa de cambio de actitud. Como se puede ver en la Figura 3.23, al inicio de la sintonización automática los valores demandados (una vez el modo de vuelo se activa se comienza el registro en el *Dataflash log*, y comienza a emitir las demandas de movimientos.)

<sup>8</sup>Imágenes tomadas de: <https://www.ardupilot.org/>

eran mucho más altos que los alcanzados, porque las ganancias de sintonización eran demasiado bajas. A medida que la sintonización progresaba los valores demandados y alcanzados empezaron a converger. La línea azul en la Figura 3.23 es el valor ATRP.P, que es la ganancia P para el controlador. Se puede ver que aumentó de 0.8 hasta un máximo de 2.2, luego cayó ligeramente a alrededor de 1.85.

### 3.4.2. Modo Manual

El modo de vuelo “*Manual*” se considera de control RC normal, sin estabilización [17]. Todas las entradas RC pasan a través de las salidas servo establecidas por su SERVOx\_FUNCTION.

Se considera que los parámetros MAN\_EXPO\_ROLL, MAN\_EXPO\_PITCH, y MAN\_EXPO\_RUDDER se aplicarán de forma exponencial a las entradas del *stick*, si no es cero, en este modo. Esto es para usuarios con transmisores que no proporcionan esta función y desean “suavizar” la sensación del *stick* alrededor del punto muerto.

Esto supone que el canal RC está siendo usado como una función de salida de servo en lugar de una seleccionada por su parámetro RCx\_OPTION. Las únicas formas en las que la entrada a una función de salida de servo puede ser diferente de las entradas en este modo es si se dispara un *failsafe*<sup>9</sup> o *geofence*<sup>10</sup> configurado, y *Plane* toma el control, el cual es un modo de vuelo en donde se deshabilita el modo manual y aterriza solo.

### 3.4.3. Modo Hover

Para llevar acabo el modo *hover* tenemos que plantear un modo de vuelo de *ardupilot*; el más cercano a ello es el modo QLOITER que intenta mantener automáticamente la posición, rumbo y altitud actuales. El piloto puede volar el cuadricóptero en modo QLOITER como si estuviera en un modo de vuelo más manual, pero cuando se sueltan los controles, el vehículo frenará hasta detenerse y mantener la posición.

Tener una buena conexión con el GPS, baja interferencia magnética en la brújula y bajas vibraciones son importantes para lograr un buen rendimiento QLOITER.<sup>11</sup>

---

<sup>9</sup>En caso un fallo detectado por el sistema de *ardupilot*, el *drone* aterrizará por seguridad.

<sup>10</sup>Se delimita una zona geográfica en el *ardupilot*, en donde si el *drone* sale de ella entonces aterrizará.

<sup>11</sup>Ambos modos de vuelo vienen desarrollados a plenitud en Ardupilot.org



# Radio comunicación

## 4.1. Comunicación extremo a extremo

En este capítulo se describirá todo el proceso de comunicación para enviar información de un punto a otro, a esto se le conoce como “comunicación extremo a extremo” ya que solo se considera el punto de envío y el punto de llegada. Es decir, solo vemos al transmisor al momento del envío y al receptor cuando recibe la información [18]. Sin embargo, al detallar todo el proceso veremos que esto implica varias consideraciones que pasan por alto al nivel de usuario.

Primero detallaremos el proceso del flujo de la información que se emplea en este trabajo, tenemos tres participantes: la estación base *A*, la estación base *B* y la estación móvil *C* (*Drone*) como se muestra en la Figura 4.1, esto en el sentido de transmitir desde la estación base *A* hacia la estación base *B*.

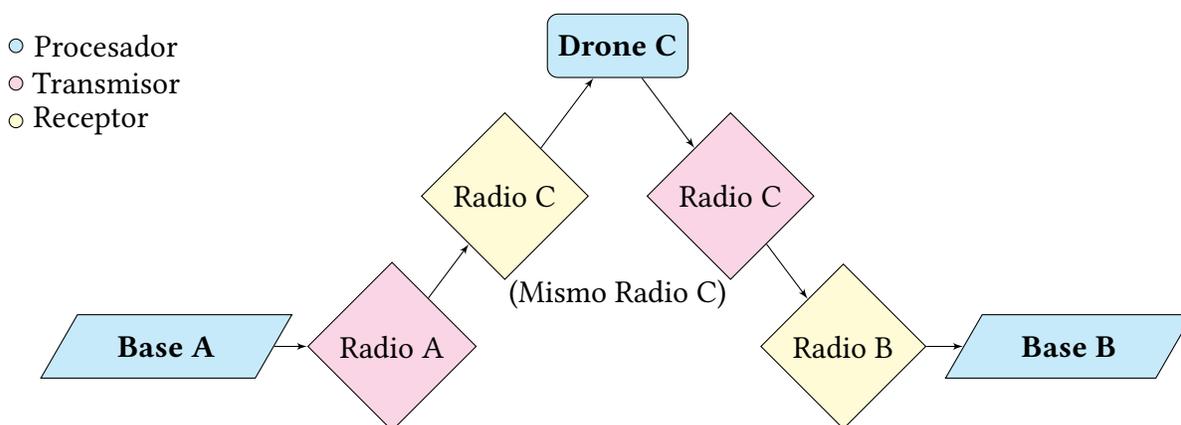


Figura 4.1: Flujo de la información.

De la Figura 4.1 observamos que la comunicación se lleva acabo de extremo a extremo desde la estación base *A* hacia la estación base *B*, y este es el objetivo que buscamos; sin embargo, el envío de la información en la Figura 4.1 considera un flujo constante de información. En nuestro caso esto no es posible debido a la distancia como se plantea en el Capitulo 2, entonces nuestra estación móvil (el *drone*) se encontrará con una estación base a la vez. Por lo que se divide flujo de la información en dos partes como se muestra en la Figura 4.2. [19]

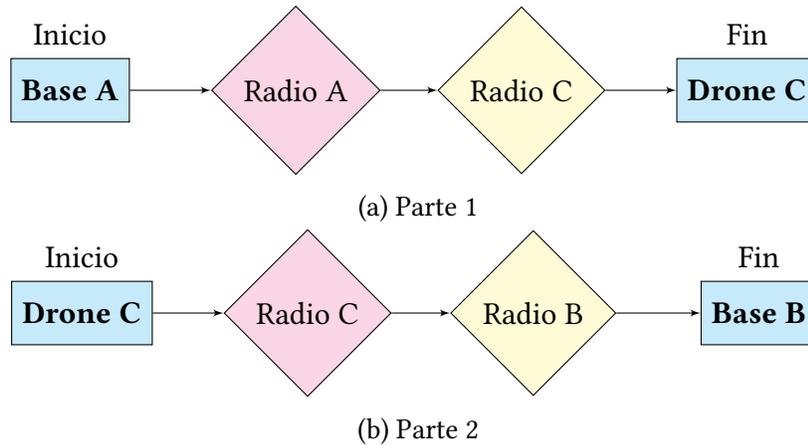


Figura 4.2: Partes del flujo de la información.

En la Figura 4.2a tenemos una comunicación extremo a extremo que va de la estación base *A* hacia el *Drone C*, la estación móvil, y otra como se muestra en la Figura 4.2b que va del *Drone C* hacia la estación base *B*. Retomando que, debido a la distancia, el *Drone C* no puede comunicarse con ambas estaciones a la vez.

De forma general podemos decir que tenemos una comunicación extremo a extremo de la estación base *A* hacia la estación base *B*, pero, en la práctica, dadas las condiciones de la misión, esto no es posible. El usuario podrá ver la información en la estación base *A* y en la estación base *B*, pero no podrá consultarla mientras se encuentre en vuelo en el *Drone C*. De aquí que el usuario solo verá los extremos de las estaciones base, por lo que simula a una sola comunicación de extremo a extremo.

En ambas partes de la Figura 4.2 contamos con cuatro elementos que participan en la transmisión para que llegue de un extremo a otro (contando el origen y el destino). A estos elementos se les conoce como “nodos” y es necesario establecer cómo se comunica cada uno de estos nodos si queremos que la información llegue al otro extremo.

## 4.2. Comunicación de Radio a PC

Como se menciona en la Sección 2.2.3, tanto las estaciones base como el *Drone* tienen una PC para procesar la información. Para poder definir la comunicación entre las estaciones base y el cuadricóptero, es necesario antes definir la comunicación entre los radios y las PC, ya que estos son los nodos que permitirán llevar a cabo la comunicación de extremo a extremo.

Los radios se comunican a través de arreglos de 8-bits, los cuales son enviados bajo un protocolo de comunicación llamado “mejor esfuerzo”[9], esto implica que si se detecta un error, éste simplemente es descartado junto con la trama y se notifica a la PC. Esto solo aplica a la comunicación de radio a PC; en el caso de radio a radio si el mensaje tiene algún error, el radio receptor lo descartará sin notificar. Sin embargo, en este trabajo se implementará un protocolo de comunicación más robusto que permita no solo descartar la información corrompida sino también recuperarla incluso cuando el radio del otro extremo decida descartarla.

Vamos a plantear una comunicación extremo a extremo entre dos PC como se muestra en la Figura 4.3. En esta sección y en la que sigue, solo hablaremos de lo que se encuentra dentro del rectángulo rojo, el nodo 1 y nodo 2 que en conjunto forman una estación base. Solo la PC y su respectivo radio. Este es el primer paso de tres para llegar a la otra PC, los otros dos serían entre el nodo 2 y nodo 3, y por último el nodo 3 con el nodo 4.

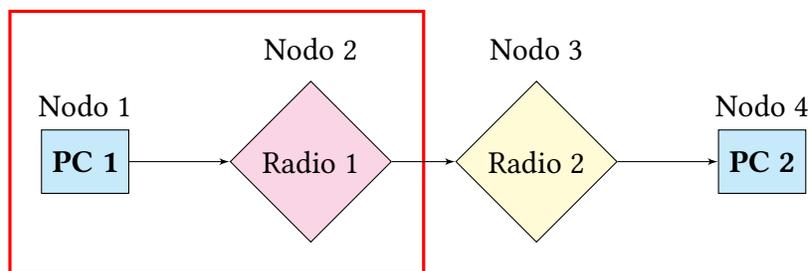


Figura 4.3: Comunicación PC a PC.

En la Figura 4.4 se muestran los medios de transmisión entre pares de nodos.

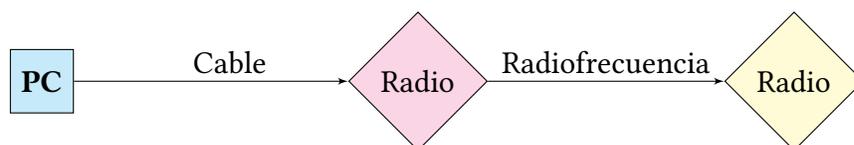


Figura 4.4: Medio de transmisión entre nodos.

### 4.2.1. Interconexión Radio a PC

Los radios poseen un puerto mini-USB. Un cable con conector USB en un extremo (para conectar la PC) y mini-USB en el otro (para conectar el radio) puede usarse para hacer la conexión. Desde la PC los radios son identificados como puertos seriales virtuales, los cuales transmiten utilizando el puerto USB, pero cuyas señales cumplen con las especificaciones lógicas de una transmisión serial. Para lograr esto, los módulos de radio poseen un puente de comunicación serial a USB basado en el circuito integrado CP2104 de *Silicon Laboratories* [10].

Los arreglos de bits enviados por el radio se envían a través de este puerto serial virtual, el cual es el medio de comunicación entre la PC y el radio.

La comunicación a través del puerto serial está siempre dada en bits, sin embargo hay varios parámetros que se deben configurar que dependen del radio. Esto es así ya que una computadora se puede configurar de múltiples maneras, pero el radio solo de una. En el caso de los radios, éstos mandan arreglos de 8-bits. Adicional a los 8-bits se pueden incluir: el bit de arranque, el bit de paridad y el bit de parada. El bit de arranque indica el inicio de los 8 bits o, dicho de otro modo, donde comienza un nuevo caracter. El bit de paridad va después del bit de parada, el cual se obtiene a partir de una operación booleana y sirve para determinar si hay un error en el arreglo. Por último tenemos el bit de parada, que indica el fin del arreglo. Estos bits solo están presentes en la comunicación serial asíncrona, lo cual quiere decir que los datos se transfieren sin el respaldo de una señal de reloj conectada entre sistemas. En el caso de los radios solo tienen el bit de arranque y parada.

Al abrir la conexión del puerto serial en la computadora, debemos definir los parámetros de configuración del mismo para poder leer lo que nos envía el radio.

Parámetro	Valor
Baud rate	57600 bits/seg
Bits de datos	8
Bits de paro	1
Paridad	No

Tabla 4.1: Parámetros de configuración del puerto serial.

En la Tabla 4.1 se muestran los parámetros con sus respectivos valores. El *Baudrate* es la velocidad de transmisión que tiene el radio, ésta está dada en unidades de *bits/seg*, lo que quiere

decir que el radio puede transmitir 57600 bits en un segundo. Los bits de datos son aquellos que poseen información y van después del bit de arranque, es decir, que después del bit de arranque se leen los siguientes 8 bits para determinar el carácter. El bit de paro es posterior a los 8-bits, y quiere decir que ahí termina la información. Y por último indicamos que no viene un bit de paridad, éste iría entre los 8-bits y el bit de paro.

#### 4.2.2. Arreglo de bits

Los arreglos enviados por el radio vienen ordenados en 8-bits, por lo que podemos interpretar la comunicación como un arreglo de arreglos de 8-bits. Un arreglo de 8-bits puede almacenar 256 valores, que van de 0 a 255. Esto es bastante práctico para los radios porque en hexadecimal con solo dos caracteres se cubre la misma cantidad de valores que van de 0 a 255. Siendo en hexadecimal de  $0x00^1$  a  $0xFF$ , por lo que el radio transfiere datos en hexadecimal convertidos a binario a través de arreglos de 8-bits. Esto se debe a que es más útil usar una notación con números y algunas letras a escribir todo en binario, además de la simplificación en la notación. Un ejemplo de ello sería la PC con dirección  $0xF001$ , que en binario sería  $0b1111\ 0000\ 0000\ 0001^2$ , en donde podemos referirnos a una computadora con cuatro caracteres en lugar de dieciséis.

Esto es usado de forma general en la comunicación por puerto serial como se muestra en el ejemplo de la Figura 4.5 en donde se usa entre una PC y una impresora.

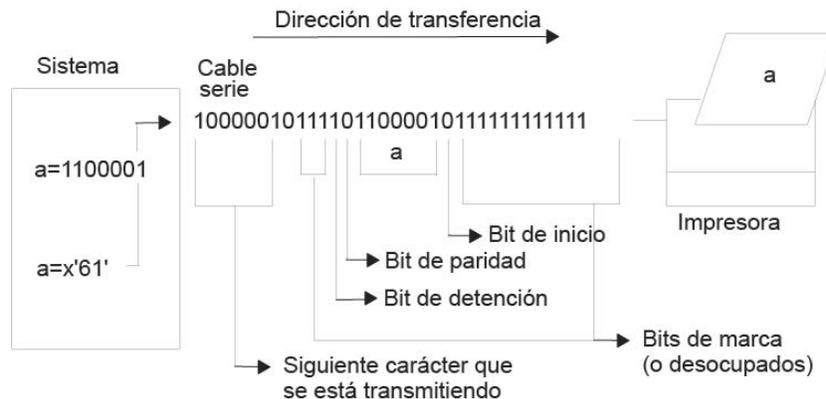


Figura 4.5: Comunicación por el puerto serial de una PC a una impresora.

<sup>1</sup>La notación  $0x$  previa a un número se utiliza para referir que el número se encuentra en sistema hexadecimal.

<sup>2</sup>La notación  $0b$  previa a un número se utiliza para referir que el número se encuentra en sistema binario.

### 4.3. Comunicación de PC a radio

En la sección anterior se habla del puerto serial y los arreglos de 8-bit que nos envía el radio. Sin embargo, a pesar de conectar el cable USB, no podremos darle casi ninguna instrucción o solicitar ningún tipo de información. Esto se debe a que no estamos registrados en la red, y para que el radio no descarte nuestra información es necesario registrarnos. Cuando nos conectamos al radio hay dos mensajes que nos envía periódicamente. Estos son: la lista de dispositivos y PING.

La lista de dispositivos sirve para verificar si estamos registrados en la red, si no lo estamos debemos proceder a registrarnos, de lo contrario no aceptará casi ninguna instrucción. Esto se puede hacer buscando nuestro identificador de la PC en la lista de dispositivos que nos envía. Si estamos registrados ignoramos ese mensaje y, por el contrario, si no estamos registrados debemos informarle al radio que deseamos registrarnos en la red.

En el caso de PING, el radio lo envía para saber si estamos activos: en caso de inactividad el radio nos dará de baja de la red y tendremos que realizar de nuevo el proceso de registro.

Las tramas enviadas por la PC al radio una vez que se encuentra registrada pueden ser solicitudes de información o envío de datos. En el caso de una solicitud de información, el radio nos hace saber a través de un reconocimiento que hemos solicitado información. Mientras que cuando enviamos datos, no nos da dicha información. Esto es importante al momento de plantear un protocolo de enlace de comunicación, ya que algunos protocolos hacen uso de esas tramas de “reconocimiento” en donde se indica que la información ha llegado correctamente o incluso que no llegó o llegó corrompida. También en algunas ocasiones es posible recibir un “reconocimiento” en casos de algunos otros errores, como puede ser que la trama no llegó por saturación del puerto serial.

Toda la comunicación entre la PC y el radio, en ambos sentidos, es a través de los arreglos de arreglos de 8-bits que llamamos tramas. Sin embargo, es necesario saber que estas tramas ya están previamente programadas por lo que es importante tener conocimiento tanto de su estructura como de las solicitudes o envíos de información que se pueden llevar a cabo. Dicho esto, también es posible usar la PC para reconocer tramas que no tenemos registradas; de esta misma manera se pueden enviar tramas no registradas.

Por lo que es posible hacer uso de las tramas de envío de información para crear nuevas tramas o instrucciones propias que solo las puede reconocer otra PC, es decir, lo que para el radio sería un envío de trama de información puede presentar una trama compleja solo reconocida por la PC destino que posea la información de cómo identificar su estructura.

### **4.3.1. Mantenerme en la red**

Una vez que nos registremos, podemos enviar o solicitar cualquier tipo de información deseada a través de los radios, sin embargo no debemos ignorar las tramas PING. Aunque ya hemos hablado un poco acerca de ellas, en la Sección 4.5 se explica qué es una trama y cómo reconocerla. Mientras tanto vamos a considerar a la trama PING como un mensaje que el radio envía a la PC para saber si estamos activos en la red y lo reenvía cada 6 segundos hasta reenviar una tercera trama PING, en caso de no responder la tercera trama entonces la PC es dada de baja de la red y tendrá que registrarse de nuevo. En la Figura 4.6 podemos observar el diagrama de flujo del proceso de registro y como mantenerse conectado en la red.

## **4.4. RSSI**

### **4.4.1. Definición**

Viene del inglés “*received signal strength indicator*”, que quiere decir “indicador de la potencia de señal recibida”, y en comunicaciones se suele usar este término para hablar acerca de la potencia de las señales de radio recibidas, el cual es nuestro caso. Haciendo una analogía, serían las barras de intensidad de señal de la red en nuestros teléfonos móviles.

### **4.4.2. Monitoreo del RSSI**

¿Por qué es importante solicitar esta información entre los radios? El *RSSI* nos puede dar mucha información con solo decirnos que tan buena es la potencia de señal, a esto se le conoce como la calidad del enlace (ver Figura 4.7). La calidad del enlace nos permite saber si la conexión es buena y por lo tanto anticipar errores, o al menos aquellos relacionados a la calidad del enlace. En el caso de una mala calidad, la conexión puede fallar; esto sucede debido

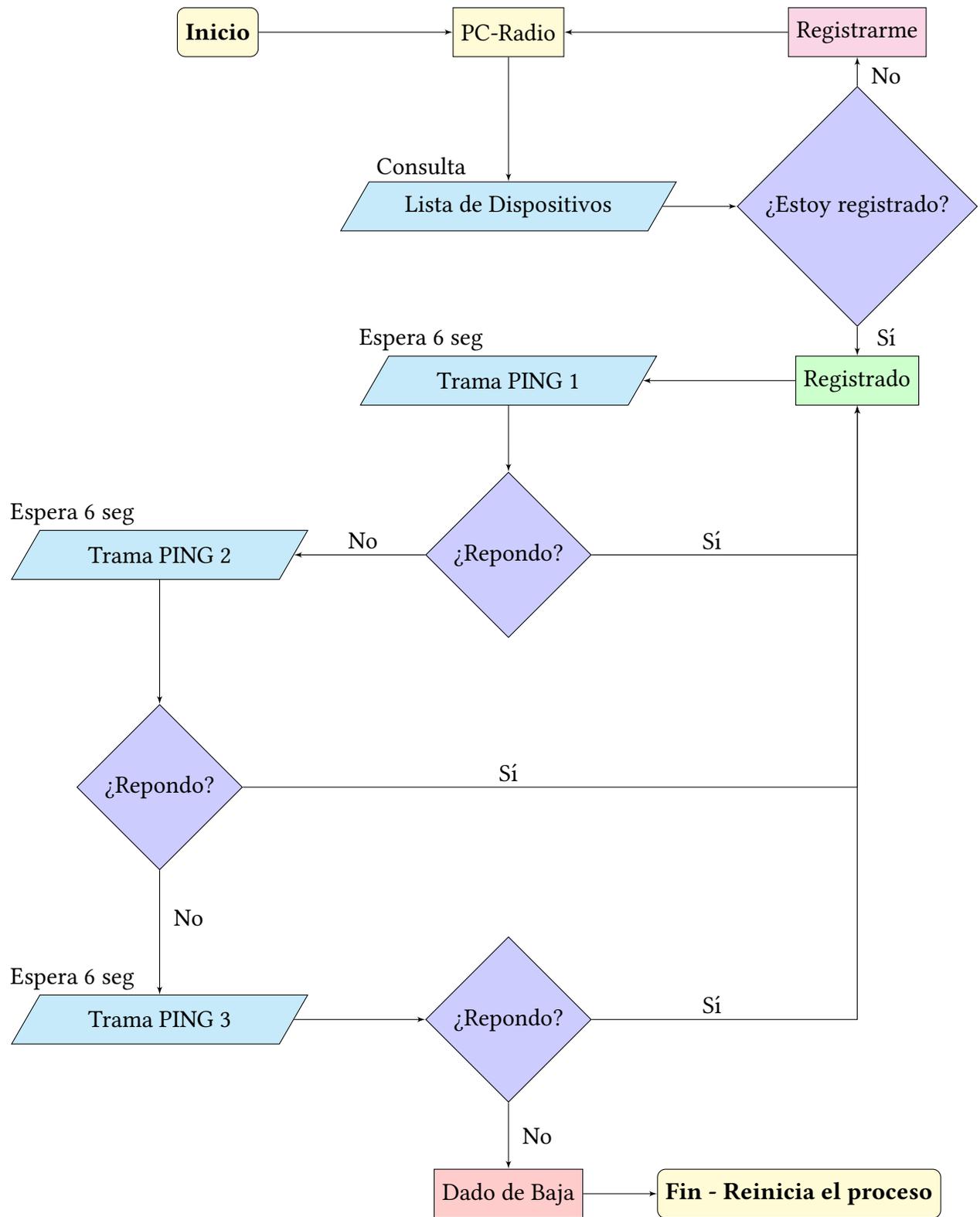


Figura 4.6: Registrarse en la red.

a dos posibles motivos: interferencia o distancia. Dadas las condiciones de campo abierto sin obstáculos, los radios pueden comunicarse libremente sin interferencia. De esta manera el *RSSI* nos permite estimar la distancia del subsistema de comunicación móvil respecto a las bases, debido a la calidad del enlace.

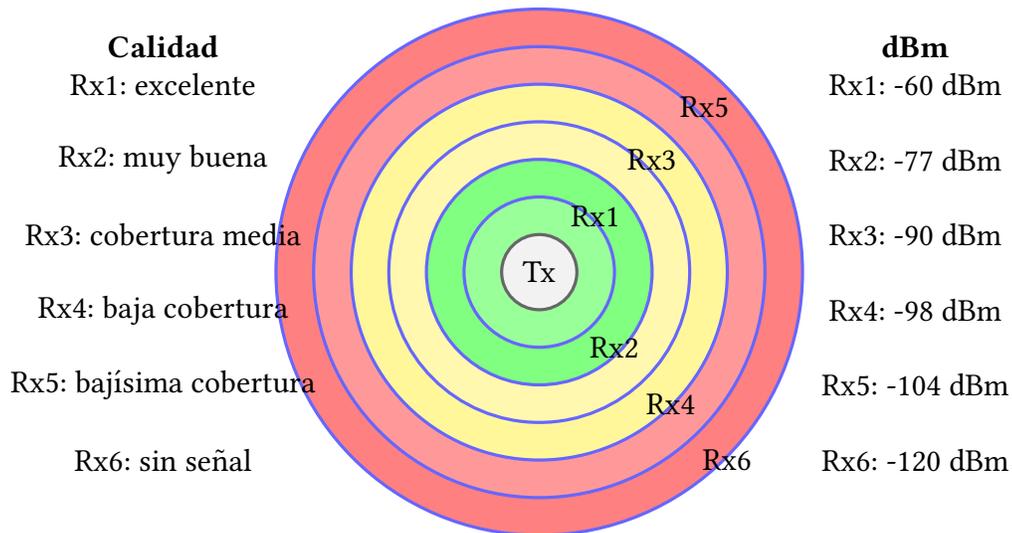


Figura 4.7: RSSI según la distancia.

## 4.5. Reconocimiento de tramas

Durante toda la sección se ha hablado acerca de las tramas. Los arreglos de arreglos de 8-bits enviados por el radio, una vez convertidos a hexadecimal se conocen como “Tramas”, es decir que cada arreglo de 8-bits es un campo de la trama. Estas tramas tienen una estructura definida que nos permite identificarlas. Algunos campos se dividen en “parte alta” y “parte baja”, por lo que el conjunto de ambos nos da un arreglo de 16-bits.



Tabla 4.2: Formato de tramas de comunicación.

De la Tabla 4.2 tenemos tres campos de 16-bits de los cuales, algunos contienen parte alta y parte baja: **Do**, **Dd** y **C**. De éstos, el campo **C** aunque posee una parte alta y una parte baja, solo nos da información cuando son los 16-bits, en el caso de los campos **Do** y **Dd**, la parte alta de 8-bits nos dice qué tipo de dispositivo es (aunque en algunas ocasiones puede variar la

nomenclatura) y la parte baja nos dice su identificador (ID), es decir, una vez que convertimos la secuencia del puerto serial a la trama, se puede identificar qué tipo de dispositivos se están comunicando a través de los radios y qué identificador tienen.

Es estrictamente importante saber el ID del radio que se emplea (ver Tabla 4.3), ya que al enviar una trama al radio si ponemos una ID de otro radio, nuestro radio la descartará por no ser para él. Por otro lado, en el caso de la PC, podemos usar cualquier término que tengamos registrado, ya que lo identificará y podrá comunicarse. Aunque es altamente recomendable usar el propuesto por el fabricante, ya que algunos valores son restringidos o usados para otro tipo de destino o nodo, como es el caso del valor 0XFF que significa que se transmite de manera *broadcast*, es decir que de un radio hacia todos los nodos que puedan recibirlo.

Código	Tipo	ID	Ejemplo	Dispositivo
0XF0	PC	Va de 0X00 a 0XFF	0XF00A	PC número 10
0X30	Radio	Va de 0X00 a 0XFF	0X3020	Radio número 32

Tabla 4.3: ID de dispositivo PC y radio.

#### 4.5.1. Identificar una trama

La identificación de una trama proveniente de los radios está relacionada directamente con conocer el formato de la misma. Este formato posee la información suficiente para reconocer cualquier tipo de trama enviada e identificar su función.

En la Tabla 4.4 vemos como está conformada una trama enviada de un radio a la PC [10].



Tabla 4.4: Trama enviada por el radio.

Cada uno de estos campos posee valores determinados, longitud en bits y una descripción. Se puede consultar la descripción completa de cada campo en el Anexo A.2. En la Tabla 4.5 solo se muestra lo necesario para identificar una trama.

De la Tabla 4.4 se observa que los campos **IT** y **FT** tienen la función de encapsular la trama. Ambos campos poseen valores fijos, a diferencia de los demás que son variables. Otro dato muy importante que obtenemos de la tabla es la posición relativa de los campos, es decir, el campo

<b>Campo</b>	<b>Nombre</b>	<b>Valor</b>
<b>IT</b>	Inicio de trama	Su valor siempre será 0xFE.
<b>L</b>	Longitud de trama	Cuenta el contenido de los demás campos sin incluir los campos IT y FT.
<b>Do</b>	ID del radio conectado a la PC	El valor de la MAC será 0x30XX.
<b>Dd</b>	ID de la PC conectada al radio	El valor de la MAC será 0xF0XX.
<b>I</b>	Instrucción	El valor del campo va de 0X00 a 0XFF.
<b>N</b>	Número de Secuencia	El valor del campo va de 0X00 a 0XFF.
<b>R</b>	Reservado	Este es un campo reservado. Puede variar cuando el radio regresa valores de estado, pero en su defecto es 0X00.
<b>M</b>	Mensaje	Variable (Máximo 712 bits).
<b>C</b>	Código de redundancia cíclica (CRC)	Se calcula con base a los campos L, Do, Dd, I, N, R, M de la trama.
<b>FT</b>	Final de trama	Su valor siempre será 0xEF.

Tabla 4.5: Descripción de los campos de la trama enviada del radio a la PC.

**L** siempre se encuentra después del campo **IT**, y así sucesivamente con cada uno de los campos hasta llegar al campo **FT** que se encuentra después del campo **C**.

## CRC

El orden de los campos es así obligatoriamente ya que si se busca por ejemplo leer la longitud de la trama se acude al campo número dos, y si no es correcta la longitud (suponiendo que la conocemos), entonces podemos descartar la trama por error lo cual se puede reconocer con el campo **C** que hace un cálculo booleano<sup>3</sup> con los bits de toda la trama sin considerar los campos **IT**, **C** y **FT**.

A diferencia del radio, la PC no descarta automáticamente las tramas con errores, por lo que se hace uso del código de redundancia cíclica (CRC) para poder descartarlos. Al ser enviada una trama, el emisor calcula el CRC y se envía en el campo **C**; el receptor puede volver a calcular el CRC y comparar con el CRC que recibió en el campo **C**; si ambos coinciden, entonces se estima que la trama no sufrió ninguna modificación, ya que la probabilidad de error disminuye.

<sup>3</sup>Este cálculo es un algoritmo estándar empleado por la empresa TTR [10] que regresa 16-bits.

### Identificando la trama recibida

Durante toda la Sección 4.5 se ha mencionado el reconocimiento e identificación de tramas, sin embargo no se ha dado ningún método o algoritmo que permita dicha identificación. Esto es con la intención de dejar claro que no hay un sólo algoritmo o método para identificarlas o reconocerlas, el algoritmo que se plantea a continuación es un diseño *ad hoc* para este trabajo, pero posee algunas generalidades básicas que incluso permiten reconocer tramas no identificadas por la PC.

El algoritmo completo se puede dividir en dos partes a gran escala: identificación y creación de tramas. En esta sección solo se verá la parte de identificación y en la Sección 4.6 su complemento de creación de tramas.

El algoritmo emplea cuatro campos de la trama para poder reconocerla, estos son: el inicio de trama, la longitud de trama, el CRC y el final de la trama. Se tiene conocimiento de que tanto el inicio de trama como el final poseen valores fijos. El campo de longitud de trama contiene la cantidad de bits que hay entre los campos de inicio de trama y final de trama, esta cantidad también se puede verla como una posición, es decir, si el inicio de trama es la posición uno, sumando la longitud de toda la trama (el campo **L**, en la posición dos) se tiene la posición del fin de la trama. Con esto podemos construir una trama completa.

El algoritmo de reconocimiento se puede representar en un diagrama de flujo, como se muestra en la Figura 4.8. Es importante destacar que se tiene la consideración de que el puerto serial ya ha sido configurado y se encuentra abierto, por lo que contamos con un flujo de información. Entonces podemos indicar paso a paso el proceso como se muestra a continuación:

Con este algoritmo es posible recuperar toda la trama desde el campo **IT** hasta el campo **FT**, teniendo el conocimiento de la posición de cada uno de los campos, se puede tomar los valores del campo **L** hasta el campo anterior al campo **C**, con esto es posible calcular el CRC y comparar con el campo **C** que fue recibido. Si ambos valores son iguales, entonces se asume que la trama no sufrió ninguna alteración. Ahora hay que identificar qué tipo de trama fue recibida.

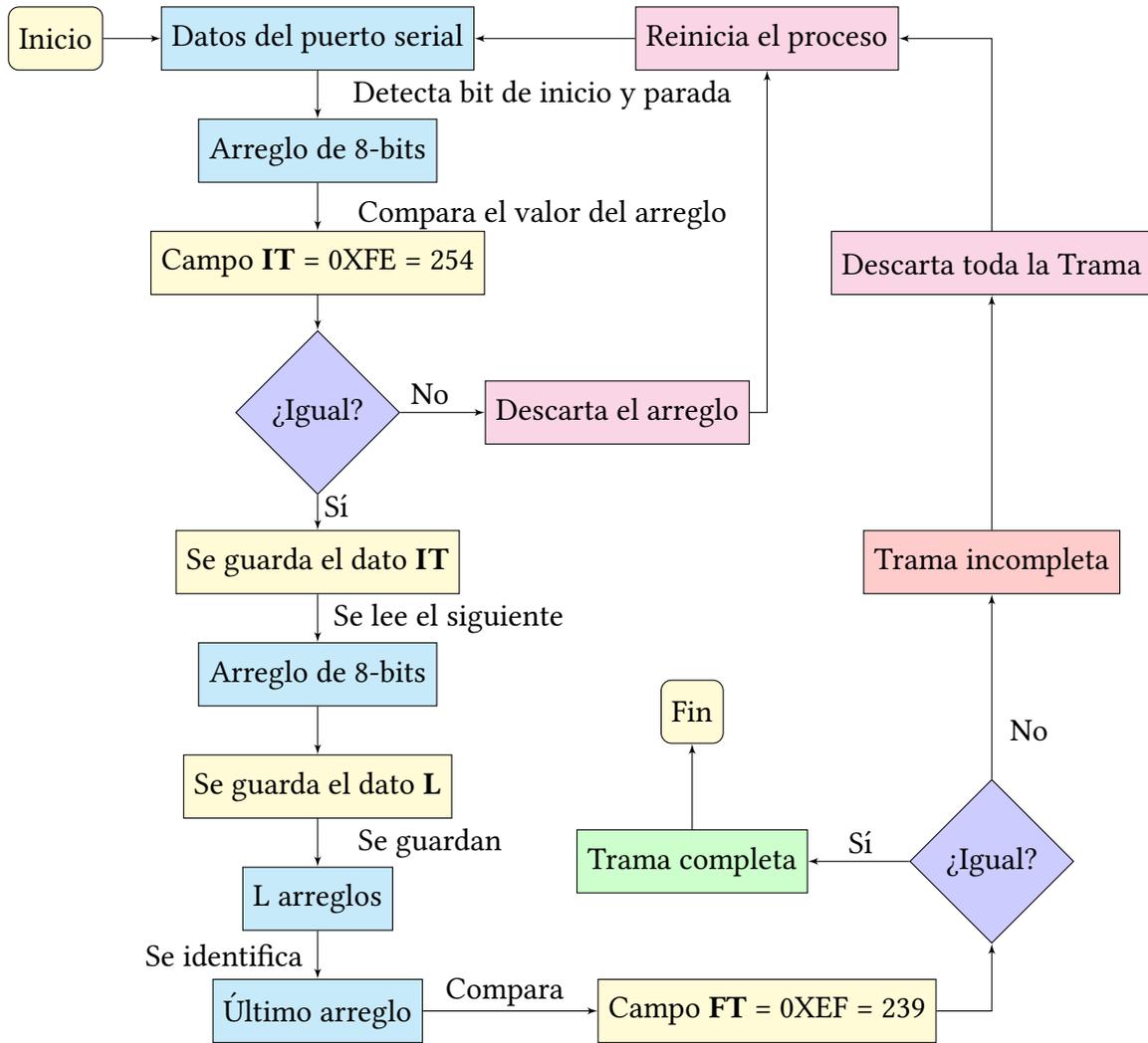


Figura 4.8: Algoritmo de reconocimiento de trama.

### 4.5.2. Identificando el tipo de trama

Una vez que se sabe la cantidad de campos que vienen en la trama, y ésta ha sido verificada por el CRC, tenemos ubicada la posición de los campos de la misma como se muestra en la Tabla 4.6.

#1	#2	#3	#4	#5	#6	#7	#8	#9	#?	#(n-2)	#(n-1)	#n
IT	L	Do <sub>H</sub>	Do <sub>L</sub>	Dd <sub>H</sub>	Dd <sub>L</sub>	I	N	R	M	C <sub>H</sub>	C <sub>L</sub>	FT

Tabla 4.6: Posición de campos de la trama de Comunicación.

La posición de los campos **M**, **C** y **FT** no son continuos al campo **R** porque la longitud del campo **M** es variable y en algunos casos inexistente. Por ello se toma de referencia la última posición de la misma manera que evaluamos el CRC en la Sección 4.5.

La posición relevante es la #7, el campo Instrucción, este campo da a conocer la finalidad de la trama y su contenido. A grandes rasgos se pueden dividir las tramas en dos categorías: las de instrucción directa y la de envío de datos. De esta manera se pueden reconocer aquellas tramas que envían información y se puede recibir y decidir que hacer con ella. Hay cinco tramas esenciales que se deben identificar, las cuales se muestran en la Tabla 4.7.

Campo I	Operación
0x03	Trama Lista de Dispositivos. Esta trama sirve para que el radio le informe a la PC de los dispositivos que se encuentran registrados en la red. Al inicio, el único dispositivo registrado será el propio radio.
0xFE	Trama de reconocimiento a la solicitud de registro. Cuando la PC solicita su registro, el radio le contesta con esta instrucción de reconocimiento de recepción de su solicitud de registro.
0x04	Instrucción para enviar una trama de datos. Mediante esta instrucción el radio le indica a la PC que el campo <b>M</b> contiene un paquete de datos.
0x01	Trama de verificación de dispositivo conectado (trama PING). Esta trama sirve para que el radio verifique que la PC sigue aún conectada.
0xFC	Trama de reconocimiento a la solicitud de envío de datos. Cuando la PC envía una trama con datos, el radio antes de encaminar los datos, debe responderle a la PC con esta trama de reconocimiento de recepción e indicarle mediante un código especial si su petición puede ser procesada.

Tabla 4.7: Descripción de las instrucciones enviadas del radio a la PC [10].

## 4.6. Creación de tramas

Se ha hablado sobre cómo hacer la conexión entre el radio y la PC, y cómo reconocer las tramas que el radio envía a la PC. Ahora se describirá cómo se arman las tramas para la comunicación con el radio. Esto complementará todo el proceso de extremo a extremo tanto de radio a PC como de PC a radio. Y una vez concluido este capítulo, se podrá entrar en detalle sobre la misión a realizar con el cuadricóptero.

Cuando se lleva a cabo la conexión de la PC con el radio, la primera trama que se recibe es la “Trama Lista de Dispositivos” como se muestra en la Tabla 4.7. En esta trama, al inicio, el único dispositivo registrado será el propio radio. La PC, al analizar esta trama y verificar que su dirección MAC no se encuentra en ella, debe proceder a registrarse. Por lo que se debe crear la primera trama de la PC al radio. En la Tabla 4.8 se ve como esta conformada la trama que debemos crear.

IT	L	Do	Dd	I	N	R	M	C	FT
----	---	----	----	---	---	---	---	---	----

Tabla 4.8: Trama enviada por la PC.

La descripción de sus campos es idéntica, solo cambian de posición el campo **Do** y **Dd**.

Campo	Nombre	Valor
IT	Inicio de trama	Su valor siempre será 0xFE.
L	Longitud de trama	Cuenta el contenido de los demás campos sin incluir los campos IT y FT.
Do	ID de la PC conectada al radio	El valor de la MAC será 0xF0XX.
Dd	ID del radio conectado a la PC	El valor de la MAC será 0x30XX.
I	Instrucción	El valor del campo va de 0X00 a 0XFF.
N	Número de Secuencia	El valor del campo va de 0X00 a 0XFF.
R	Reservado	Este es un campo reservado. Puede variar cuando el radio regresa valores de estado, pero en su defecto es 0X00.
M	Mensaje	Variable (Máximo 712 bits).
C	Código de redundancia cíclica (CRC)	Se calcula con base a los campos L, Do, Dd, I, N, R, M de la trama.
FT	Final de trama	Su valor siempre será 0xEF.

Tabla 4.9: Descripción de los campos de la trama enviada de la PC al radio.

#### 4. Radio comunicación

---

Como se ve en la Tabla 4.9 las descripciones son las mismas, pero cambian el tipo de instrucciones que se envían. Tres de las instrucciones que se pueden emplear se muestran en la Tabla 4.10.<sup>4</sup>

Campo I	Operación
0x02	Trama de Registro. Esta trama sirve para que la PC le anuncie al radio que desea formar parte de la red y hacer uso de la misma para transmitir información.
0xFF	Trama de respuesta a las instrucciones PING provenientes del radio. Cada vez que la PC recibe una instrucción de tipo PING proveniente del radio, debe contestar con esta trama para mantenerse registrada.
0x04	Instrucción para enviar una trama de datos. Mediante esta instrucción la PC le indica al radio que el campo <b>M</b> contiene un paquete de datos.

Tabla 4.10: Descripción de las instrucciones enviadas de la PC al radio.

Con estas tres instrucciones se puede llevar a cabo toda la misión; la pieza clave es la instrucción para enviar una trama de datos, tanto de parte del radio como de parte de la PC. En este tipo de trama se puede mandar información de una PC a otra que se encuentre registrada en la misma red. Como se muestra en la Figura 4.9.

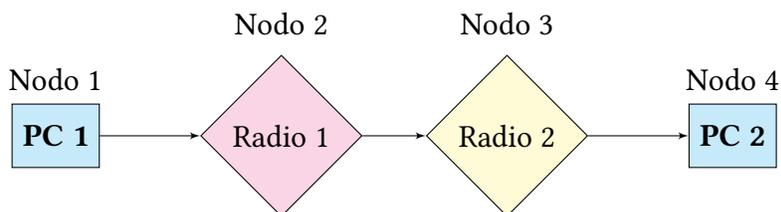


Figura 4.9: Comunicación PC 1 a PC 2.

Para ello es necesario entender que cada PC se comunica a nivel local con su respectivo radio. Una vez registradas y activas, entonces se puede enviar tramas de datos a través de la red. Estos radios tienen la capacidad de formar una red tipo *mesh*, por lo que mientras se encuentren dentro del alcance de transmisión todos los radios y PCs formarán parte de la misma red. Las redes tipo *mesh* son aquellas donde todos los nodos de radio están conectados entre sí, por lo tanto también sus respectivas PC. Por ende, este trabajo se puede extender al uso de múltiples cuadricópteros simultáneamente para transferir un mayor volumen de información.

---

<sup>4</sup>Existen otras instrucciones que se pueden consultar en el Anexo A.2, sin embargo para la elaboración de esta tesis nunca se utilizaron.

### 4.6.1. Trama de datos

La trama de datos que se usa para enviar información de una PC a otra y puede tener las funciones que nosotros le demos, esto se debe a que la trama de datos es una subtrama encapsulada dentro de una trama. La trama debe tener una estructura definida, esto es igual para la subtrama, aunque se puede usar el campo **M** de la subtrama es decir, el campo **Mm**, para crear campos que la trama no contempla, esto siempre y cuando el receptor esté al tanto de estos cambios. El campo **M** tiene una estructura como se muestra en la Tabla 4.11. En otras palabras, el campo **M** es un conjunto de campos, o una subtrama.

<b>Df</b>	<b>Ds</b>	<b>IT</b>	<b>Lm</b>	<b>Nm</b>	<b>Im</b>	<b>Mm</b>	<b>Cm</b>	<b>FT</b>
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

Tabla 4.11: Estructura del campo **M**

Esta subtrama cuenta con algunas diferencias a la trama principal, entre ellas podemos observar la posición de los campos de transmisión y receptor, y la ausencia del campo **R**. Esto último se debe a que el campo **R** es utilizado por el radio para indicar si la trama principal llegó bien o hubo algún problema. En el caso de las subtramas, éstas son enviadas a otra PC y las tramas son enviadas bajo el protocolo de “mejor esfuerzo”, por ende el campo **R** resulta innecesario; sin embargo, para el objetivo de la misión e implementación del protocolo, se tiene que tomar en cuenta.

Al momento de crear la subtrama de datos se trata de un arreglo. Dentro de ese arreglo debemos preparar el campo **M** como se muestra en la Tabla 4.11 y la trama principal como se muestra en la Tabla 4.8. Para la creación de la trama es importante el orden como se llena el arreglo que se va a enviar. Los primeros campos a llenar del arreglo son aquellos que se mantienen fijos, como lo son en la trama principal el campo **IT**, el campo **FT**, el campo **Do** y el campo **Dd**. También tenemos el campo **I**, y el campo **R**, que en este caso deben ser 0x04 y 0x00, respectivamente. El siguiente campo a llenar es el campo **N**, el cual es el número de secuencia y determina que número de mensaje es: si es el primero se puede comenzar en 0 o 1, en caso contrario se aumente el valor de 1 en 1. Con todo esto, solo queda pendiente el campo **L**, el campo **M** y el campo **C**, ahora es momento de llenar los campos de la subtrama en el arreglo. Hay diferentes instrucciones de subtrama para enviar datos como se muestra a continuación en la Tabla 4.12.

<b>Campo Im</b>	<b>Operación</b>
0x04	Instrucción para enviar una trama de datos. Mediante esta instrucción la PC le indica al radio que el campo <b>M</b> contiene un paquete de datos.
0x43	Paquete de solicitud de RSSI. La función de este paquete es la de solicitarle al radio su RSSI respecto a otro radio especificado en el campo <b>M</b> .

Tabla 4.12: Descripción de las instrucciones enviadas de la PC al radio.

De las dos instrucciones que se muestran en la Tabla 4.12, usaremos la instrucción 0x43 para marcar “banderas de posición”, y la instrucción 0x04 para el protocolo que se empleará, para este ejemplo se usará la instrucción 0x04, ya que esta usa más campos que la otra.

Una “bandera de posición” es la posición donde el *Drone* determina que se encuentra a una distancia suficiente para comenzar a transmitir la información. Dicha distancia es considerada suficiente cuando la calidad del enlace es excelente, tomando como referencia la Figura 4.7, donde se habla sobre el RSSI.

En el Capítulo 5 se describe a detalle el proceso de las “banderas de posición” y como se lleva acabo la transmisión de la información así como qué consideraciones se toman para elegir el RSSI deseado.

Lo más importante es el protocolo “*RQ Idle*” [9] modificado que se emplea y las modificaciones que se realizan al protocolo de los radios.

---

# Solución del Problema

---

En este capítulo se muestra la solución propuesta de todo el proyecto de tesis desde la perspectiva externa del *drone* y la perspectiva interna del protocolo de comunicación.

## 5.1. Planteamiento de la solución

Para resolver el problema se plantea una misión que consiste en el establecimiento de conectividad en redes de comunicaciones mediante un vehículo aéreo autónomo. Para ello se realiza una ruta con el cuadricóptero que permite transferir información entre dos nodos. Estos nodos se encuentran ubicados en un campo de fútbol, uno en cada portería (ver Figura 5.1), por lo que es necesario trasladarse a lo largo del campo de fútbol para establecer la comunicación. La disposición de los elementos se muestra en la Figura 5.1.

Los nodos son el conjunto de una PC y un radio. Se le conoce como nodo base al nodo que transmite información y nodo destino al nodo que la recibe, y también es posible que un nodo pueda llevar a cabo ambas funciones, por lo que es un nodo base/destino dependiendo de la función que lleve a cabo en ese momento. En el caso de los nodos ubicados en las porterías, se tiene un nodo base que quiere transmitir a un nodo destino, que es la Estación B (ver Figura 5.1). El cuadricóptero es tanto nodo base como nodo destino ya que primero recibe la información y posteriormente la transfiere. El alcance de cada uno de estos radios es de alrededor de 300 m [10], pero se ha reducido su alcance a través del uso de software para ejemplificar la misión en el campo de fútbol. Para fines prácticos este tipo de misiones se llevan a cabo a mayor escala, como por ejemplo rodear una montaña para conectar dos pueblos a través de un enjambre de UAVs, donde cada pueblo representaría uno de nuestros nodos.

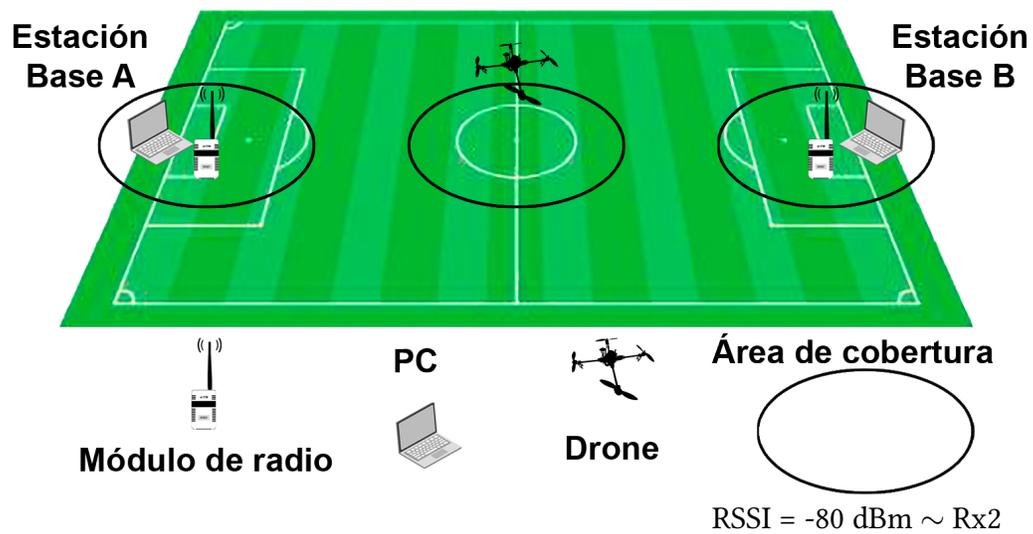


Figura 5.1: Campo de fútbol.

Un nodo de red es un punto de conexión en una red de comunicaciones, como el caso de las PC en la Figura 5.2. Cada nodo es un punto final para la transmisión o redistribución de datos. Los nodos tienen una capacidad programada o diseñada para reconocer, procesar y reenviar transmisiones a otros nodos de la red como en el caso de la red tipo malla [20].

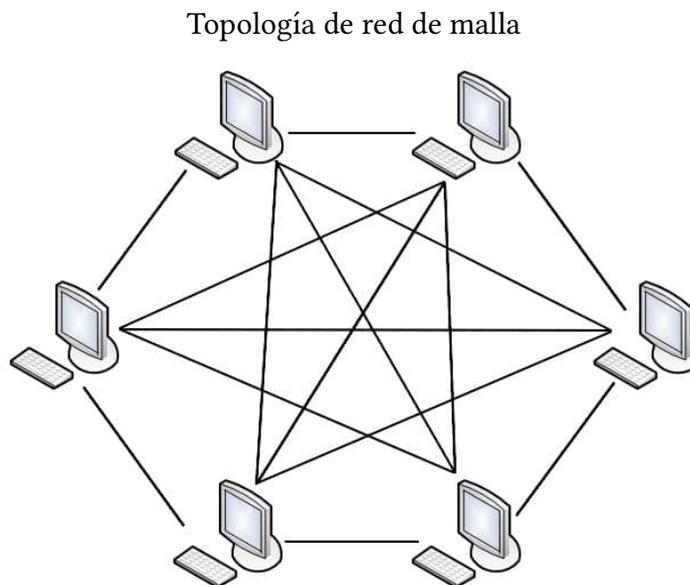


Figura 5.2: Red tipo malla.

En el caso particular de la misión estudiada en el presente trabajo de tesis, los nodos ubicados en las porterías están desconectados debido a la distancia entre ellos, por lo que se usará el nodo móvil central, que es el dron.

En este trabajo se maneja de manera indistinta el nombre de nodos de red, nodos de comunicación o estaciones base, refiriéndonos en todos los casos al conjunto de una PC y un radio. Al referirnos a la estación base A nos referimos a la PC con identificador en la red “BA0A” y el radio “0X3021”, en el caso de la estación base B nos referimos a la PC con identificador “BA0B” y el radio “0X3022”. Así mismo, el nodo móvil o estación móvil C también se le dice de forma indistinta *drone* o cuadróptero el cual se refiere a la Raspberry “BAD0” y el radio “0X3020” (ver Tabla 5.1 y Figura 5.3).

Nombre	Estación	ID de PC	ID de radio
Estación Base A	A	BA0A	0X3021
Estación Base B	B	BA0B	0X3022
Cuadróptero/Drone	C	BAD0	0X3020

Tabla 5.1: Estaciones Base y sus identificadores.

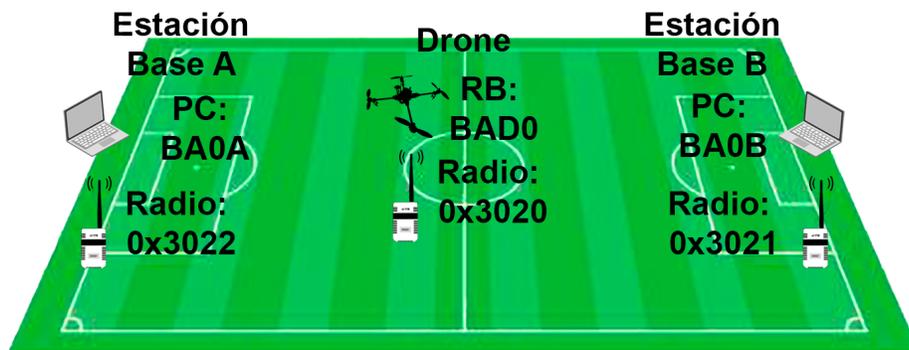


Figura 5.3: Estaciones Base y sus identificadores.

## 5.2. Plan de vuelo

El plan de vuelo presenta la ruta que el *drone* debe seguir paso a paso, incluidos los momentos donde debe transmitir, quedarse en modo de vuelo “*hover*” o moverse de un punto a otro. Para esta misión el plan de vuelo consta de 7 pasos distintos que serán descritos a continuación, a los cuales se les denominará etapas.

Etapa 1. Durante la primera etapa se tiene el despegue en el centro del campo (ver Figura 5.4).

En este punto todos los sistemas se encuentran funcionando, el radio abordo del *drone* trata de comunicarse con el nodo base B (Estación Base B) (derecha en la Figura 5.4). Una vez que alcanza cierta altura establecida (aproximadamente 10 metros) durante el experimento, el *drone* pasa a la etapa 2.

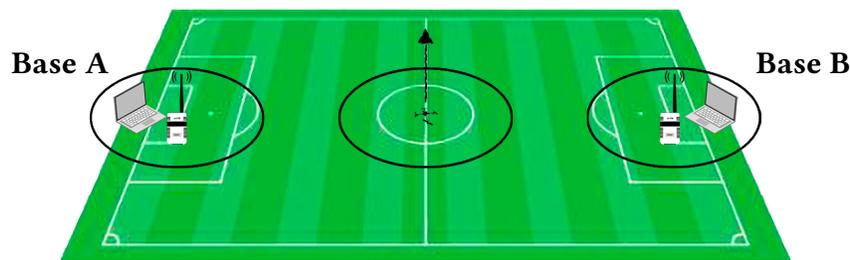


Figura 5.4: Etapa 1 del plan de vuelo.

Etapa 2. Después de alcanzar la altura establecida el *drone* debe comenzar su vuelo hacia la portería marcada en la posición de la Estación Base B (derecha). Durante el recorrido que se muestra en la Figura 5.5 tenemos la comunicación del radio del *drone* con el radio de la portería en la posición B, en donde se solicita la intensidad de señal a partir de una trama y nos permite saber la calidad del enlace (ver Figura 4.7), lo cual es posible convertir a nuestra aproximación en distancia. El *drone* comenzará a transmitir a partir de una calidad de enlace Rx2 equivalente a sobrevolar la estación base B.

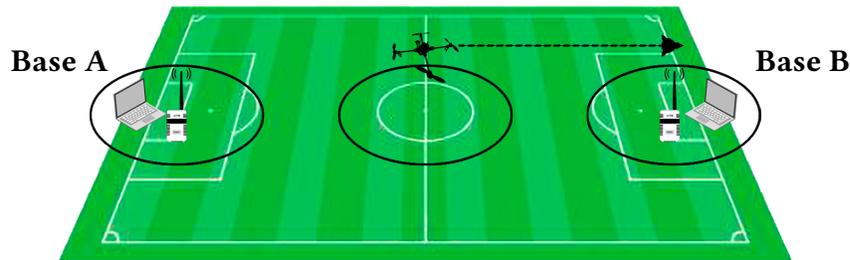


Figura 5.5: Etapa 2 del plan de vuelo.

Etapa 3. Al final de la etapa 2 se llega a la posición de la Estación Base B, donde comienza la etapa 3 entrando en modo de vuelo “*hover*” sobre la Estación Base B, ubicada en la portería derecha de la Figura 5.6. El *drone* permanecerá ahí un tiempo establecido, el cual será suficiente para descargar todos los paquetes de información solicitados como se muestra en la Figura 5.6.

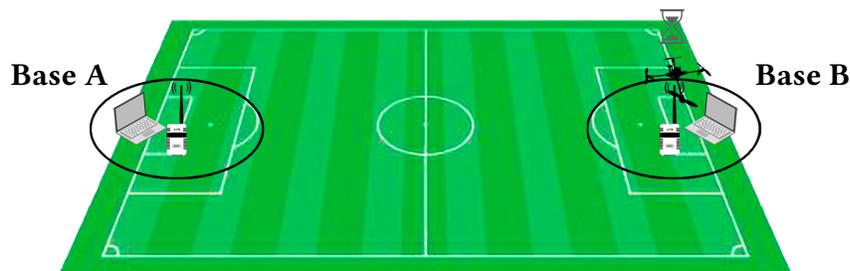


Figura 5.6: Etapa 3 del plan de vuelo.

Etapa 4. En este punto, una vez que la transmisión descrita en la etapa 3 esté completa, se procede a la siguiente etapa (etapa 4). Aunque es posible disminuir el tiempo de la trayectoria del *drone*, transmitir durante la aproximación a la estación base B y durante su salida, la calidad de enlace elegida para el experimento de campo permite

comenzar la transmisión tan pronto se entra a la zona de cobertura y transmitir hasta su salida para asegurar que el documento esté completo.

Durante la etapa 4 como se muestra en la Figura 5.7, se lleva a cabo un recorrido del *drone* desde la portería en la Estación Base *B* hasta la portería en la Estación Base *A*. Durante esta trayectoria se aleja de la Estación Base *B*, por lo que disminuye la intensidad de señal con esta estación y aumenta la intensidad de señal con la Estación Base *A*. Este recorrido de portería a portería representa la distancia que no permite la comunicación entre ambas Estaciones Bases en tierra.

Una vez que el *drone* tenga el documento completo descargado a partir del envío de múltiples paquetes, la estación Base *B* dejará de enviarle información y solo solicitará la calidad de enlace para dar seguimiento al *drone* mientras se aleja del área de cobertura.

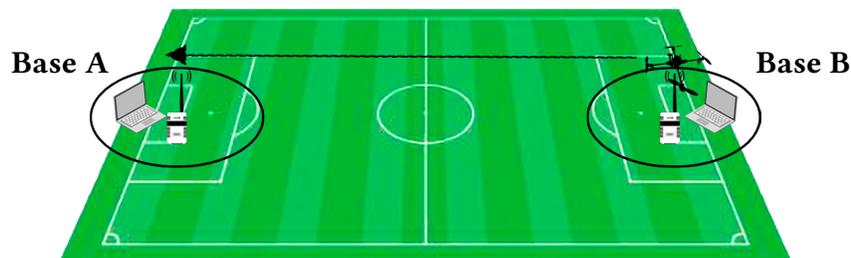


Figura 5.7: Etapa 4 del plan de vuelo.

Etapa 5. Una vez que se alcanza la intensidad de señal deseada y se sobrevuela la Estación Base *A*, se comienza a transferir la información del *drone* a la base. En esta etapa, como se muestra en la Figura 5.8, el *drone* se encuentra en modo de vuelo “*hover*” durante un minuto y medio (dicho tiempo se calculó a partir de los tiempos de transferencia de paquetes que se obtuvo en pruebas realizadas en laboratorio) para concluir la transferencia de información.

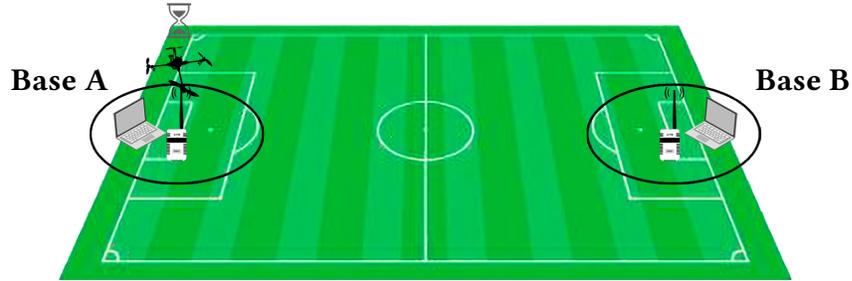


Figura 5.8: Etapa 5 del plan de vuelo.

Etapa 6. Durante la etapa 6 de la misión la descarga del documento está completa. El *drone* volverá al centro del campo justo donde fue su posición inicial y preparará su aterrizaje (ver Figura 5.9); para esta etapa el *drone* sigue enviando la intensidad de señal a ambas bases que lo solicitan. Esto es relevante solo cuando el *drone* lleva acabo una misión con envíos “parciales”, lo cual se explica más adelante, ya que para esta misión se considera un envío del documento completo, lo cual significa tiempos de espera en modo de vuelo “*hover*” hasta concluir los envíos entre las estaciones base y el dron.

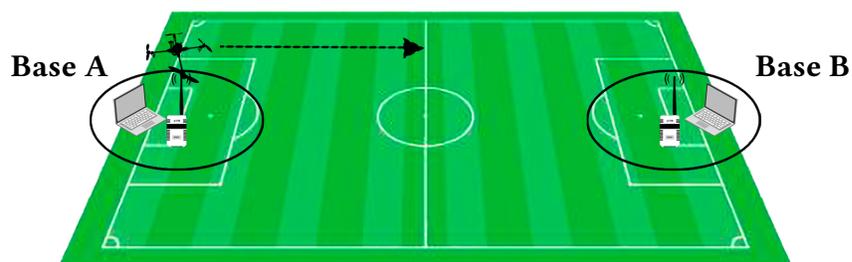


Figura 5.9: Etapa 6 del plan de vuelo.

Para un envío parcial, el *drone* en lugar de pasar a la etapa 7 de aterrizaje (ver Figura 5.10), el *drone* vuela de vuelta hasta la estación base B donde continua alternando entre las estaciones base, donde deja de transmitir del *drone* a la estación base A dejando la transmisión incompleta y vuela hasta la estación base B recibiendo los paquetes de información de esta estación base para entrar en un bucle desde la etapa 3 a la etapa 6 hasta concluir la transferencia de todo el documento.

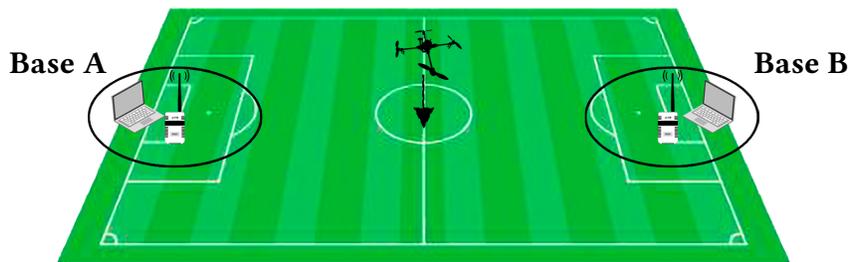


Figura 5.10: Etapa 7 del plan de vuelo.

Etapa 7. La etapa 7 es donde se concluye la misión teniendo al *drone* volando de vuelta a su posición inicial y aterrizando para completar la misión (ver Figura 5.10).

Se muestra el resumen de las diferentes etapas en la Figura 5.11.

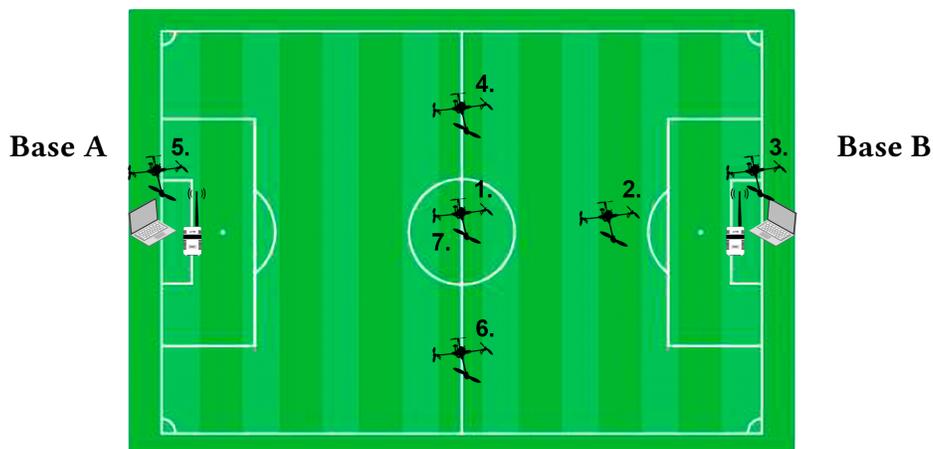


Figura 5.11: Resumen del plan de vuelo.

## 5.3. Programación de la misión

Durante el plan de vuelo se considera que el *drone* vuela hacia las porterías donde lleva a cabo la transferencia del archivo, sin embargo, los programas realizan sus funciones de manera independiente.

### 5.3.1. Programación de la ruta de vuelo

En el Capítulo 3 se menciona el cuadricóptero que se emplea durante la misión y la adaptación del prototipo para su desempeño adecuado. Para la configuración del plan de vuelo, se usó el *software* “*Mission Planner*” [21], que permite diseñar un plan de vuelo a partir de la ubicación GPS del drone.

*Mission Planner* es una interfaz que permite establecer una estación de control terrestre para proyectos de Ardupilot (ver Anexo A.1.4) de la cual se hizo uso para configurar una ruta de vuelo (ver Figura 5.12).

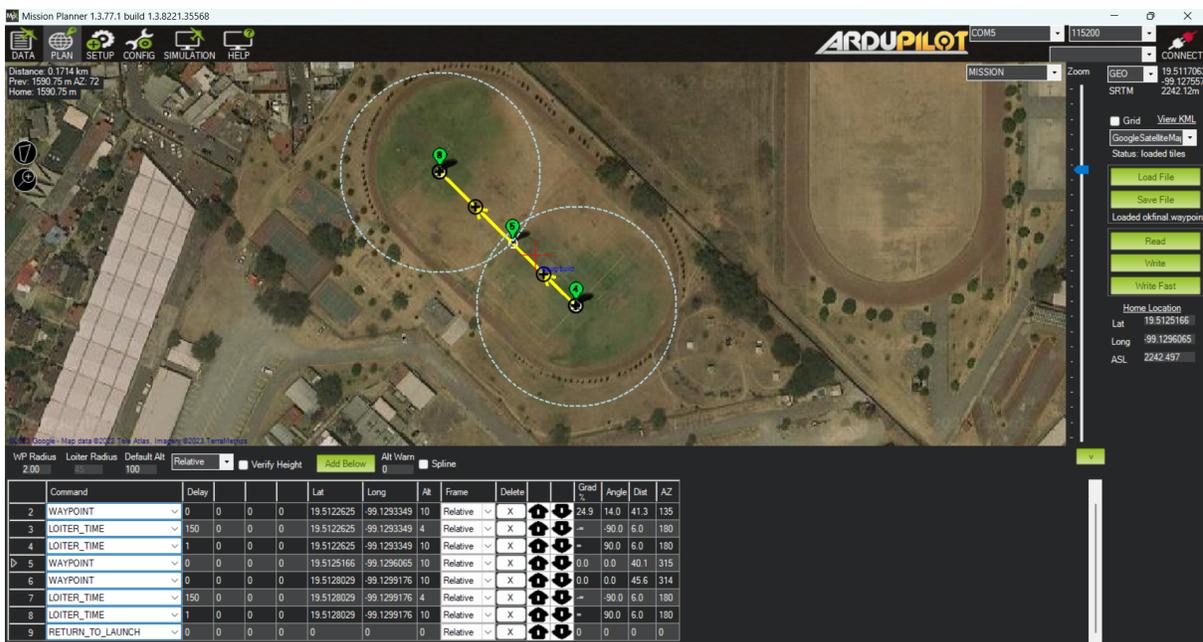


Figura 5.12: Diseño de ruta de vuelo.

La ruta programada sigue cada una de las etapas que se muestran en el resumen del plan de vuelo (ver Figura 5.11). Para más información sobre la configuración de *Mission Planner* para

el diseño del plan de vuelo se puede consultar el Anexo A.1.4.

### 5.3.2. Programación del protocolo de comunicación

Durante las pruebas de laboratorio respecto al protocolo de comunicación, se hace uso del protocolo RQ Inactivo (ver Anexo A.3); sin embargo, es necesario realizar algunas adaptaciones al protocolo original de los radios ya que estos trabajan con un protocolo de mejor esfuerzo.

Las funciones básicas del radio, tales como registro y comunicación entre radio y PC, se dejan funcionando sin modificación alguna. El protocolo se implementa (ver Figura 5.13) solo cuando se transmite o descarga el archivo de texto que se desea enviar de una base a otra a través del cuadricóptero, esto mediante las tramas de envío de datos.

Este protocolo usado por los radios sirve para mantenerse conectado dentro de la red. Estos procesos se consideran fuera del protocolo que empleamos ya que hacen uso de tramas específicas para poder mantenerse en operación; por ello, aunque todos estos procesos se lleven a cabo en segundo plano, no los consideraremos para describir el protocolo.

Para la implementación del protocolo se buscó que el tiempo de envío y descarga sea el menor posible. Esto es debido a que contamos con un tiempo límite independiente del sistema de comunicaciones, correspondiente al tiempo de vuelo. Dicho tiempo va ligado al consumo de los motores y la electrónica del *drone*. Durante la adaptación del *drone* se emplearon en todo momento los mismos motores, por lo que solo se vio afectada su configuración y peso. Por ello se consultó un estudio de relación de peso y autonomía donde se determina que a mayor peso mayor consumo [22] (ver Figura 5.14).

El tiempo de vuelo depende de la batería del cuadricóptero, cuya duración depende del consumo principalmente de los motores, lo que va relacionado al tiempo de vuelo; pero también se ve afectado el tiempo de vuelo por las variaciones de velocidad. En dicho estudio, realizado en la Universidad de Sevilla [22], se muestra la gráfica anterior de relación de autonomía y peso del drone que, aunque va a tamaños superiores al modelo utilizado en este trabajo, el comportamiento de la gráfica se mantiene para modelos de menor peso (esto puede variar dependiendo del tipo de batería empleada, aumentando el tiempo de vuelo pero conservando la curva de la gráfica).

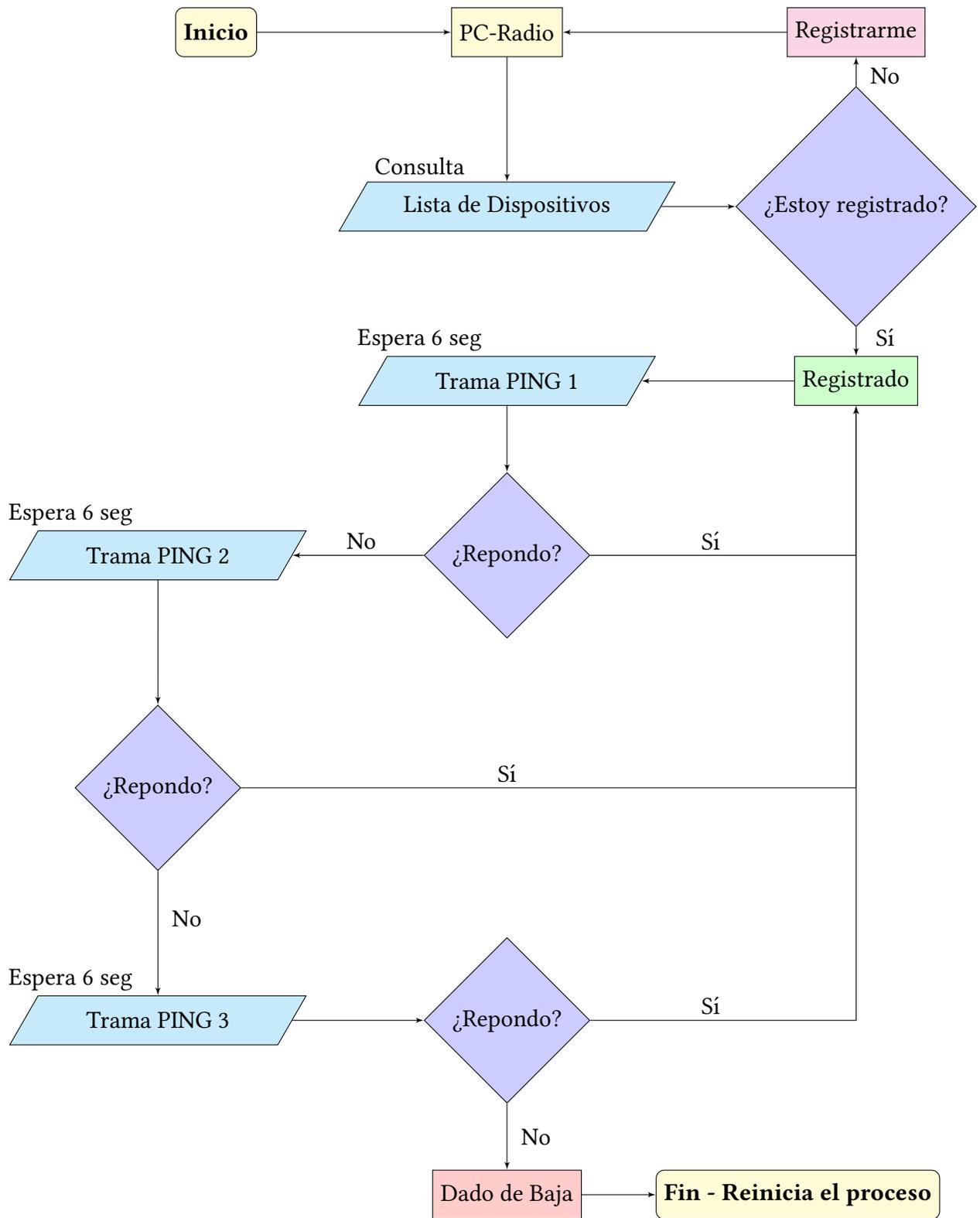
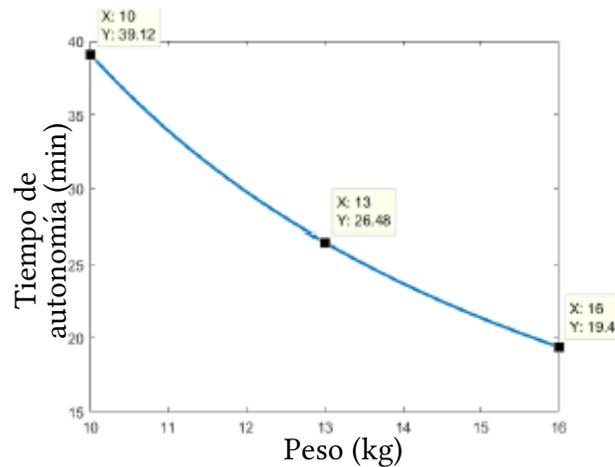


Figura 5.13: Protocolo del radio.

Figura 5.14: Autonomía vs peso del *drone*. [22]

## 5.4. Protocolo y pruebas de campo

En el Capítulo 4 se mencionan las tramas de comunicación que emplea el radio y su funcionamiento (ver Tabla 5.2). En este capítulo se hablará sobre la implementación del protocolo RQ: Inactivo solicitud Explícita (ver Anexo A.3.1) y las modificaciones sobre las tramas empleadas en los radios. Para implementar dicho protocolo es necesario modificar la estructura del campo **M** (sólo el campo interno **Mm**); esto debido a que los demás campos son verificados por el radio y dan información sobre la propia trama, lo cual al modificarse causaría el descarte de toda la trama por el mismo radio.



Tabla 5.2: Formato de tramas de comunicación.

### 5.4.1. Modificaciones al campo **M**

El campo **M** representa por si mismo una subtrama (una trama dentro de la trama principal) de la cual se explica toda su estructura en el Capítulo 4. El campo **Mm** porta información no relevante para el radio, es por eso que podemos transmitir en ese campo cualquier contenido de caracteres. En la subtrama del campo **M** podemos observar el campo **Mm** (ver Tabla 5.3).

El campo **Mm** es el único campo de longitud variable (ver Tabla 5.4), variando entre 0 y 89 bytes según la información a transmitir. Una trama no puede enviar más de 89 bytes de



Tabla 5.3: Estructura del campo M

información. En caso de intentar enviar más de 89 bytes de información, el radio receptor tomará solo los primeros 100 bytes y eso dejaría una trama trunca causando que en la posición 100 no se encuentre la trama final y sea descartada por el radio como trama incompleta, por lo que para cualquier archivo que tenga un tamaño superior, es necesario dividir la información en múltiples tramas.



Tabla 5.4: Bytes de cada campo de la subtrama del campo de Mensaje.

Para implementar el protocolo RQ Inactivo, es necesario tener campos de información exclusivos del protocolo. En este caso se utilizan tres nuevos campos: ID de mensaje, número de mensaje y total de mensajes. Estos campos se ajustan a los requerimientos particulares de la misión, donde el protocolo podría funcionar con solo dos campos, sin embargo se justifica la existencia del tercer campo más adelante.

El campo de ID de mensaje sirve para corroborar qué tipo de información estamos enviando. El protocolo RQ: Inactivo hace uso de mensajes, acuses de mensajes, y no acuses de mensajes, por ello damos un identificador (ID) a cada uno de éstos (ver Tabla 5.5).

Id de mensaje	Interpretación
0	Mensaje
1	Acuse
2	No-acuse

Tabla 5.5: Campo ID de mensaje.

De acuerdo al protocolo RQ: Inactivo (ver Figura 5.15), si se envía un mensaje, el receptor debe responder con un acuse de recibido siempre y cuando el mensaje sea correcto sin distorsiones y sin haber sido dañado durante el proceso de transmisión (interferencias). Una vez que se envió el mensaje, si no se recibe de vuelta un acuse, hay dos posibles opciones: el mensaje no llegó o llegó dañado. Si el mensaje se perdió, entonces tenemos un tiempo de espera para recibir el acuse; si éste se termina, simplemente reenviamos el mensaje. Si el mensaje llegó

dañado, el receptor manda un no-acuse y se reenvía el mensaje. Esto último permite reducir el tiempo de espera de los acuses y agilizar el proceso de transferencia.

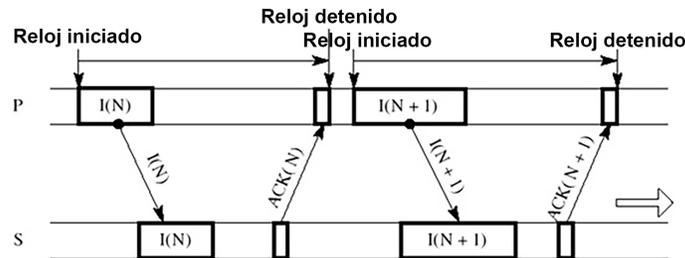


Figura 5.15: Tiempos de espera del protocolo RQ Inactivo.

El campo de número de mensaje suena algo redundante debido al campo **Nm** (ver Tabla 5.3) pero no lo es. El campo **Nm** indica el número de subtrama, mientras que el campo número de mensaje hace referencia al número de mensajes enviados de toda la información por enviar. Si se envía el primer mensaje, entonces el campo **Nm** y el campo **#m** comenzarían en 1, pero si el primer mensaje debe reenviarse entonces el campo **Nm** pasa a valer 2 y el campo **#m** sigue siendo el 1. Esto se traduce a que el campo **Nm** cuenta todos los mensajes incluidos los reenvíos, mientras que el campo **#m** solo considera el número del mensaje que se intenta enviar. Si extrapolamos al campo **N** (ver Tabla 5.2), este campo considera el número de tramas enviadas, incluidas aquellas que no son mensajes (por ejemplo, las tramas de registro o respuesta de ping.).

### 5.4.2. Transferencia del documento

Hasta este punto ya se ha descrito el funcionamiento de básico de los radios, las modificaciones para la implementación del protocolo de comunicación y como es la ruta de vuelo. Ahora se hará uso de todo lo mencionado para describir la misión con todo lo que sucede de forma interna tanto en el *drone* como en las bases, así como su interacción al aproximarse. Para ello usaremos las etapas mencionadas en el plan de vuelo.

Retomando las etapas de vuelo, se explicará ahora la parte interna del código mientras el *drone* se encuentra en movimiento, dividiéndolo también en etapas:

1. Inicialización del código.
2. La Base *B* busca al *Drone*.
3. La Base *B* inicia la transferencia al *Drone* y se mandan de forma continua los paquetes de la Base *B* al *Drone* hasta completar el envío.
4. La Base *A* busca al *Drone*.
5. La Base *A* solicita la información al *Drone* y se descargan de forma continua los paquetes del *Drone* hasta completar la descarga.

### Etapa 1 del código.

Durante esta etapa el *drone* se encuentra en tierra al igual que las Bases (ver Figura 5.16); y en este punto se arrancan los programas. Las estaciones base inician cada una su proceso de registro con el radio al igual que el *drone*. En el caso de la estación base B, ésta prepara el documento que se quiere enviar a la estación base A, para ello hace lectura de todo el archivo y divide el archivo en tramas de 50 caracteres cada una. Por otro lado, la estación base A intenta buscar al *drone*; en caso de recibir una señal adecuada con una calidad de Rx2 (ver Figura 4.7), el *drone* le dirá que aún no tiene el documento.

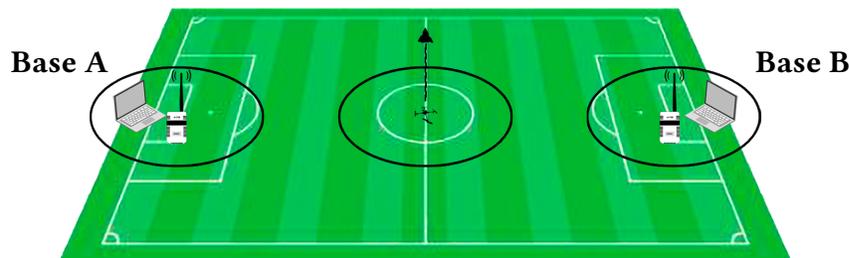


Figura 5.16: Etapa 1: Iniciando el código.

Lo más importante de esta etapa es la preparación del documento y que todos los programas se hayan registrado exitosamente. El periodo de registro de los radios es menor al tiempo de

espera de la conexión de satélites con el *pixhawk* requeridos para el vuelo; por lo tanto una vez que el *drone* esté listo para volar, los programas estarán listos para transmitir el documento.

Para la preparación del documento, se dividen todos los caracteres del archivo en una matriz de  $N \times 50$  (donde  $N$  es la cantidad de filas necesarias para todo el documento), en donde se va llenando de 50 caracteres hasta que se agote el documento y quede una última línea. Esta última por lo general no tendrá 50 caracteres ya que serán los caracteres que sobraron.

Para realizar la prueba se utiliza un archivo de texto consistente en un poema. El poema a enviar se trata de “*The Phoenix and the Turtle*” (ver Anexo A.3.2) por el autor William Shakespeare. Este poema consta con alrededor de 2100 caracteres<sup>1</sup>, por lo que se requieren al menos 42 tramas (2096 caracteres enviados en mensajes con 50 espacios requiere el envío de 41.92 mensajes).

En el caso de las pruebas de campo, se añadieron líneas de texto por los sinodales para comprobar la veracidad de la prueba.

### **Etapa 2 del código.**

Una vez que los programas se iniciaron y el documento se preparó, así como el *drone* realizó su conexión con los satélites y este listo para volar, tanto la Base *A* como la Base *B* intentarán comunicarse con el *drone* mientras éste inicia su ruta (ver Figura 5.17).

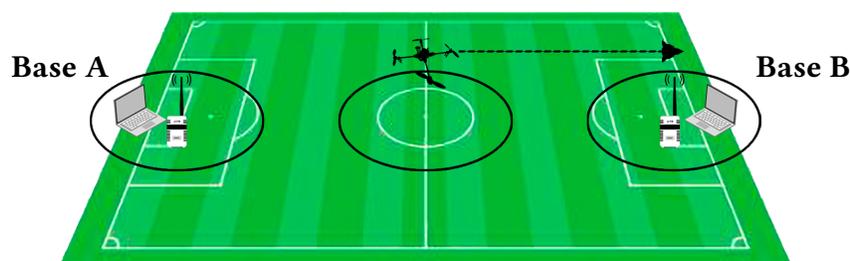


Figura 5.17: Etapa 2: Búsqueda del *drone*.

---

<sup>1</sup>A partir del texto procesado donde los espacios pasan a ser caracteres especiales y se obtiene una línea continua de texto.

Para ello, ambos radios de las bases intentarán saber el *RSSI* del radio del *drone*, solo que en este punto el radio de la Base *A* no puede comunicarse con el radio del *drone*, ya que el *drone* tiene un marcador que solo se activará una vez que tenga el documento completo como lo indica el diagrama de la Figura 5.18.

Entonces el radio de la Base *A* intentará saber el *RSSI* pero no lo conseguirá si el *drone* no ha interactuado con la Base *B*; todo esto es para evitar interferencias de la Base *A* con la Base *B*. Los resultados de medición del *RSSI* obtenidos desde la Base *B* se pueden ver en la Figura 5.19, donde se observan dos *RSSI* distintos: el actual y el filtrado. El actual se usa con el propósito de verificar que todo funciona bien ya que el programa responde solo al *RSSI* filtrado. El *RSSI* filtrado usa un filtro para atenuar la curva de la gráfica y evitar bajar del umbral inferior.

El valor de *RSSI* deseado debe mantenerse por encima del umbral inferior, pero debe pasar el umbral superior para iniciar la transmisión, estos se determinaron a partir de una calidad de enlace Rx2 y una calidad de enlace Rx1 respectivamente (ver Figura 4.7), una vez el *drone* supere el umbral superior iniciará su transmisión. En caso de que el valor de *RSSI* disminuya por debajo del umbral inferior el *drone* dejará de transmitir (Lo ideal es mantenerse entre los dos umbrales o por encima del umbral superior). El *RSSI* de la gráfica (Figura 5.19) se solicita cada 3 segundos y el *drone* se desplaza a gran velocidad ( $4.5\text{ m/s}$ ), por lo que es posible ver “picos” en la gráfica. Estos picos pueden causar que estemos cruzando un umbral constantemente, por ello se usa una histéresis para evitar que el *drone* deje de transmitir con algún movimiento brusco que pueda producir un pico. La histéresis es el espacio en la gráfica que se forma entre el umbral superior e inferior: mientras el *RSSI* se mantenga por encima o debajo de esa región no habrá un cambio en la transmisión. (ver Figura 5.20)

### **Etapa 3 del código.**

Durante esta etapa inicia la primera transmisión que va desde la Base *B* hacia el *drone*. En este punto se considerará que el *drone* siempre se encuentra dentro de la región para transmitir. Para lograr esto el *drone* se posiciona cerca de la base y baja de una altura de 10 m a solo 4 m cambiando a modo de vuelo *hover* (ver Figura 5.21). El *drone* no vuela por encima de la estación debido a que se comprobó que volar por encima disminuye la calidad del enlace.

Una vez que el *drone* se acerca a la base *B* en términos del *RSSI* se activa una función de la

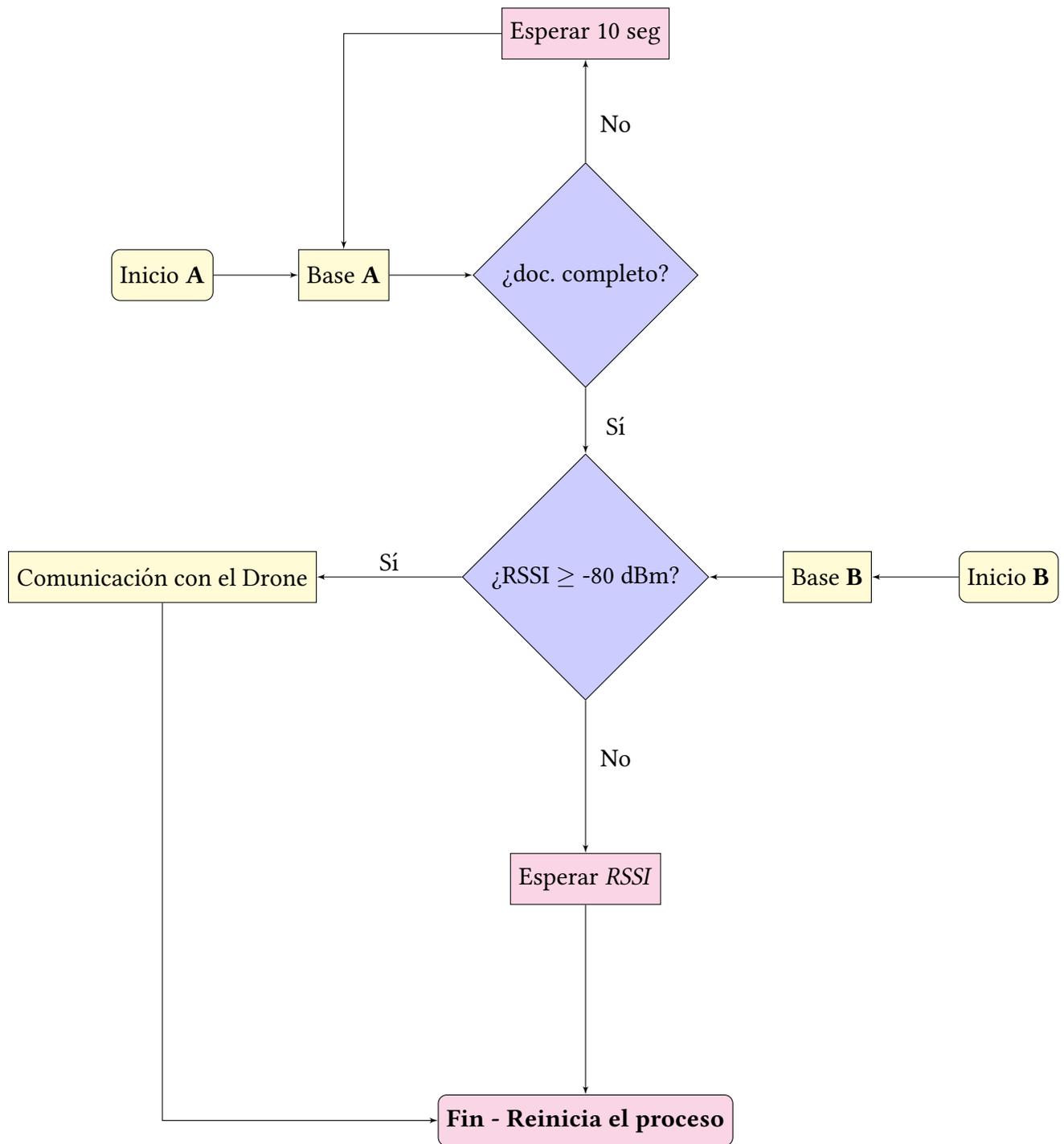


Figura 5.18: Comunicación de las bases con el radio.

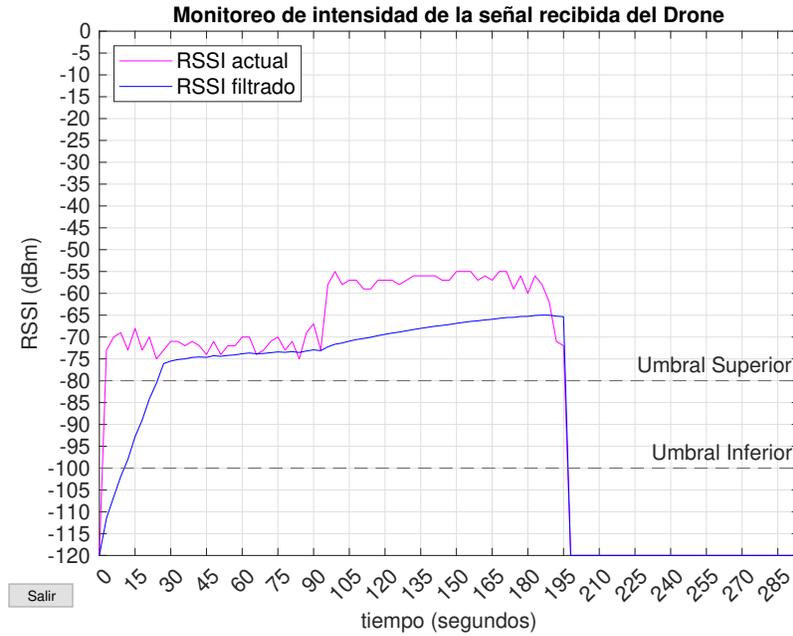


Figura 5.19: Gráfica del RSSI obtenido del *drone* hacia la Base B.

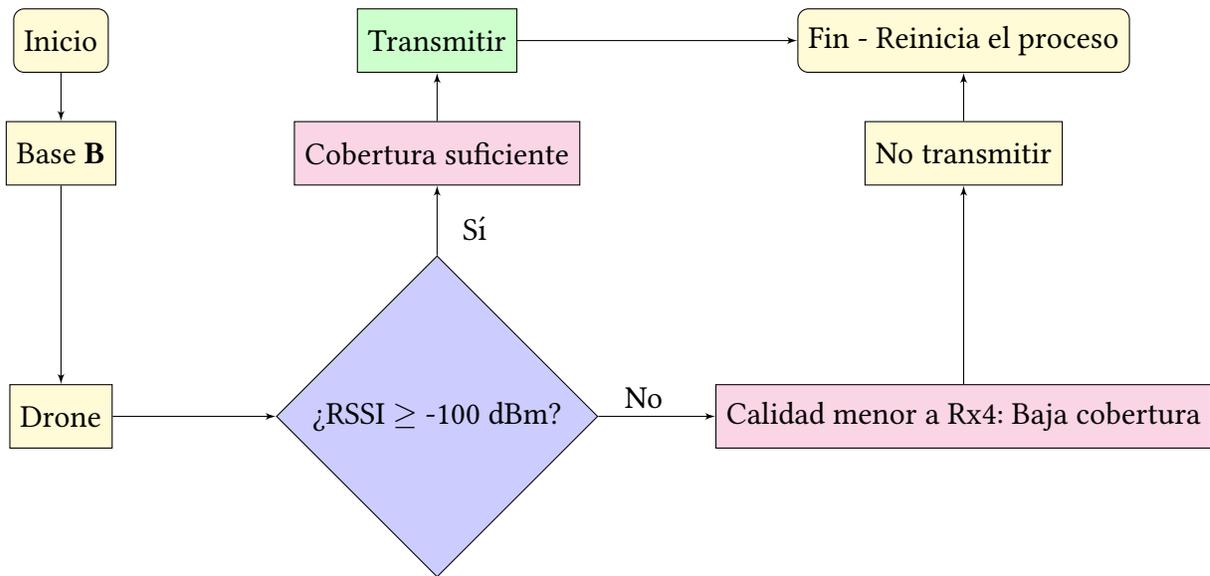


Figura 5.20: Inicio de la transmisión de Base B con el *drone*.

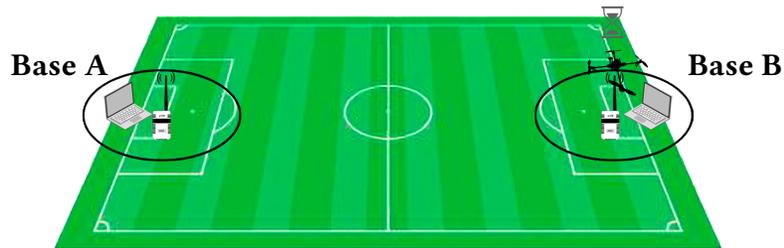


Figura 5.21: Etapa 3: Inicia la transmisión.

base donde comienza a enviar las tramas hacia el *drone*. Este envío de tramas es el protocolo de comunicación que se adaptó a la misión. Se puede distinguir que la Base B va mandando todos los paquetes y el *drone* solo contesta con acuses, esto es para mantener la parte del procesamiento en los equipos de procesador *Intel* en lugar de usar la *Raspberry* que controla el radio del *drone*. Dichos paquetes tienen una estructura definida anteriormente (ver Tabla 5.6). Estas tramas son las que se enviarán hacia el *drone* para que pueda recrear el documento y poder llevarlo hacia la otra base.

#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13-62	#63	#64	#65
IT	L	Do <sub>H</sub>	Do <sub>L</sub>	Dd <sub>H</sub>	Dd <sub>L</sub>	I	N	R	M	M	M	M	C <sub>H</sub>	C <sub>L</sub>	FT

Tabla 5.6: Posición de campos de la trama de comunicación enviada de la Base al *Drone*.

En la Tabla 5.6 se observan 4 campos de color naranja referentes al campo de mensaje; se divide de esta forma porque en el Capítulo 4 se menciona que se toman algunos de sus campos para implementar el protocolo de comunicación. Hablaremos primero de los campos #10, #11 y #12 ya que el resto será el contenido del mensaje (campos que van del #13 al #62, representando los 50 caracteres).

Los campos naranjas antes del contenido del mensaje (ver Tabla 5.6) tienen ahora un valor significativo para el protocolo, ya que permitirá distinguir entre diferentes tipos de mensaje (ver Tabla 5.7).

El campo #10 es el tipo de mensaje, el cual servirá para distinguir el propósito del mismo (ver Tabla 5.8), puede ser para notificar si se trata de contenido, acuse o incluso notificar que

Campo #	Nombre
10	Tipo de mensaje
11	Total de mensajes
12	Número de mensaje

Tabla 5.7: Nuevos identificadores en el campo **M**.

el documento se ha completado.

ID	Nombre	Significado
0	Mensaje	El contenido será parte del archivo.
1	Acuse	Se confirma que la trama recibida es correcta.
2	No-acuse	Se notifica que la trama fue incorrecta.
3	Contenido completo	Se informa que el documento está completo.

Tabla 5.8: Tipos de mensaje.

El campo #11 es el total de mensajes; esto sirve para dar a conocer al receptor la cantidad de mensajes que debe recibir y preparé el orden de cada uno. De la misma manera se manda en el campo #12 el número de mensaje; ambos campos tienen el propósito de dar un seguimiento al mensaje enviado y a su almacenamiento. Esto último permite también trazabilidad al saber qué mensaje está siendo mandado y, al mismo tiempo, saber cuál se pierde y mandar el no-acuse correspondiente. Se mandarían los 45 mensajes (aproximadamente) de forma consecutiva y cada uno tendrá una respuesta de acuse. Una vez completado se manda el mensaje de recibido del documento completo, el cual detendrá la comunicación de la Base **B**.

#### **Etapa 4 del código.**

Una vez que el documento esté en el *drone* y se va a preparar el archivo para ser descargado en la Base *A*, aquí el *drone* se trasladará de la Base *B* a la Base *A* (ver Figura 5.22).

La Base *A* va a preguntar al *drone* si posee el documento y ahora que lo tiene entonces comenzará a solicitar su *RSSI* como se muestran los resultados observados en la Figura 5.23.

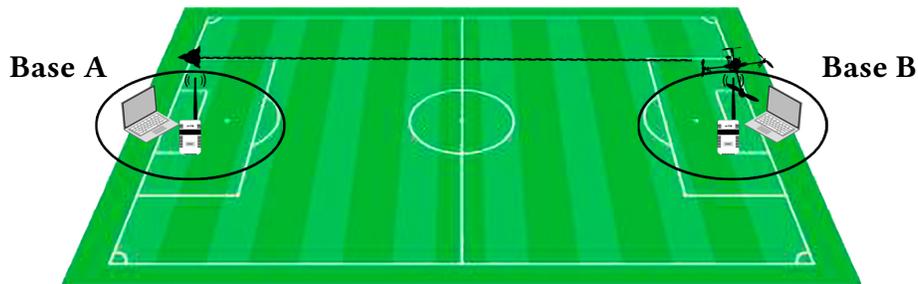


Figura 5.22: Etapa 4: Base A solicita RSSI al dron.

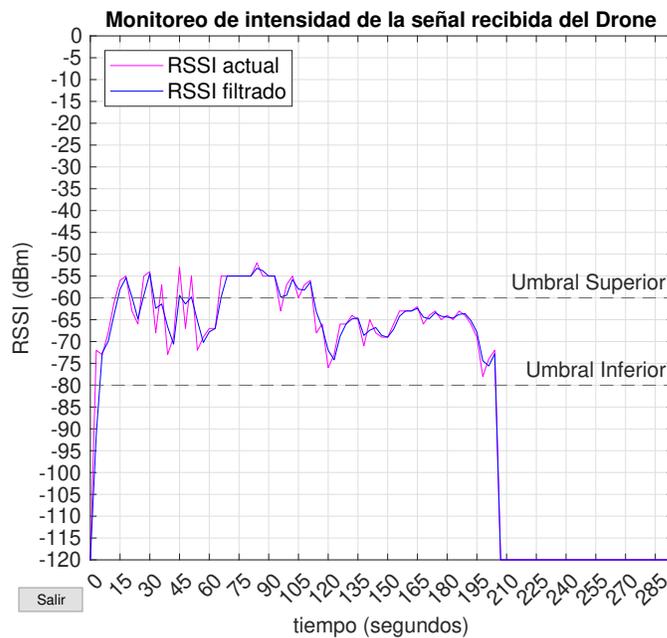


Figura 5.23: Gráficas del RSSI en aproximación a la Base A.

Una vez que el RSSI supera el umbral superior (en este caso ahora es de calidad Rx1, y la histéresis entre Rx1 y Rx2 de la Figura 4.7) entonces comenzará la descarga del documento. En este caso la Base A solicitará el documento al dron. Esto para llevar acabo el procesamiento desde la Base A y no en el dron (esto es conocido como “Piggybacking” en comunicaciones [23]). El proceso consiste en mandar un “acuse” que solicita el mensaje #n y el dron contestará

con ese #n de mensaje.

### Etapa 5 del código.

Para el código esta será la última etapa, esto es porque el documento será descargado en la Base A y posteriormente se desentenderá del *drone*. Para lograr esto, el *drone* entrará en modo *hover* (ver Figura 5.24) bajando de 10 m a 4 m de altura (las antenas son las mismas en todos los radios, por lo que también se pone a un lado de la base) hasta completar la descarga descrita en el paso anterior.



Figura 5.24: Etapa 5 del plan de vuelo.

### 5.4.3. Comparación del RSSI entre A y B

Ambas gráficas del RSSI varían entre sí; lo cual se debe a la programación de ambas estaciones base. Además de que estaban configuradas para empezar a transmitir a diferentes distancias, ya que los umbrales superior e inferior de cada estación eran distintos. Éstos valores se determinaron a partir de las mismas pruebas de campo y se consideró tiempo de acercamiento y alejamiento del *drone*. En la Figura 5.25b se observa cuando cruza por primera vez el umbral

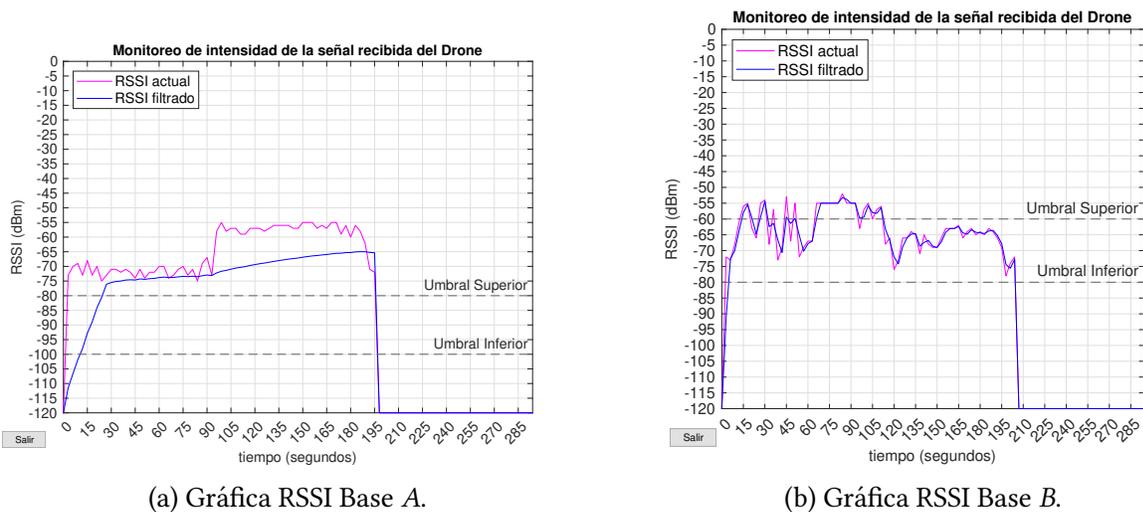


Figura 5.25: Comparación de gráficas.

superior y varía alrededor de 20 dBm, esto se debe a la interferencia que se da en el medio campo donde la estación de la base A intenta comunicarse, después hay una zona estable entre el segundo 75 y el segundo 120, luego por ráfagas de viento el *drone* mueve su posición y la recupera retomando la estabilidad de señal del segundo 120 hasta el 210.

Por otro lado en la Figura 5.25a se observa que el *drone* no da su posición (se puede observar la diferencia entre las gráficas ya que en la Figura 5.25b se muestra un incremento del RSSI en los primeros 15 segundos y la Figura 5.25a es instantáneo), donde en ni interactúa con el radio de la base A, y por lo tanto, no grafica porque aun no tiene el documento completo y cuando recibe el primer dato el *drone* ya lleva más de medio camino a su posición en la estación A, al segundo 90 se ve como el *drone* mejora la señal al descender de los 10 metros a 4 metros (la calidad del enlace está dada por la cercanía de los radios, por ello la calidad aumenta cuando la altura disminuye, se puede observar en la Figura 5.25a en el segundo 90, y en la Figura 5.25b

en el segundo 60).

## 5.5. Comentarios de las pruebas de campo

Las pruebas de campo se realizaron en la cancha de fútbol del CINVESTAV, durante estas pruebas se determinaron los umbrales superiores e inferiores. Se realizaron varios vuelos de prueba durante diferentes semanas bajo diversos horarios y condiciones climáticas como: sol y lluvia. En el caso de los días soleados el rendimiento de los equipos bajo considerablemente, haciendo que la transmisión tardará más de un minuto treinta segundos que se tenían considerados. En el caso de las pruebas nocturnas en su mayoría había viento fuerte que demandaba mayor potencia de los motores del dron para mantener su posición en modo de vuelo *hover* por lo que los tiempos de prueba eran menores. Por último, los días con lluvia muy ligera solo se voló con viento de lluvia en donde se percibía la humedad y los vientos fuertes, pero se suspendieron las pruebas ante el incremento de lluvia debido a que los equipos no son aptos para ésta.

Tras todas las pruebas realizadas, también se detectaron algunas zonas de interferencia debido a ubicaciones de otros radios en las instalaciones del Cinvestav, pero se modificó el alcance de los radios y los algoritmos para poder filtrar mensajes externos a la prueba, por ello los umbrales de la estación base A y B son diferentes.

## 5.6. Resultados

El documento fue transmitido de la Base B al *drone* en 1 minuto, y menor a un minuto en la descarga de la Base A, por lo que toda la operación dura menos de 5 minutos. El tiempo de transferencia fue el adecuado dentro del tiempo de uso de la batería (15 min), además, en el documento final está libre de errores. Finalmente, se corroboró que el texto añadido por los nodos al texto original también estaba contenido en el documento que se descargó en la Base A. A partir de todo lo anterior se considera que los resultados fueron exitosos.



---

# Conclusiones

---

## 6.1. Resumen de los Objetivos de Investigación

Recordando los objetivos de la investigación:

### 6.1.1. Objetivo General

Comunicar dos estaciones base utilizando una memoria portátil en un vehículo aéreo como transporte de la información, haciendo uso de radios con el protocolo IEEE 802.11 para transferir los datos entre el vehículo aéreo y las estaciones base.

### 6.1.2. Objetivos Particulares

- Adaptar un prototipo de vehículo aéreo autónomo para desempeñar las funciones de establecer la conectividad en redes de comunicación.
- Adaptar un protocolo de comunicación y sobreponerlo a un protocolo establecido de mayor esfuerzo para asegurar la fiabilidad de nuestra información.
- Establecer un protocolo de comunicación con el porcentaje mínimo de error en la recepción.
- Programar una ruta de vuelo en el prototipo de vehículo aéreo para que vuele de forma autónoma de portería a portería en el campo de fútbol en las instalaciones del CINVESTAV.

- Validar experimentalmente el uso del vehículo aéreo autónomo transmitiendo información con el protocolo de comunicación adaptado.

### 6.2. Cumplimiento de los Objetivos

El objetivo general es la culminación de completar de forma exitosa todos los demás objetivos particulares, para ello fue necesario adaptar el prototipo inicial del vehículo aéreo autónomo y establecer una ruta de portería a portería de forma autónoma sin complicaciones. Esto se logró tomando en cuenta las adaptaciones para tener un vehículo robusto que pueda permanecer estable en modo de vuelo *hover* incluso ante situaciones de ráfagas de viento.

Por otra parte la adaptación del protocolo de comunicación fue esencial para lograr una transmisión libre de errores y de forma eficiente. Todo esto se implementó usando los módulos de radio que transmiten con mejor esfuerzo y adaptando el protocolo de *RQ Idle* a la misión. La presentación experimental fue validada por los sinodales y asesores de tesis, en donde se cumplen todos los objetivos anteriores y se valida el objetivo general.

### 6.3. Interpretación de los Resultados

Con los resultados obtenidos se verificó que la prueba era libre de errores en la transferencia del archivo, así como los tiempos de vuelo necesarios (5 minutos para completar toda la operación) y la duración de la batería del *drone* (15 minutos aproximadamente) y se determinó si el protocolo empleado cumplía con la transmisión de forma exitosa.

### 6.4. Limitaciones del Estudio

Se contaba con un prototipo inicial de vehículo aéreo autónomo así como con los radios a emplear, por lo que se partió de modelos ya establecidos para adaptarlos a nuestro propósito. Sin embargo, los modelos que se emplearon cubrieron de forma satisfactoria las pruebas que se realizaron y los objetivos que se plantearon desde un inicio.

Por otra parte el tipo de estaciones (PC) que se emplearon junto a los radios limitaron la

capacidad de procesamiento por lo que se desarrolló un sistema de transmisión donde el trabajo de procesamiento se realizara en tierra y evitar el procesamiento en el *drone*.

## 6.5. Recomendaciones para Futuras Investigaciones

Durante el trabajo de tesis se empleó un cuadricóptero; sin embargo, la mejor opción sería un *drone* de ala fija ya que el consumo de energía es demasiado alto para toda la operación. Otras observaciones son el trabajo en conjunto de diferentes *drones*, se puede usar un enjambre para desarrollar un protocolo a mayor escala con una red tipo malla.

También como trabajo a futuro se busca desarrollar un protocolo de comunicación donde se comuniquen la *pixhawk* del *drone* con la *raspberry* para coordinar el plan de vuelo en tiempo real con el envío de información.

Posteriormente, el uso de un *drone* de ala fija aumentaría el tiempo de vuelo y las distancias recorridas, ya que estos *drones* son más eficientes respecto al uso de energía durante el vuelo.

## 6.6. Implicaciones Prácticas y Teóricas

Durante las pruebas de campo hubo accidentes que requerían reparaciones y ralentizaron el proceso de investigación, así como señales que interferían con la transmisión, todo esto en contraste con las pruebas realizadas en el laboratorio en donde el procedimiento de transferencia era más rápido y sin interferencias que ralentizaran el proceso (se filtraron los radios que se detectaron en las instalaciones, para evitar recibir sus tramas) ya que al solo usar las PC y la Raspberry de forma estacionaria y cercana, se tenía una calidad de enlace de Rx1 de principio a fin, por lo que los tiempos de transferencia del archivo fueron menores a un minuto tanto en carga como descarga.

## **6.7. Conclusión General**

En conclusión, esta tesis representa un paso importante hacia el desarrollo de soluciones innovadoras para el establecimiento de comunicaciones mediante vehículos aéreos autónomos. La combinación de tecnologías de comunicación y vehículos aéreos autónomos presenta un gran potencial para mejorar la conectividad en áreas de difícil acceso, proporcionando una alternativa viable y escalable a las infraestructuras de comunicación tradicionales.

## A.1. Cuadricóptero

Un cuadricóptero es un vehículo aéreo no tripulado (UAV) o drone con cuatro rotores, cada uno con un motor y una hélice. Un cuadricóptero puede controlarse manualmente o ser autónomo. También se llama helicóptero cuadricóptero o cuadrirotor. Pertenece a una clase más general de vehículos aéreos llamados multicópteros o multirrotores.

### A.1.1. Modelo dinámico del Cuadricóptero

#### Formulación de Euler-Lagrange.

Para obtener el modelo dinámico se supone al cuadricóptero como un cuerpo rígido en el espacio, sujeto a una fuerza principal (empuje) y tres momentos (pares). Todo el desarrollo para llegar a las siguientes ecuaciones se encuentra en [14] y se omite el procedimiento ya que el modelo dinámico de un cuadricóptero es algo muy explorado a la fecha pero se incluyen las ecuaciones para demostrar que no hubo modificaciones. [24][25]

De la dinámica de traslación se tienen las ecuaciones finales:

$$m \begin{bmatrix} d^2x/dt^2 \\ d^2y/dt^2 \\ d^2z/dt^2 \end{bmatrix} = \begin{bmatrix} u_z(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \\ u_z(\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \\ u_z(\cos \phi \cos \theta) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}. \quad (\text{A.1})$$

De la dinámica de rotación se tiene:

$$\mathbb{J}\ddot{\eta} = \tau - C(\eta, \dot{\eta}). \quad (\text{A.2})$$

Finalmente el modelo matemático del cuadricóptero está dado por (A.1) y (A.2).

### **A.1.2. ArduPilot**

ArduPilot es un sistema de piloto automático fiable, versátil y de código abierto que admite muchos tipos de vehículos: multicopteros, helicópteros tradicionales, aviones de ala fija, barcos, submarinos, vehículos exploradores y mucho más. El código fuente es desarrollado por una gran comunidad de profesionales y entusiastas. Al ser de código abierto es una gran opción para proyectos ya que permite la manipulación del código y adaptarlo a las necesidades del proyecto en cuestión.

### **A.1.3. Pixhawk**

*Pixhawk* es un proyecto independiente de hardware abierto que ofrece diseños de hardware de piloto automático de bajo costo y alta gama a las comunidades académica, industrial y de aficionados.

### **A.1.4. Mission Planner**

*Mission Planner* es una estación de control de tierra para Avión, *Copter* (Helicópteros de 1 o más rotores) y *Rover* (Vehículo terrestre). Sólo es compatible con Windows. *Mission Planner* puede ser utilizado como una utilidad de configuración o como un suplemento de control dinámico para su vehículo autónomo.

## A.2. Radios TTR - MRAPC

Los **MRAPC** (Módulos de radio a computadora personal) son capaces de comunicarse con una computadora personal. Los módulos de radio poseen un conector mini-USB. Un cable con conector USB en un extremo (para conectar a la PC) y mini-USB en el otro (para conectar al módulo de radio) puede usarse para hacer la conexión. Desde la PC los nodos MRAPC son identificados como puertos seriales virtuales, los cuales transmiten utilizando el puerto USB pero cuyas señales cumplen con las especificaciones lógicas de una transmisión serial. Para lograr esto, los módulos de radio poseen un puente de comunicación serial a USB basado en el circuito integrado CP2104 de *Silicon Laboratories*.

### A.2.1. Tabla de estructura de trama

Campo	Significado	Longitud (bits)	Descripción
<b>IT</b>	Inicio de trama	8	Indica el inicio de la trama. Su valor siempre será 0xFE.
<b>L</b>	Longitud de trama	8	Indica el número de bytes que conforman toda la trama de transmisión sin incluir los campos IT y FT.
<b>Do</b>	Dirección MAC del dispositivo origen	16	Es la dirección MAC de la PC. El valor de la MAC será 0xFXXX, donde las X se eligen de forma arbitraria (a excepción de 0xFFFF por ser la dirección reservada para tramas de difusión) por el usuario pero cuidando que no exista otra PC conectada a la red con la misma MAC. El primer byte debe ser siempre 0xF.
<b>Dd</b>	Dirección MAC del dispositivo destino	16	Es la dirección MAC del MRAMF o MRAPC que se encuentra conectado al puerto USB de la PC.
<b>I</b>	Instrucción	8	Este campo indica el tipo de trama que se está transmitiendo. Diferentes tipos de tramas indican distintas instrucciones y/o solicitudes dirigidas al MRAPC.
<b>N</b>	Número de Secuencia	8	Este campo contiene el número de secuencia del paquete que se envía y sirve para propósitos de control de flujo entre la PC y el MRAPC.

Tabla A.1: Descripción de los campos de la trama enviada de la PC al radio. Parte 1

<b>Campo</b>	<b>Significado</b>	<b>Longitud (bits)</b>	<b>Descripción</b>
<b>R</b>	Reservado	8	Este es un campo reservado. Algunas de las instrucciones pueden emplearlo para transmitir información de estado.
<b>M</b>	Mensaje	Variable (Máximo 712)	Representa información a ser entregada al dispositivo destino. La estructura de este campo depende del campo I de instrucción. Actualmente se encuentra limitado a una longitud máxima de 89 bytes.
<b>C</b>	Código de redundancia cíclica (CRC)	16	Son los 16 bits del código de redundancia cíclica calculados en base a los campos L, Do, Dd, I, N, R, M de la trama, es decir, se excluyen los campos de inicio y fin de trama.
<b>FT</b>	Final de trama	8	Indica el final de la trama. Su valor siempre será 0xEF.

Tabla A.2: Descripción de los campos de la trama enviada de la PC al radio. Parte 2

### A.2.2. Tabla de instrucciones PC a MRAPC

Existen diversos tipos de tramas de comunicación que pueden ser enviadas al radio. La distinción entre ellas se realiza mediante el campo de instrucción **I**.

<b>Campo I</b>	<b>Descripción de Operación</b>
<b>0x02</b>	Trama de Registro. Esta trama sirve para que la PC le anuncie al MRAPC que desea formar parte de la red y hacer uso de la misma para transmitir información. Este registro es necesario para que el MRAPC pueda anunciarle a toda la red que un nuevo dispositivo (en este caso la PC) se encuentra en la red y otros conozcan su existencia. Si la PC no se registra, el MRAPC no procesará sus solicitudes de envío de información, por lo que el registro es obligatorio.
<b>0xFF</b>	Trama de respuesta a las instrucciones PING provenientes del MRAPC. Cada vez que la PC recibe una instrucción de tipo PING proveniente del MRAPC, debe contestar con esta trama para mantenerse registrada. Si no responde a tres tramas PING consecutivas, el MRAPC le dará de baja de la red y tendrá que volverse a registrar para permanecer nuevamente. Este mecanismo de mantenimiento del registro es importante para brindarle a toda la red la capacidad de determinar la entrada y salida de los dispositivos conectados a la misma.

Tabla A.3: Instrucciones de la PC al MRAPC. Parte 1

<b>Campo I</b>	<b>Descripción de Operación</b>
<b>0x04</b>	Instrucción para enviar una trama de datos. Mediante esta instrucción la PC le indica al MRAPC que el campo M contiene un paquete de datos. El MRAPC debe encaminar este paquete a su destino utilizando el protocolo de encaminamiento de la red.
<b>0x0A</b>	Trama de Baja. Esta trama sirve para que la PC le anuncie al MRAPC que desea darse de baja de la red.
<b>0x0B</b>	Trama de Programación de Reinicio. El MRAPC cuenta con la posibilidad de que el usuario programe las condiciones bajo las cuales debe reinicializarse. Esta versatilidad es importante en la etapa de depuración del sistema, debido a que no se desea que el MRAPC se reinicialice cuando se realizan pruebas bajo condiciones especiales. Se hace notar que por defecto las tres condiciones de reinicio descritas en la columna de la derecha se encuentran habilitadas. También se hace notar que la programación del MRAPC se elimina cuando el mismo se desconecta de la fuente de energía, es decir, esta programación no es permanente.
<b>0x05</b>	Trama de Eco. El MRAPC responde con la trama que se le envía.
<b>0x07</b>	Trama de Reinicio de la Red. El objetivo de esta trama es el de brindarle al usuario la posibilidad de reinicializar completamente la red. Cuando esta trama se emite el MRAPC calendariza su reinicio 5 segundos en el futuro. Además reenvía la trama a todos los nodos de radio de la red para que la red completa sea reiniciada. El reinicio de cada nodo de la red se ejecuta independientemente de la configuración de su reinicio programada con la instrucción 0x0B.

Tabla A.4: Instrucciones de la PC al MRAPC. Parte 2

### A.2.3. Tabla de instrucciones MRAPC a PC

<b>Campo I</b>	<b>Descripción de Operación</b>
<b>0x03</b>	Trama Lista de Dispositivos.
<b>0xFE</b>	Trama de reconocimiento a la solicitud de registro.
<b>0x04</b>	Instrucción para enviar una trama de datos.
<b>0x01</b>	Trama de verificación de dispositivo conectado (trama PING).
<b>0xF6</b>	Trama de reconocimiento de la solicitud de baja.
<b>0xF5</b>	Trama de reconocimiento de la programación de reinicio.
<b>0xFC</b>	Trama de reconocimiento a la solicitud de envío de datos.
<b>0x05</b>	Trama de Eco.
<b>0xF9</b>	Trama de reconocimiento de la solicitud de reiniciar la red.

Tabla A.5: Instrucciones del MRAPC a la PC.

### A.3. Protocolo *RQ Idle*

El protocolo de *RQ Idle* consiste en enviar un mensaje del canal primario (P) al canal secundario (S), y si el mensaje es correcto responder con un acuse (*acknowledgement, ACK*), cuando el canal primario recibe el acuse entonces procede a mandar el siguiente mensaje. Cuando un mensaje es enviado se inicia un temporizador, en donde si el canal secundario no ha respondido, entonces el mensaje se reenvía hasta que el acuse sea recibido como se muestra en la Figura A.1.

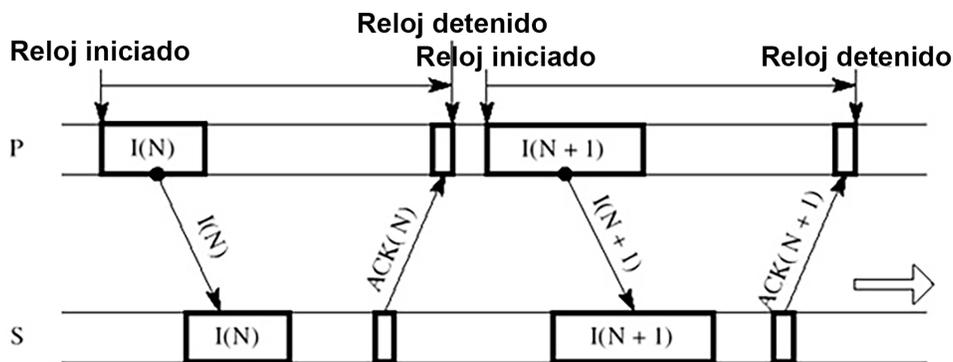


Figura A.1: Protocolo *RQ Idle*.

#### A.3.1. *RQ Idle*: Solicitud explícita

En la variante de solicitud explícita se agrega una nueva respuesta, el "no acuse" (*Non-acknowledgement, NACK*), esta es la variante de la cual partieron las adaptaciones al protocolo empleado en el trabajo de tesis, la diferencia aquí es que si el canal secundario llega a recibir un mensaje pero este llega con algún error, entonces se manda el no acuse e inmediatamente se reenvía el mensaje en lugar de esperar el tiempo habitual de reenvío. Esto consume más recursos pero se ahorra mucho tiempo ya que elimina los tiempos de espera muertos al recibir mensajes con errores.

**A.3.2. Poema enviado**

The Phoenix and the Turtle  
BY WILLIAM SHAKESPEARE [26]

Let the bird of loudest lay  
On the sole Arabian tree  
Herald sad and trumpet be,  
To whose sound chaste wings obey.

But thou shrieking harbinger,  
Foul precurrer of the fiend,  
Augur of the fever's end,  
To this troop come thou not near.

From this session interdict  
Every fowl of tyrant wing,  
Save the eagle, feather'd king;  
Keep the obsequy so strict.

Let the priest in surplice white,  
That defunctive music can,  
Be the death-divining swan,  
Lest the requiem lack his right.

And thou treble-dated crow,  
That thy sable gender mak'st  
With the breath thou giv'st and tak'st,  
'Mongst our mourners shalt thou go.

Here the anthem doth commence:  
Love and constancy is dead;  
Phoenix and the Turtle fled  
In a mutual flame from hence.

So they lov'd, as love in twain  
Had the essence but in one;  
Two distincts, division none:  
Number there in love was slain.

Hearts remote, yet not asunder;  
Distance and no space was seen  
'Twixt this Turtle and his queen:  
But in them it were a wonder.

So between them love did shine  
That the Turtle saw his right  
Flaming in the Phoenix' sight:  
Either was the other's mine.

Property was thus appalled  
That the self was not the same;  
Single nature's double name  
Neither two nor one was called.

Reason, in itself confounded,  
Saw division grow together,  
To themselves yet either neither,

Simple were so well compounded;

That it cried, "How true a twain  
Seemeth this concordant one!  
Love has reason, reason none,  
If what parts can so remain."

Whereupon it made this threne  
To the Phoenix and the Dove,  
Co-supremes and stars of love,  
As chorus to their tragic scene:

threnos

Beauty, truth, and rarity,  
Grace in all simplicity,  
Here enclos'd, in cinders lie.

Death is now the Phoenix' nest,  
And the Turtle's loyal breast  
To eternity doth rest,

Leaving no posterity:  
'Twas not their infirmity,  
It was married chastity.

Truth may seem but cannot be;  
Beauty brag but 'tis not she;  
Truth and beauty buried be.

To this urn let those repair  
That are either true or fair;  
For these dead birds sigh a prayer.

# Referencias

---

- [1] Dà-Jiāng. «Innovations Science and Technology Co. Ltd. Agrasmg-1.» (2006), dirección: <https://www.dji.com/es/mg-1> (visitado 30-07-2018).
- [2] U. Ali, H. Cai, Y. Mostofi e Y. Wardi, «Motion and communication co-optimization with path planning and online channel prediction,» en *2016 American Control Conference (ACC)*, IEEE, 2016, págs. 7079-7084.
- [3] D. B. Licea, M. Bonilla, M. Ghogho y M. Malabre, «UAV Trajectory Planning for Delay Tolerant Communications,» en *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, 2019, págs. 4166-4171.
- [4] H. Shakhathreh, A. H. Sawalmeh, A. Al-Fuqaha et al., «Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges,» *Ieee Access*, vol. 7, págs. 48 572-48 634, 2019.
- [5] D. B. Licea, G. Silano, M. Ghogho y M. Saska, «Optimum Trajectory Planning for Multi-Rotor UAV Relays with Tilt and Antenna Orientation Variations,» 2021.
- [6] T. H. Cox, C. J. Nagy, M. A. Skoog, I. A. Somers y R. Warner, «Civil UAV Capability Assessment,»
- [7] J. Fink, A. Ribeiro y V. Kumar, «Robust control of mobility and communications in autonomous robot teams,» *IEEE Access*, vol. 1, págs. 290-309, 2013.
- [8] Z. Sarris y S. Atlas, «Survey of UAV applications in civil markets,» en *IEEE Mediterranean Conference on Control and Automation*, 2001, pág. 11.
- [9] F. Halsall, *Data communications, computer networks and open systems*, 4.<sup>a</sup> ed. Addison Wesley Longman Publishing Co., Inc., 1996.
- [10] TTR, *Especificación de Tramas de Comunicación entre una PC y un Radio*. Telemática, Telemetría y Radiofrecuencia S.A. de C.V.
- [11] A. O. Hero, *telecommunications media*, 2015. dirección: <https://www.britannica.com/topic/telecommunications-media>.
- [12] J. A. Fleming, *The principles of electric wave telegraphy and telephony*. Longmans, Green, 1919.
- [13] A. A. Ghirardi, *Radio Physics Course*, 2.<sup>a</sup> ed. Radio & Technical publishing Co., 1933.

- [14] B. Sánchez, L. Ángel, E. Estrada Cruz et al., «Síntesis de un esquema de linealización robusta por realimentación del estado estructural para un cuadricóptero,» 2020.
- [15] S. de Relaciones Exteriores, *Requisitos para el ingreso de drones a México*, 2023. dirección: <https://embamex.sre.gob.mx/>.
- [16] A. D. Team, *Autotune*, 2023. dirección: <https://ardupilot.org/copter/docs/autotune.html>.
- [17] A. D. Team, *Ardupilot.org*, 2023. dirección: <https://ardupilot.org/copter/docs/>.
- [18] A. J. Carfang, N. Wagle y E. W. Frew, «Improving data ferrying by iteratively learning the radio frequency environment,» en *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2014, págs. 1182-1188.
- [19] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.
- [20] B. Lutkevich y K. Gerwig, *What is a network node? - definition from searchnetworking*, oct. de 2021. dirección: <https://www.techtarget.com/searchnetworking/definition/node>.
- [21] A. D. Team, *Ardupilot.org*, 2023. dirección: <https://ardupilot.org/planner/docs/mission-planner-overview.html>.
- [22] V. M. E. Llori, «Modelado de consumo energético en vehículos aéreos no tripulados de ala rotatoria,» 2021.
- [23] A. S. Tanenbaum, *Computer networks*. Pearson Education India, 2003.
- [24] W. Dong, G.-Y. Gu, X. Zhu y H. Ding, «Modeling and control of a quadrotor UAV with aerodynamic concepts,» en *Proceedings of World Academy of Science, Engineering and Technology*, World Academy of Science, Engineering y Technology (WASET), 2013, pág. 437.
- [25] G. Hoffmann, H. Huang, S. Waslander y C. Tomlin, «Quadrotor helicopter flight dynamics and control: Theory and experiment,» en *AIAA guidance, navigation and control conference and exhibit*, 2007, pág. 6461.
- [26] R. Chester, *Love's Martyr: Or, Rosalins Complaint (1601)*. Trübner, 1878.