

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

### **IMPLEMENTACIÓN DE UN PROTOTIPO PARA LA OBTENCIÓN DE IMÁGENES TÉRMICAS GEO-REFERENCIADAS DESDE EL DRON DJI PHANTOM 3 STANDARD**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN**

**RENATO ANTONIO ESCANDÓN VINCENT**  
renato.escandon@outlook.com

**DIRECTOR: ING. RICARDO XAVIER LLUGSI CAÑAR, MSc.**  
ricardo.llugsi@epn.edu.ec

**CODIRECTOR: ING. ROBIN GERARDO ALVAREZ RUEDA, PhD.**  
robin.alvarez@epn.edu.ec

**Quito, julio 2017**

## **DECLARACIÓN**

Yo, Renato Antonio Escandón Vincent, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

Renato Antonio Escandón Vincent

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Renato Antonio Escandón Vincent, bajo mi supervisión.

---

**Ing. Ricardo Xavier Llugsí, MSc.**  
**DIRECTOR DEL TRABAJO DE TITULACIÓN**

---

**Ing. Robin Gerardo Alvarez Rueda, PhD.**  
**CODIRECTOR DEL TRABAJO DE TITULACIÓN**

## **AGRADECIMIENTOS**

A mis padres, Carmen y Homero por su sacrificio, dedicación y apoyo incondicional que me han demostrado durante toda mi vida. Son la fuente de mi inspiración por su ejemplo de lucha constante.

A mi esposa, por el amor y respeto que me demuestra todos los días.

A los Ingenieros Ricardo Llugsi, Ramón Chango y al Dr. Robin Alvarez, por el apoyo para que este trabajo pueda realizarse.

**Renato Escandón Vincent**

## **DEDICATORIA**

A mi familia entera que siempre me ha apoyado en las decisiones que he tomado en mi vida.

De manera muy especial para mi hija Doménica Escandón que es mi razón de vivir.

**Renato Escandón Vincent**

## CONTENIDO

DECLARACIÓN.....	I
CERTIFICACIÓN.....	II
AGRADECIMIENTOS.....	III
DEDICATORIA.....	IV
CONTENIDO.....	V
ÍNDICE DE FIGURAS .....	VIII
ÍNDICE DE TABLAS .....	XII
ÍNDICE DE FRAGMENTOS DE CÓDIGO.....	XIII
RESUMEN.....	XIV
PRESENTACIÓN .....	XV
<b>CAPÍTULO 1. FUNDAMENTOS TEÓRICOS.....</b>	<b>1</b>
1.1 Luz Infrarroja.....	1
1.2 Imágenes Infrarrojas .....	2
1.2.1 Representación de imágenes infrarrojas .....	3
1.3 Streaming de Video.....	4
1.3.1 Funcionamiento .....	4
1.3.2 Tipos de Streaming.....	5
1.4 Red ad hoc.....	6
1.4.1 Elementos.....	6
1.4.2 Ventajas.....	6
1.4.3 Desventajas .....	7
1.5 Dron .....	7
1.5.1 Historia .....	8
1.5.2 Componentes .....	9
1.5.3 Tipos de Drones .....	12
1.6 Cámara Térmica .....	13
1.6.1 Componentes .....	13
1.6.2 Aplicaciones.....	14
1.7 Dispositivo GPS (Global Position System).....	16
1.7.1 Funcionamiento .....	17
1.7.2 Protocolo NMEA 0183 .....	17
1.8 Tarjeta Raspberry Pi .....	18

1.9	Sistema Operativo Raspbian .....	20
CAPÍTULO 2. DISEÑO DEL PROTOTIPO .....		21
2.1	Sistema Aéreo .....	21
2.1.1	Dron.....	22
2.1.2	Tarjeta Raspberry Pi 3.....	24
2.1.3	Tarjeta Inalámbrica .....	27
2.1.4	GPS UBLOX NEO -7M.....	27
2.1.5	Cámara Térmica.....	29
2.1.6	Batería .....	31
2.1.7	Armazón .....	32
2.1.8	Software .....	33
2.2	Sistema Tierra.....	35
2.2.1	Computador Portátil.....	35
2.2.2	Software .....	36
2.2.3	Base de Datos .....	36
2.3	Diseño Prototipo.....	37
2.3.1	Diseño Sistema Aéreo .....	37
2.3.2	Diseño Sistema Tierra .....	44
2.4	Direccionamiento IP .....	48
CAPÍTULO 3. Implementación y Pruebas del Prototipo.....		49
3.1	Configuración de Hardware .....	49
3.1.1	Configuración de Raspberry Pi 3 .....	49
3.1.2	Configuración Cámara Térmica .....	52
3.1.3	Configuración módulo gps .....	56
3.1.4	Configuración Red Ad Hoc .....	58
3.2	Implementación de Software.....	61
3.2.1	Implementación Software del Sistema Aéreo .....	61
3.2.2	Implementación de Software de Sistema Tierra .....	68
3.3	Construcción del Sistema Aéreo.....	74
3.3.1	Armazón poliestireno expandido.....	74
3.3.2	Armazón plástico .....	76
3.4	Pruebas de funcionamiento .....	77
3.4.1	Pruebas de funcionamiento cancha Escuela Politécnica Nacional...	78

3.4.2 Pruebas en el Volcán Pululahua.....	84
CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES .....	90
4.1 CONCLUSIONES .....	90
4.2 Recomendaciones .....	92
REFERENCIAS BIBLIOGRÁFICAS.....	93
ANEXOS.....	95

## ÍNDICE DE FIGURAS

Figura 1.1 Rango Infrarrojo [1] .....	1
Figura 1.2 Imágenes Infrarrojas [2] .....	2
Figura 1.3 Visualización de pérdidas de energía [3].....	3
Figura 1.4 Paleta de colores [3].....	3
Figura 1.5 Elementos del streaming .....	4
Figura 1.6 Red Ad Hoc.....	6
Figura 1.7 Guerra de 1849 [4] .....	8
Figura 1.8 Avión bomba GB-1 Glide [4].....	9
Figura 1.9 Chasis de dron [5] .....	10
Figura 1.10 Motor del dron [6] .....	10
Figura 1.11 Hélices del dron [7].....	11
Figura 1.12 Baterías del dron DJI [7].....	11
Figura 1.13 Drones según sus alas [8].....	12
Figura 1.14 Cámara térmica [9].....	14
Figura 1.15 Triangulación de satélites [10].....	17
Figura 2.1 Partes del sistema aéreo.....	21
Figura 2.2 DJI PHANTOM 3 STANDARD [13] .....	22
Figura 2.3 Control remoto [14].....	23
Figura 2.4 Partes de la tarjeta Raspberry Pi 3 [15].....	25
Figura 2.5 Diagrama de pines GPIO [16] .....	25
Figura 2.6 LAN Inalámbrica 802.11 b/g/n [17].....	27
Figura 2.7 GPS Ublox Neo-7M [18].....	28
Figura 2.8 Diagrama de Conexión GPS .....	29
Figura 2.9 Cámara térmica Flir Lepton.....	29
Figura 2.10 Pines cámara térmica.....	30
Figura 2.11 Batería externa [19].....	32
Figura 2.12 Protección de dispositivos electrónicos del Sistema Aéreo.....	32
Figura 2.13 Sistema Operativo Raspbian [20].....	33
Figura 2.14 Sistemas Operativos para Raspberry Pi [20] .....	34
Figura 2.15 Partes del Sistema Tierra.....	35
Figura 2.16 Laptop DELL 3421 [21] .....	36
Figura 2.17 Diagrama de bloques del hardware aéreo .....	37

Figura 2.18 Diagrama de bloques software aéreo.....	38
Figura 2.19 Diagrama de conexión .....	40
Figura 2.20 Diagrama casos de uso Sistema Aéreo .....	41
Figura 2.21 Armazón Sistema Aéreo .....	43
Figura 2.22 Diagrama general Sistema Tierra.....	44
Figura 2.23 Diagrama casos de uso Sistema Tierra .....	45
Figura 2.24 Diagrama de base de datos .....	46
Figura 2.25 Sketch interfaz principal .....	47
Figura 2.26 Sketch interfaz de almacenamiento .....	47
Figura 2.27 Sketch interfaz ítem específico .....	48
Figura 3.1 Aplicación NOOBS [20] .....	50
Figura 3.2 Archivos de la aplicación NOOBS .....	50
Figura 3.3 Log's de encendido de la tarjeta Raspberry Pi.....	51
Figura 3.4 Instalación sistema operativo Raspbian .....	51
Figura 3.5 Usuario tarjeta Pi.....	52
Figura 3.6 Conexión cámara térmica .....	53
Figura 3.7 Conexión cámara térmica real .....	53
Figura 3.8 Utilidad raspi-config.....	54
Figura 3.9 Habilitar interfaces SPI – I2C .....	54
Figura 3.10 Pruebas cámara térmica .....	55
Figura 3.11 Salida del puerto ttyAMA0.....	57
Figura 3.12 Aplicativo cgps -s .....	58
Figura 3.13 Creación Red AD HOC .....	58
Figura 3.14 Configuración red ad hoc .....	59
Figura 3.15 Parámetros de configuración .....	59
Figura 3.16 Dirección IP Sistema Tierra.....	60
Figura 3.17 Pruebas de conectividad .....	60
Figura 3.18 Fecha y Hora UTC .....	62
Figura 3.19 Servicio Servicio_GPS.service .....	66
Figura 3.20 Servicio Servicio_Camara.service.....	66
Figura 3.21 Prueba Programa_GPS.py.....	67
Figura 3.22 Prueba Programa_Camara.py .....	67
Figura 3.23 Interfaz principal .....	68

Figura 3.24 Mensaje de conexión .....	69
Figura 3.25 Botones del interfaz principal .....	70
Figura 3.26 Registro de capturas térmicas.....	72
Figura 3.27 Conexión base de datos.....	72
Figura 3.28 Consulta base de datos.....	73
Figura 3.29 Formulario Datos Imagen.....	73
Figura 3.30 Armazón de espuma flex.....	75
Figura 3.31 Prueba de vuelo 1 .....	75
Figura 3.32 Raspberry dentro de armazón.....	76
Figura 3.33 Posición cámara térmica .....	76
Figura 3.34 Armazón Plástico .....	77
Figura 3.35 Cancha de fútbol de la Escuela Politécnica Nacional.....	78
Figura 3.36 Sitio de pruebas de funcionamiento .....	78
Figura 3.37 Escuela Politécnica Nacional: Captura 1.....	79
Figura 3.38 Escuela Politécnica Nacional: Captura 2.....	80
Figura 3.39 Borde de Cancha .....	80
Figura 3.40 Escuela Politécnica Nacional: Captura 3.....	81
Figura 3.41 Escuela Politécnica Nacional: Captura 4.....	82
Figura 3.42 Escuela Politécnica Nacional: Captura 5.....	83
Figura 3.43 Escuela Politécnica Nacional: Captura 6.....	83
Figura 3.44 Sector Pululahua .....	84
Figura 3.45 Medición de ubicación.....	85
Figura 3.46 Coordenadas sector Pululahua .....	85
Figura 3.47 Prueba en neblina: Captura 1 .....	86
Figura 3.48 Prueba con neblina: Captura 2.....	86
Figura 3.49 Prueba en neblina: Captura 3 .....	87
Figura 3.50 Prueba con neblina: Captura 4.....	88
Figura 3.51 Prueba con neblina: Imagen Google Earth .....	88
Figura 3.52 Prueba con neblina: Captura 5.....	89
Figura 3.53 Prueba con neblina: Conectividad.....	89

## ÍNDICE DE TABLAS

Tabla 1.1 Sentencias protocolo NMEA.....	18
Tabla 1.2 Tipos de Raspberry Pi [12] .....	19
Tabla 2.1 Dimensiones, peso y capacidades del dron DJI PHANTOM 3 STANDARD [13] .....	22
Tabla 2.2 Características de la batería del dron DJI PHANTOM 3 STANDARD [13] .....	23
Tabla 2.3 Características del control remoto del dron DJI PHANTOM 3 STANDARD [13] .....	23
Tabla 2.4 Distribución de Pines.....	31
Tabla 2.5 Especificaciones Técnicas Laptop [21].....	36
Tabla 2.6 Caso de uso: Iniciar Sistema Aéreo .....	41
Tabla 2.7 Caso de uso: Capturar Imágenes Térmicas .....	42
Tabla 2.8 Caso de uso: Capturar Datos GPS.....	42
Tabla 2.9 Caso de uso: Sincronizar .....	42
Tabla 2.10 Caso de uso: Enviar datos a Sistema Tierra .....	43
Tabla 2.11 Caso de uso: Iniciar Sistema Tierra.....	45
Tabla 2.12 Caso de uso: Recibir mensajes del Sistema Aéreo.....	45
Tabla 2.13 Caso de uso: Ver imágenes capturadas.....	46
Tabla 2.14 Caso de uso: Guardar en base de datos.....	46
Tabla 2.15 Direccionamiento IP .....	48

## ÍNDICE DE FRAGMENTOS DE CÓDIGO

Fragmento de Código 3.1 Librerías para Script “Programa_GPS” .....	61
Fragmento de Código 3.2 Conexión a módulo GPS. ....	61
Fragmento de Código 3.3 Procesamiento de información GPS .....	62
Fragmento de Código 3.4 Transformación de fecha y hora a zona local .....	62
Fragmento de Código 3.5 Escritura de archivo gps.txt.....	63
Fragmento de Código 3.6 Librerías script Programa_Camara .....	63
Fragmento de Código 3.7 Servidor TCP .....	64
Fragmento de Código 3.8 Lectura de archivo gps.txt.....	64
Fragmento de Código 3.9 Captura de la foto térmica.....	65
Fragmento de Código 3.10 Procesamiento de la imagen “Foto_Termica.jpg” .....	65
Fragmento de Código 3.11 Envío de información a Sistema Tierra .....	65
Fragmento de Código 3.12 Comunicación con Sistema Aéreo .....	69
Fragmento de Código 3.13 Mensaje del tipo string .....	69
Fragmento de Código 3.14 Fragmentación del mensaje.....	70
Fragmento de Código 3.15 Código botón Conectar .....	71
Fragmento de Código 3.16 Código botón Capturar.....	71
Fragmento de Código 3.17 Método Guardalmagen .....	71
Fragmento de Código 3.18 Código para enlistar los registros de guardados.....	74
Fragmento de Código 3.19 Código para obtener la imagen detalle .....	74

## RESUMEN

En el campo de la búsqueda y rescate existen situaciones que no permiten al personal que integra estos grupos cumplir a cabalidad con su trabajo porque muchas veces, la visión se puede ver afectada por aspectos externos como por ejemplo el clima, postes, vegetación o porque las posibles víctimas están en lugares inaccesibles.

En el presente trabajo de titulación se presenta el diseño e implementación de un sistema de captura imágenes térmicas geo-referenciadas tomadas desde un dron. El prototipo pretende ser un equipo de apoyo para inspecciones en terrenos de difícil tránsito.

El sistema está compuesto por un Sistema Aéreo y un Sistema Tierra. Los componentes se comunican inalámbricamente utilizando el estándar 802.11g en la banda de 2.4Ghz. Las funciones del Sistema Aéreo son obtener y enviar la información al Sistema Tierra. Las funciones del Sistema Tierra son recibir, presentar, almacenar los datos importantes para el usuario y permitir revisar las imágenes de interés que se hayan capturado.

La captura de imágenes se realizó mediante la utilización de la cámara térmica FLIR Lepton, la cual detecta la emisión de rayos infrarrojos del espectro electromagnético, construye imágenes con una paleta de colores mediante software los cuales corresponden a diferentes niveles de radiación térmica o temperatura.

El Sistema Aéreo envía al Sistema Tierra las coordenadas geográficas mediante la utilización de un módulo GPS.

El prototipo se somete a pruebas de funcionamiento en la cancha de fútbol de la Escuela Politécnica Nacional y en un lugar con neblina para poder verificar el tipo de imágenes que puede entregar y cómo se comporta en este tipo de ambientes. Para la prueba con neblina se escogió los alrededores del volcán Pululahua. El prototipo realizado se presenta como punto de partida para realizar estudios posteriores donde se busque el reconocimiento de objetos desde una imagen térmica.

## PRESENTACIÓN

En el presente trabajo se implementó un sistema que permite capturar imágenes térmicas geo-referenciadas y almacenarlas en una base de datos para posteriormente permitir la revisión de estos registros. El proceso para la implementación de este estudio técnico se presenta en cuatro capítulos que se detalla a continuación.

En el capítulo 1 se describen las características de las imágenes infrarrojas, *streaming* de video, redes ad hoc y características técnicas de los dispositivos a utilizarse en el prototipo. Además, se describe las características del sistema operativo Raspbian.

En el capítulo 2 se presentan los requerimientos que se utilizaron en el prototipo tanto de hardware como de software. Adicionalmente se presenta el diseño del prototipo que consta de un Sistema Aéreo y un Sistema Tierra.

En el capítulo 3 se explica el proceso de configuración del hardware del sistema. Se presenta los módulos de software implementados para el funcionamiento del prototipo. Se realiza las pruebas de funcionamiento a cada uno de los elementos del prototipo, así como de comunicación entre los sistemas y los módulos de software que intervienen en el prototipo. Finalmente se presenta imágenes térmicas tomadas en la cancha de la Escuela Politécnica Nacional e imágenes que se capturaron en las cercanías del volcán Pululahua con la presencia de neblina.

En el capítulo 4 se detalla las conclusiones y recomendaciones obtenidas de la implementación del prototipo y que podrían ser de ayuda en el desarrollo de estudios posteriores para potencializar el presente trabajo de titulación.

## CAPÍTULO 1. FUNDAMENTOS TEÓRICOS

En el presente capítulo se aborda la base teórica de los elementos que integran el prototipo, se describen las características de las imágenes infrarrojas, *streaming* de video, redes ad hoc y de los dispositivos a utilizarse en el prototipo. Además, se describe brevemente puntos principales del sistema operativo Raspbian.

### 1.1 LUZ INFRARROJA

La luz infrarroja es una parte del espectro electromagnético (EM). La longitud de onda de la luz infrarroja se ubica entre 0,7 y 1000 micrómetros, este rango está por encima de la longitud de onda de la luz visible es por ello por lo que el ojo humano no puede observar la luz infrarroja. En la Figura 1.1 se puede observar el lugar que ocupa la luz infrarroja en el espectro electromagnético.

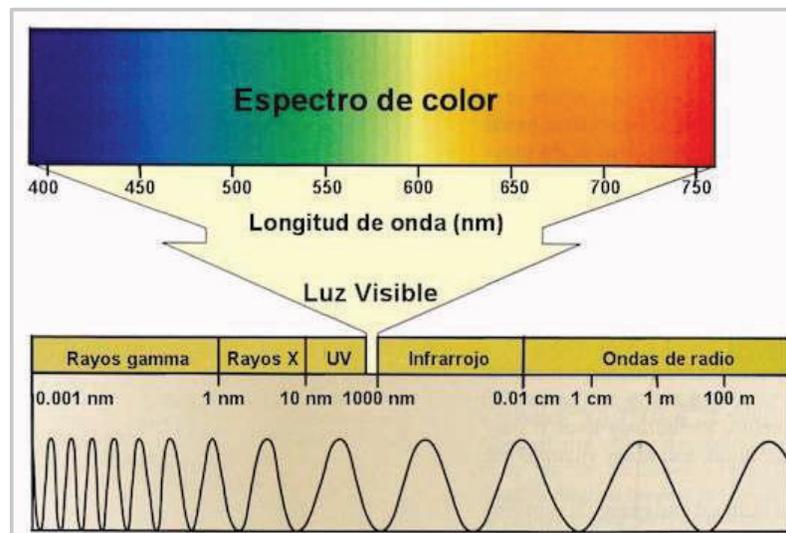


Figura 1.1 Rango Infrarrojo [1]

Todo cuerpo que tenga una temperatura superior a los 0 grados Kelvin o -273.15 grados Celsius emite radiación infrarroja, de acuerdo con esto se puede decir que hasta cuerpos fríos como el hielo emiten este tipo de radiación.

La radiación infrarroja se puede clasificar de acuerdo con su longitud de onda en:

- Infrarrojo cercano (0,7 - 2,5  $\mu\text{m}$ )
- Infrarrojo medio (2,5 - 50  $\mu\text{m}$ )
- Infrarrojo lejano (50 -1000  $\mu\text{m}$ )

## 1.2 IMÁGENES INFRARROJAS

Son fotografías que se obtienen con dispositivos que capturan los rayos del canal infrarrojo. Este tipo de imágenes muestran la distribución de temperatura que tiene la superficie de un objeto.

Dichas imágenes utilizan una pequeña fracción del rango infrarrojo el cual comprende entre  $0,9 \mu\text{m}$  a  $14 \mu\text{m}$  y a su vez esta porción del espectro electromagnético se divide en tres bandas de longitud de onda que son:

1. Onda Corta:  $0,9 - 1,7 \mu\text{m}$
2. Onda Mediana:  $3 - 5 \mu\text{m}$
3. Onda Larga:  $7 - 14 \mu\text{m}$

En la Figura 1.2 se puede observar una fotografía con luz visible y otra que se obtuvo a partir de la radiación infrarroja que emiten los vasos. En la imagen de la derecha se puede ver la distribución de temperatura sobre la superficie de los vasos. Mediante el procesamiento de la imagen infrarroja se puede obtener la temperatura del cuerpo sin tener ningún tipo de contacto.

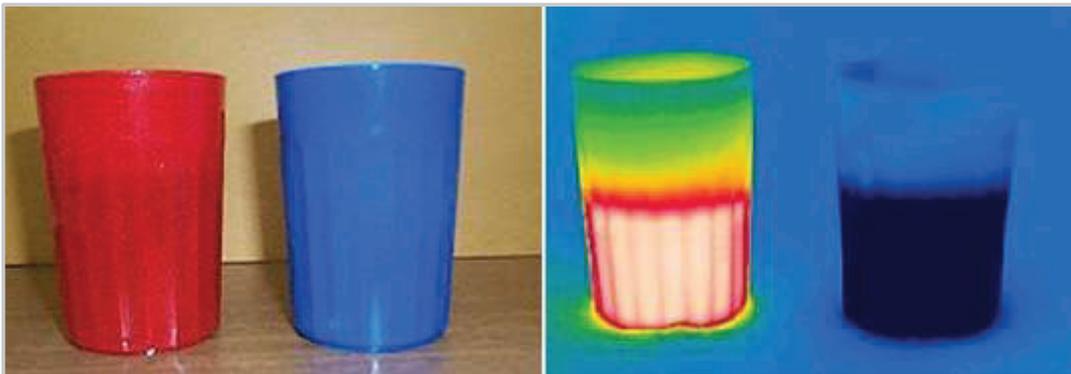


Figura 1.2 Imágenes Infrarrojas [2]

Las imágenes infrarrojas se pueden aplicar en diferentes campos, por ejemplo:

- Vigilancia
- Búsqueda nocturna.
- Detección de fugas de energía.
- Construcción de hornos.
- Localización de fallas eléctricas.

En la Figura 1.3 se presentan tres imágenes infrarrojas, en este caso se están utilizando las imágenes infrarrojas para la inspeccionar una edificación con humedad.



Figura 1.3 Visualización de pérdidas de energía [3]

### 1.2.1 REPRESENTACIÓN DE IMÁGENES INFRARROJAS

Las imágenes infrarrojas al ser procesadas pueden ser representadas por tres tipos de paletas de colores de acuerdo con el uso.

1. Escala de grises
2. Paleta de color hierro
3. Paleta de color arcoíris

En la Figura 1.4 se puede observar las tres paletas de colores. La primera imagen se encuentra representada en escala de grises y no permite diferenciar con facilidad los detalles de la imagen. La segunda imagen está representada por la paleta de colores hierro que representa las superficies con mayor temperatura con los colores que presenta el hierro cuando es fundido. La tercera imagen está representada por la paleta de colores arcoíris que utiliza los colores oscuros para las superficies con baja temperatura y colores vivos para representar altas temperaturas.



Figura 1.4 Paleta de colores [3]

## 1.3 STREAMING DE VIDEO

Es un flujo continuo de video que es entregado por un emisor y reproducido por uno o varios receptores. Para que el *streaming* funcione el receptor debe contar con un *buffer*<sup>1</sup> donde se almacena el contenido para mostrarlo continuamente. El *streaming* se diferencia con una descarga normal de archivos porque se debe esperar que los archivos se descarguen por completo para poder utilizarlo y en *streaming* la descarga es continua por lo que se transfiere al mismo ritmo que se muestra.

### 1.3.1 FUNCIONAMIENTO

El funcionamiento del *streaming* está conformado por los elementos que se presentan en la Figura 1.5.



Figura 1.5 Elementos del streaming

#### 1.3.1.1 Conexión

El punto inicial de una transmisión de *streaming* es cuando el sitio remoto se conecta al servidor y este empieza a entregarle el contenido.

#### 1.3.1.2 Almacenamiento

Cuando el usuario empieza a recibir el contenido, la información se guarda dentro de un espacio de memoria para posteriormente ser visualizada.

---

<sup>1</sup> *Buffer*: es un espacio de memoria que se reserva para contener información digital temporal.

### 1.3.1.3 Reproducción

La etapa de reproducción se presenta una vez que el espacio de memoria se llena con información, el contenido es presentado en un interfaz de visualización.

### 1.3.1.4 Velocidad

La velocidad de conexión es un elemento muy importante para el funcionamiento del *streaming*, si existe una degradación en la velocidad, el contenido se detiene porque se tiene que esperar que el espacio de memoria se llene nueva información.

## 1.3.2 TIPOS DE STREAMING

El *streaming* de contenido se puede clasificar de acuerdo con el modo de trabajo del servidor que emita el contenido es por ello que se tiene los siguientes tipos:

- Descarga progresiva

Se presenta cuando el proceso de visualización y descarga se presenta al mismo tiempo.

Si el *buffer* se queda sin contenido el usuario tendrá que esperar a que el *buffer* se llene con información y el reproductor cliente reproduzca el contenido descargado.

- Contenido en vivo

Es una transmisión similar a la de un canal de televisión, el contenido es capturado por el emisor e inmediatamente es transmitido por un servidor de *streaming* al receptor.

El contenido que está siendo presentado en vivo podría grabarse y convertirse en un *streaming* progresivo para usuarios que no lo pudieron ver cuando fue transmitido.

- Contenido bajo demanda

El contenido es capturado, almacenado y es reproducido por el cliente cuando él lo decida, es decir el receptor interactúa con el emisor. El emisor tiene las opciones de reproducir, parar adelantar o retroceder.

## 1.4 RED AD HOC

Es una red inalámbrica en la cual los *end point* se conectan directamente uno con otro utilizando señales de radio es decir no hay un punto de interconexión o *access point* como se puede apreciar en la Figura 1.6.

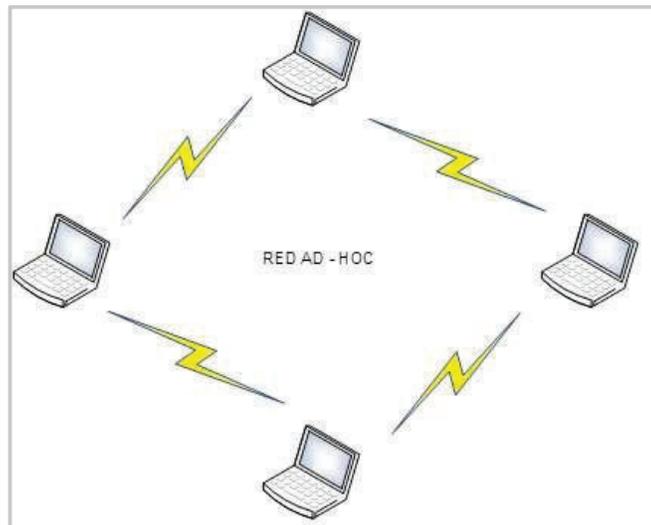


Figura 1.6 Red Ad Hoc

El tráfico de información se transfiere directamente entre los equipos finales utilizando la capa de enlace y no están sujetos a un equipo de jerarquía superior que centralice el intercambio de información como por ejemplo un conmutador.

Son redes que se encuentra aisladas es decir no tienen conexión a otro tipo de redes por lo que simplemente son utilizadas para el intercambio de información entre los integrantes de la red ad hoc.

### 1.4.1 ELEMENTOS

Para conformar una red ad hoc es necesario los siguientes elementos:

- Mínimo dos máquinas.
- Tarjetería inalámbrica.
- Drivers de las tarjetas inalámbricas.

### 1.4.2 VENTAJAS

- Mínima configuración.

- Los datos son directamente enviados desde emisor al receptor.
- No existe punto central.
- Se puede conectar diferentes tipos dispositivos.

### 1.4.3 DESVENTAJAS

- Podrían ingresar integrantes no deseados a la red.
- Los integrantes de la red ad hoc no tienen acceso a internet por ser una red aislada.
- Existen colisiones por la ocupación del medio.
- Podría existir incompatibilidad en tarjetería inalámbrica.
- El área de cobertura podría ser limitada.
- Los usuarios no pueden acceder a otras redes.

## 1.5 DRON

Vehículo Aéreo No Tripulado (VANT) o dron, es una aeronave operada remotamente, de un tamaño relativamente pequeño que realiza diversas actividades. El manejo de estos vehículos aéreos se puede presentar desde una estación base en tierra o pueden manejarse por códigos o algoritmos. En sus inicios se los utilizó para fines militares, pero en la actualidad existen diversas aplicaciones civiles en las cuales están siendo usadas dichas naves.

Los drones pueden ser utilizados en diferentes campos, por ejemplo:

- Cartografía
- Logística
- Seguridad
- *Delivery*
- Investigaciones climatológicas
- Fotografía
- Cinematografía
- Búsqueda de personas desaparecidas
- Prevención y control de incendios
- Movilidad y tráfico

### 1.5.1 HISTORIA

El uso de vehículos no tripulados se viene dando desde tiempos muy anteriores a la comunicación inalámbrica o la robótica, se tiene conocimiento que los austriacos en la guerra de 1849 usaron cerca de dos centenares de globos aerostáticos no tripulados, estos contenían bombas que fueron detonadas en la ciudad de Venecia por esta razón se puede decir que es uno de los primeros usos de vehículos no tripulados [4].

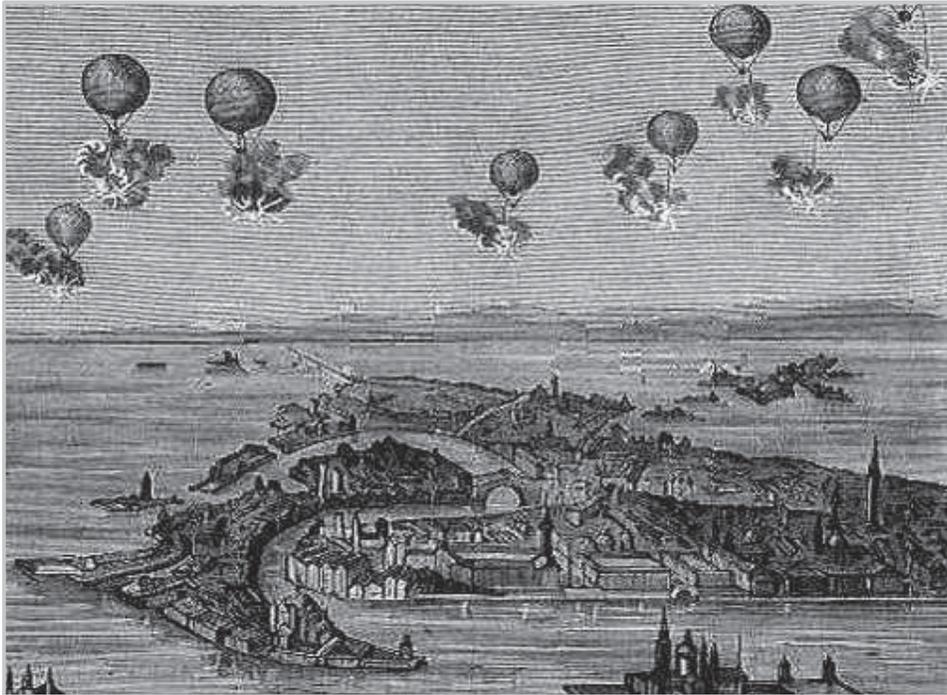


Figura 1.7 Guerra de 1849 [4]

Luego de este primer antecedente se puede detallar los siguientes hitos históricos donde se ve el crecimiento de los vehículos no tripulados y como se ve desde su inicio solo fueron con fines militares [4].

En 1916 el militar inglés Archibald M. Low supervisó el ensamblaje de aviones no tripulados equipados con bombas. Por otra parte, el Dr. Wilhelm Von Siemens entre 1915 y 1918 trabajó en un torpedo planeador que era guiado por medio de ondas de radio [4].

Luego de varios años de trabajo y prototipos de vehículos no tripulados fallidos en la segunda guerra mundial, en 1940 los Estados Unidos de América se desarrolló

el “GB-1 Glide”, que era un planeador controlado remotamente que fue usado para ataque contra los alemanes por los estadounidenses [4].



**Figura 1.8 Avión bomba GB-1 Glide [4]**

En mayo de 1964, EE. UU comenzó a considerar el envío de aviones no tripulados para reemplazar sus modelos U-2 que fueron utilizados en misiones de espionaje sobre Cuba. Las “Luciérnagas” operados por EE. UU. fueron posteriormente utilizados para vigilancia en las denominadas zonas sensibles [4].

Con todo lo que había logrado a nivel militar, Estados Unidos vio que era un arma letal y decidieron seguir invirtiendo en estas tecnologías hasta la actualidad. Se han realizado pruebas entre aviones piloteados por humanos y vehículos no tripulados en los cuales han salido ganadores las máquinas, además con el desarrollo de la informática y las comunicaciones inalámbricas cada vez aparecen drones más sofisticados y con esto ha crecido las aplicaciones de estos vehículos no tripulados [4].

## **1.5.2 COMPONENTES**

### **1.5.2.1 Chasis**

Es la parte del dron al cual se fijan todos los elementos como, por ejemplo: motores, hélices, cámara, baterías, etc. En la Figura 1.9 se puede observar un chasis de dron. Los materiales más utilizados en la construcción del chasis son de fibra de carbono, fibra de vidrio, kevlar, etc.



Figura 1.9 Chasis de dron [5]

### 1.5.2.2 Motores

Son aquellos elementos que sirven para que el vehículo no tripulado se eleve. Los motores se diferencian uno de otro por el empuje que presentan, dicha magnitud se mide en gramos, mientras mayor sea este valor mayor peso podría transportar el dron. Los motores generalmente son diseñados para poder elevar como el doble del peso del dron. En la Figura 1.10 se puede observar un motor de dron.



Figura 1.10 Motor del dron [6]

### 1.5.2.3 Reguladores

Son los encargados de controlar de forma electrónica las revoluciones de los motores. El control se realiza de forma individual ya que esto permite que el dron pueda realizar movimientos como rotaciones o elevaciones.

### 1.5.2.4 Unidad de Control

Es la parte principal del dron ya que este componente controla todos los dispositivos electrónicos que tiene el dron.

Los principales dispositivos que comúnmente maneja la unidad de control son:

- Giróscopo: estabilizador del dron.
- Acelerómetro: mide variación de velocidad del dron.
- Altimetro: mide la variación altura.
- GPS: entrega las coordenadas de donde se encuentre el dron.

#### 1.5.2.5 Hélices

Son el complemento de los motores y permiten dar el empuje al dron para que se eleve, entre más grande sea la hélice se tendrá un mayor empuje y de la misma manera tendrá un mayor consumo de energía. En la Figura 1.11 se puede observar esta parte del dron.



Figura 1.11 Hélices del dron [7]

#### 1.5.2.6 Batería

Es el elemento que brinda energía a los motores para realizar la elevación, pero también energiza a elementos como cámaras y sensores. Las baterías más utilizadas son las fabricadas con polímero de litio (LiPo) por su gran capacidad de amperaje. En la Figura 1.12 se presenta un ejemplo de batería de dron.



Figura 1.12 Baterías del dron DJI [7]

### 1.5.2.7 Control Remoto

Es el componente que sirve para comunicarse con el dron y permite al piloto realizar maniobras. También muestra el nivel de carga de la batería para realizar el descenso si la batería está por descargarse.

### 1.5.3 TIPOS DE DRONES

Los vehículos no tripulados se pueden clasificar de acuerdo con los siguientes aspectos:

#### 1.5.3.1 Según sus alas:

- Dron de alas fijas: alas estáticas y su apariencia es como la de un avión.
- Dron Multirotor: posee varias hélices que giran y debido a esta forma de funcionamiento se pueden mantener en el aire en un solo punto similar a la forma de volar de un helicóptero. Este tipo de vehículos no tripulados se los pueden encontrar con facilidad en el caso que se desee adquirirlos.



Figura 1.13 Drones según sus alas [8]

### **1.5.3.2 Según su control:**

- Dron autónomo: no necesita control remoto se guía mediante sensores.
- Dron controlado remotamente: su manejo es remoto desde una estación en tierra.
- Dron monitorizado: su manejo es mediante algoritmos diseñados para que sigan una ruta precargada.

### **1.5.3.3 Según su uso:**

- Dron militar: Son vehículos no tripulados que contienen dispositivos explosivos y son utilizados usualmente en combates para el bombardeo.
- Dron civil: Son vehículos no tripulados que son utilizados para fines científicos, investigativos, fotográficos o simplemente para el manejo de aficionados.

## **1.6 CÁMARA TÉRMICA**

Una cámara térmica es un dispositivo que percibe la energía de radiación infrarroja, luego procesa la información y finalmente traduce los valores de temperatura a un color, es decir genera una imagen con diferentes colores de acuerdo con la temperatura del objeto enfocado. Los objetos con mayor temperatura tienen más radiación infrarroja y si son captados por una cámara térmica son representados con colores e mayor intensidad como por ejemplo el amarillo, naranja o rojo.

### **1.6.1 COMPONENTES**

Los principales componentes de una cámara térmica son los siguientes:

- a) Lente
- b) Detector
- c) Procesador

En la Figura 1.14 se puede observar los principales componentes de una cámara térmica.

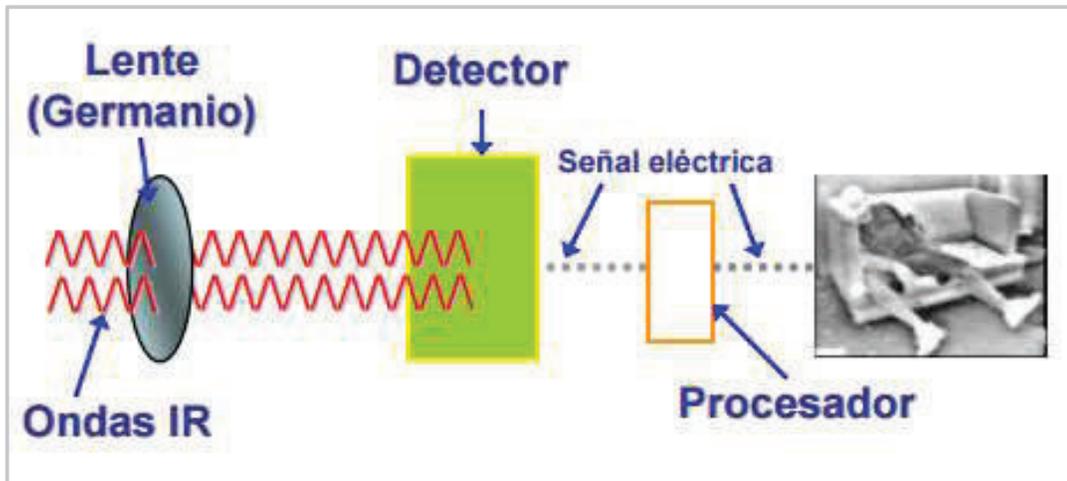


Figura 1.14 Cámara térmica [9]

La lente es la que captura los rayos infrarrojos y los dirige al detector. Generalmente el material con el que están constituidos los lentes es el germanio, ya que el vidrio común repele los rayos infrarrojos.

Detector es el componente que transforma los rayos infrarrojos en señales eléctricas. Existen dos tipos de detectores:

- a) Sensor enfriado
- b) Microbolómetro no enfriado

La diferencia entre estos dos detectores es que los sensores enfriados necesitan estar en temperaturas criogénicas antes de entrar en operación para obtener una mejor la calidad de imagen por este motivo este tipo de sensores resultan demasiado costosos. Estos fueron los primeros sensores que aparecieron, pero en la actualidad los microbolómetros no enfriados son los más utilizados ya que no requieren de ningún proceso para operar.

El procesador es un circuito electrónico que recibe la señal eléctrica enviada por el detector, la procesa y envía una nueva señal eléctrica apta para ser visualizada en una pantalla.

### 1.6.2 APLICACIONES

La capacidad de detectar la radiación térmica de los objetos convierte a este tipo de instrumentos en una herramienta muy útil en diferentes campos.

A continuación, se mencionan algunos campos donde se utiliza la cámara térmica:

a) Industrial

- Calentamiento de componentes eléctricos defectuosos.
- Fricciones en motores o máquinas eléctricas.
- Desequilibrio de cargas.
- Fugas u obstrucciones en conducciones.
- Conexiones mal realizadas.
- Predicción de incendios o daños potenciales.
- Sobrecarga en circuitos eléctricos.
- Niveles en depósitos.
- Puntos críticos en conductos.
- Reacciones químicas peligrosas.

b) Construcción

- Eficiencia energética.
- Fugas de calor.
- Humedades.
- Aislamientos deficientes.
- Escapes.
- Distribución de temperaturas en sistemas de calefacción

c) Medicina

- Detección de lesión por aumento de riego sanguíneo.
- Localización de seres vivos.
- Inmediata detección de temperatura corporal (fiebre por gripe A).
- Análisis médicos no invasivos.
- Detección de nidos o animales invasores en el hogar.

d) Seguridad

- Detección de lesión por aumento de riego sanguíneo.
- Localización de seres vivos.
- Inmediata detección de temperatura corporal.
- Análisis médicos no invasivos.
- Detección de nidos o animales invasores en el hogar.

## 1.7 DISPOSITIVO GPS (GLOBAL POSITION SYSTEM)

Es un equipo que recepta las señales enviadas por los satélites que se encuentran orbitando la tierra y su principal función es determinar la posición de objeto. Un receptor GPS al menos debe captar la señal de 3 satélites para poder calcular la ubicación, adicional a la ubicación con un dispositivo GPS se puede tener información de velocidad, altitud y tiempo.

Los receptores GPS pueden utilizar diferentes sistemas de posicionamiento y entre los cuales se puede mencionar los siguientes:

- GPS

Desarrollado y puesto en marcha por los Estados Unidos y funciona con 24 satélites orbitando alrededor de la tierra a una distancia de 20.000 km. El sistema GPS fue diseñado para que en cualquier punto y altitud el receptor tenga la visibilidad de 4 satélites.

- GLONASS

Es un sistema de posicionamiento satelital con similares características al desarrollado por los Estados Unidos, este sistema fue desarrollado por la Fuerza Espacial Rusa para uso militar y civil, su nombre GLONASS deriva de (*Global'naya Navigatsionnaya Sputnikovaya Sistema*). Actualmente el sistema consta con una constelación de 27 satélites en órbita, 24 se encuentran operacionales, 1 utilizado como repuesto y 2 para pruebas de vuelo. Los satélites de procedencia rusa giran a una distancia de la tierra de alrededor de 19.100 km.

- GALILEO

Desarrollado por la Agencia Espacial de Europa. Es un sistema completamente para uso civil y no es administrado por ningún país. Cuenta con 30 satélites orbitando alrededor de la tierra donde 6 son utilizados como repuesto, estos equipos se encuentran a una altura de la tierra de aproximadamente 23 222 km. Por la tecnología utilizada en este sistema tiene una precisión mucho mayor que el sistema estadounidense.

### 1.7.1 FUNCIONAMIENTO

La ubicación exacta de un receptor se la realiza por medio de la recepción de al menos 3 satélites para poder triangular y de esta forma encontrar la posición del receptor como se puede ver en la Figura 1.15. Los centros de cada círculo representan los satélites, con dos esferas imaginarias se tiene dos puntos de corte y para determinar cuál de los dos puntos representa la ubicación exacta se hace uso de la tercera esfera que corta las dos anteriores esferas, el punto de corte de las tres esferas imaginarias representa la ubicación de objetivo. De esta forma se tiene la posición exacta, para saber la altura a la que se encuentra el receptor se la realiza por medio de un cuarto satélite.

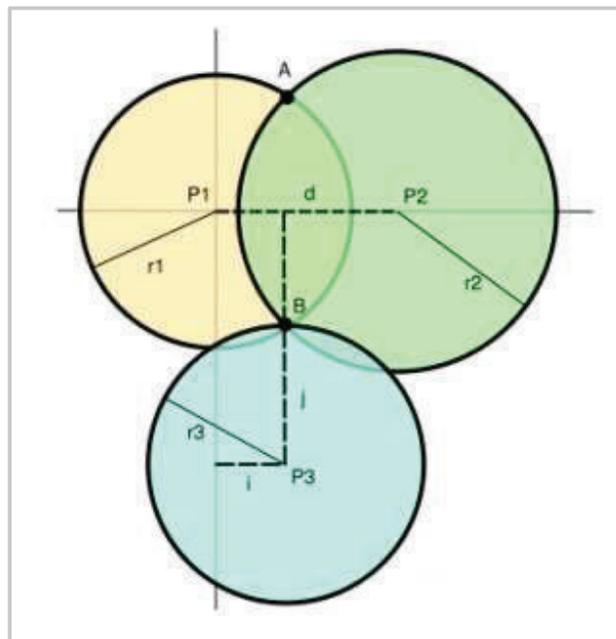


Figura 1.15 Triangulación de satélites [10]

### 1.7.2 PROTOCOLO NMEA 0183

El protocolo NMEA 0183 fue creado por *National Marine Electronics Association* (NMEA) de ahí el nombre del protocolo. Es un protocolo es utilizado para la comunicación entre dispositivos que se lleva a bordo en una embarcación.

Se ha creado la versión NMEA 2000 que principalmente aumenta la velocidad de transmisión, así como la posibilidad de conexión a un puerto USB para la lectura de los datos.

Este protocolo se caracteriza por que transmite sentencias, cada una de las cuales comienza con "\$", y termina con (CR: *Carriage Return*, LF: *Line Feed*), los 2 caracteres que preceden a "\$" son los que identifican el equipo (Para los GPS "GP"), los siguientes 3 caracteres indican la sentencia que se está enviando, hay 3 tipos de sentencias que son de consulta (*Query Sentences*), de origen del equipo (*Proprietary Sentences*) y de envío (*Talker Sentences*) [11].

A continuación, se presenta en la

Tabla 1.1 algunas sentencias utilizadas en el protocolo.

SENTENCIA	DESCRIPCIÓN
\$GPGGA	Sentencia que entrega información de: hora UTC, latitud, longitud, altitud, número de satélites utilizados, calidad de la señal. Contiene 11 campos.
\$GPGLL	Sentencia que entrega información de: hora UTC, longitud, latitud, altitud, número de satélites, calidad de señal. Contiene 8 campos.
\$GPGSA	Sentencia que entrega información de precisión de los datos.
\$GPGSV	Sentencia que entrega información del número de satélites, así como el estado de los mismos.
\$GPRMC	Sentencia que entrega información de tiempo, fecha, curso sobre la tierra y velocidad
\$GPVTG	Sentencia que entrega información de velocidad y curso sobre la tierra.

Tabla 1.1 Sentencias protocolo NMEA

Este protocolo es utilizado por los módulos de GPS que utilizan la comunicación serial y no necesitan de una señal de sincronización para los datos de información.

## 1.8 TARJETA RASPBERRY PI

Es una pequeña computadora de un costo relativamente bajo desarrollada por la universidad de Cambridge en el 2011, inicialmente fue creada para la enseñanza

de informática en las escuelas del Reino Unido posteriormente se empezó a comercializar en todo el mundo. Existen varios modelos de la tarjeta raspberry los cuales se han ido diferenciando por su capacidad de procesamiento, memoria y puertos de conexión.

En la Tabla 1.2 se indican las características principales de los diferentes tipos de Raspberry Pi.

Raspberry Pi	Raspberry Pi A	Raspberry Pi A+	Raspberry Pi B	Raspberry Pi B+	Raspberry Pi 2 B	Raspberry Pi 3 B
SoC (Chip)	Broadcom BCM 2835	Broadcom BCM 2835	Broadcom BCM 2835	Broadcom BCM 2835	Broadcom BCM 2836	Broadcom BCM 2837
CPU	ARM1176JZFS a 700 Mhz	ARM Cortex-A7 cuatro núcleos a 900 Mhz	ARM Cortex-A53 cuatro núcleos a 1,2 Ghz			
GPU	VideoCore IV a 250 Mhz	VideoCore IV a 250 Mhz				
Memoria RAM	256 MB	256 MB	512 MB	512 MB	1 GB	1GB
Entradas de vídeo	Cámara CSI	Cámara CSI				
Salidas de vídeo	HDMI 1.4, conector RCA	HDMI 1.4, conector TRRS	HDMI 1.4, conector TRRS			
Salidas de audio	Jack de 3.5 mm (auriculares), HDMI	Jack de 3.5 mm (auriculares), HDMI				
Conectores USB 2.0	1	1	2	4	4	4
Tarjetas de almacenamiento	SD	microSD	SD	microSD	microSD	microSD
Conexión a red	No	No	Ethernet 10/100 Mbit/sg	Ethernet 10/100 Mbit/sg	Ethernet 10/100 Mbit/sg	Ethernet 10/100 Mbit/sg
Tarjeta inalámbrica	No	No	No	No	No	Si
Interfaces periféricos (GPIO)	8	17	8	17	17	40
Tamaño	85.6 x 56.5 mm	65 x 56.5 mm	85.6 x 56.5 mm	85.6 x 56.5 mm	85.6 x 56.5 mm	85 x 56 mm
Peso	45 gramos	23 gramos	45 gramos	45 gramos	45 gramos	45 gramos

**Tabla 1.2 Tipos de Raspberry Pi [12]**

La tercera generación de tarjetas Raspberry reemplazó a la Raspberry pi 2 modelo B en febrero del 2016. Las mejoras más importantes que presenta la tarjeta Raspberry Pi 3 son las siguientes:

- A 1.2GHz 64-bit ARMv8 CPU
- 802.11n Wireless LAN
- Bluetooth 4.1

- Bluetooth Low Energy (BLE)

## **1.9 SISTEMA OPERATIVO RASPBIAN**

El sistema operativo Raspbian es una derivación del sistema operativo Debian por ende es un sistema gratuito y de libre acceso al público. Este sistema operativo fue concebido para que funcione de forma óptima con la placa Raspberry Pi. Está conformado por 35.000 paquetes pre compilados y especialmente diseñados para el hardware Raspberry Pi. Contiene programas como, por ejemplo:

- Python
- C++
- Scratch
- LibreOffice
- Navegador Web
- Calculadora
- Visor de Imágenes
- Editor de texto

Raspbian no está afiliado a la Fundación Raspberry Pi por el contrario fue creado por un grupo de desarrolladores, interesados en la placa Raspberry Pi con fines educativos.

## CAPÍTULO 2. DISEÑO DEL PROTOTIPO

Para la implementación del prototipo se requiere elementos de hardware como también de software, por lo que en el presente capítulo se detallará las características de los elementos utilizados y finalmente se presentará el diseño del prototipo.

El prototipo está conformado por dos sistemas:

- Sistema Aéreo
- Sistema Tierra

### 2.1 SISTEMA AÉREO

EL Sistema Aéreo del prototipo está conformado por elementos de hardware y de software. En la Figura 2.1 se puede observar las partes que conforman en sistema aéreo.



Figura 2.1 Partes del Sistema Aéreo

A continuación, se detalla las características de cada uno de estos elementos como también sus diagramas de conexión.

### 2.1.1 DRON

El vehículo no tripulado utilizado en el presente prototipo es el DJI PHANTOM 3 STANDARD como se puede observar en la Figura 2.2. La cámara del dron no es parte del prototipo, la función principal del dron es transportar a la tarjeta Raspberry Pi, módulo GPS, batería externa y la cámara térmica.



Figura 2.2 DJI PHANTOM 3 STANDARD [13]

#### 2.1.1.1 Características Técnicas

A continuación, se presentan las principales características técnicas del vehículo aéreo que se utilizará en el trabajo de titulación. Elementos como la cámara y el dispositivo GPS interno que contiene el dron no fueron tomados en cuenta ya que no forman parte del Sistema Aéreo.

En la Tabla 2.1 se presentan los valores de dimensión, peso y capacidad del vehículo no tripulado

Dimensiones, peso y capacidades	
Peso (Incluye batería y propulsores)	1216 g
Longitud diagonal total	350 mm
Velocidad máxima de ascenso	5 m/s
Velocidad máxima de descenso	3 m/s
Velocidad máxima (Sin viento)	16 m/s
Altitud máxima de vuelo	120 m
Tiempo de vuelo máximo	Aprox. 25 min
Temperatura de Operación	0°C a 40°C

Tabla 2.1 Dimensiones, peso y capacidades del dron DJI PHANTOM 3 STANDARD [13]

En la Tabla 2.2 se presentan los valores característicos de la batería que usa el dron para realizar las elevaciones y vuelos.

Batería	
Capacidad de la Batería	4480 mAh
Voltaje	15.2 V
Tipo de batería	Polímero de litio 4S
Consumo de potencia	68 Wh
Peso Neto	365 g
Potencia máxima de carga	100 W

Tabla 2.2 Características de la batería del dron DJI PHANTOM 3 STANDARD [13]

En la Tabla 2.3 se presentan los valores correspondientes a los datos técnicos del de control remoto. El dron para la transmisión de video opera en la banda de 2.4Ghz. En el control remoto también se tiene información del nivel de batería del dron.

Control Remoto	
Frecuencia de Operación	5.725 GHz – 5.825 GHz
Distancia Máxima de transmisión	FCC: 1000 m; CE: 500 m
Capacidad de la Batería	2600 mAh
Potencia de Transmisión	FCC: 20 dBm; CE: 16 dBm
Corriente / Voltaje	600 mA @ 3.7V
Puerto de carga	Micro USB

Tabla 2.3 Características del control remoto del dron DJI PHANTOM 3 STANDARD [13]

El control remoto de la Figura 2.3 monitorea el dron mediante la aplicación DJI Go.



Figura 2.3 Control remoto [14]

### 2.1.2 TARJETA RASPBERRY PI 3

Para el procesamiento de los datos en el Sistema Aéreo se escogió la placa Raspberry Pi 3 modelo B que es la tercera generación de este tipo de mini computadoras y actualmente es el último modelo las tarjetas Raspberry Pi. Este modelo fue lanzado en el año 2016 con mejoras principalmente en el procesador y su tarjetería inalámbrica.

A continuación, se muestra las características principales de la tarjeta Raspberry Pi 3:

- Un procesador de 1.2GHz 64-bit ARMv8.
- Interfaz inalámbrico LAN con el estándar 802.11n.
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)

Además, cuenta con las siguientes características:

- Memoria RAM de 1 GB
- Conector micro USB para fuente de alimentación de 2,5 A
- 1 puerto Ethernet 10/100
- 1 conector de vídeo/audio HDMI
- 1 conector de vídeo/audio RCA
- 1 conector de cámara CSI (*Camera Serial Interface*, Interfaz, Serie para Cámaras)
- 4 x puertos USB 2.0
- 40 pines GPIO (*General Purpose Input/Output*, Entrada/Salida de Propósito General)
- Conector de pantalla DSI (*Display Serial Interface*, Interfaz Serial para Pantalla)
- Ranura de tarjeta microSD
- Dimensiones: 85 x 56 x 17 mm

A continuación, se presenta la Figura 2.4 donde se puede ver las principales partes de la tarjeta computadora Raspberry Pi 3.

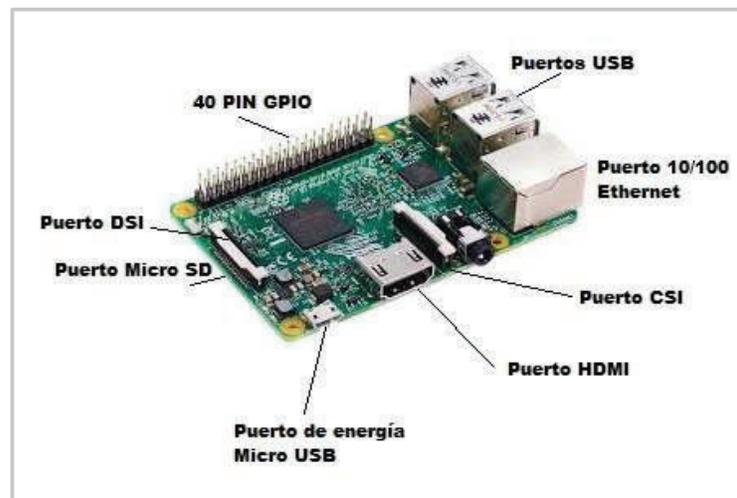


Figura 2.4 Partes de la tarjeta Raspberry Pi 3 [15]

### 2.1.2.1 Pines GPIO (General Purpose Input/Output)

Es un interfaz físico entre la tarjeta Raspberry PI 3 y dispositivos externos. La tarjeta tiene 40 pines, de los cuales 26 son pines GPIO, el resto de pines son para conexiones eléctricas de 3.3v, 5v y pines GND.

En la Figura 2.5 se puede ver el diagrama de pines de la tarjeta Raspberry Pi 3

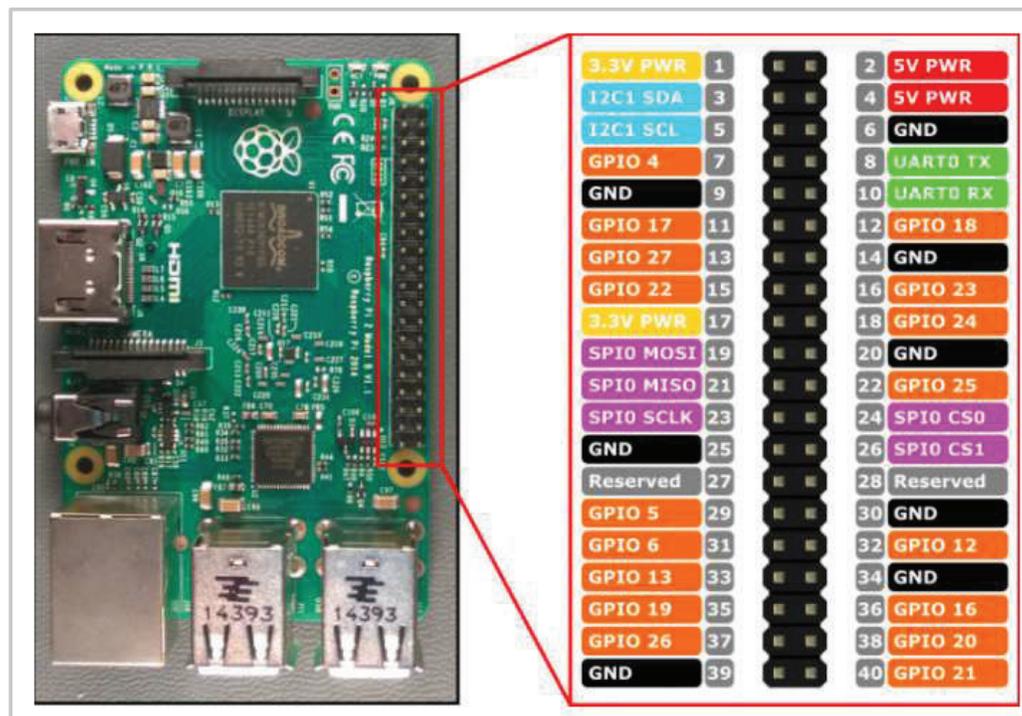


Figura 2.5 Diagrama de pines GPIO [16]

### 2.1.2.1.1 Interfaz I2C (*Inter-Integrated Circuit*)

En 1982 Philips Semiconductors vio la necesidad de crear una norma para la conexión de circuitos integrados ya que la comunicación de varios bloques de circuitos electrónicos se convirtió en un problema. El interfaz fue diseñado para que los datos que intercambian los circuitos electrónicos vayan con una velocidad admisible. El bus I2C consta de 3 líneas:

- SCL (*System Clock*) es la línea de los pulsos de reloj que sincronizan el sistema.
- SDA (*System Data*) es la línea por donde se transfiere los datos entre los dispositivos.
- GND (Masa) común de la interconexión entre todos los dispositivos.

### 2.1.2.1.2 Interfaz UART (*Universal Asynchronous Receiver / Transmisor*)

Es un interfaz de comunicación serie para dispositivos que se comunican enviando dato por dato sin la necesidad de una señal de reloj. Esta forma de comunicación fue remplazada por USB.

En el presente trabajo se utilizó estos pines para la comunicación del GPS con la tarjeta Raspberry Pi. Como se puede ver en la Figura 2.5 los pines 8 transmisión y el pin 10 recepción que corresponden al interfaz UART (*Universal Asynchronous Receiver-Transmitter*)

### 2.1.2.1.3 Interfaz SPI (*Serial Peripheral Interface*)

Es una interfaz de comunicación para integrar circuitos electrónicos. El estándar SPI se utiliza cuando los dispositivos se comunican enviando un flujo de bits en serie con la ayuda de un reloj de sincronización (comunicación sincrónica).

La sincronización y la transmisión de datos se realiza por medio de 4 señales:

- SCLK (*Clock*): Es el pulso que marca la sincronización. Con cada pulso de este reloj, envía un bit.
- MOSI (*Master Output Slave Input*): Salida de datos del *master* y entrada de datos al *slave*.

- MISO (*Master Input Slave Output*): Salida de datos del *slave* y entrada al *master*.
- SS/*Select*: Para seleccionar un *slave*, o para que el *master* le diga al *slave* que se active.

### 2.1.3 TARJETA INALÁMBRICA

Una parte fundamental entre el Sistema Aéreo y el Sistema Tierra es la comunicación inalámbrica es por ello que se aprovechó las mejoras que tuvo el módulo Raspberry Pi 3 Modelo B con respecto a su antecesora el módulo Raspberry Pi 2 Modelo B, ya que el nuevo módulo cuenta con una tarjeta inalámbrica incorporada a la placa por lo que se aprovechó esta nueva característica para la comunicación con el Sistema Tierra sin la necesidad de añadir una tarjeta inalámbrica externa ahorrando consumo de energía y costos en el prototipo.

En la Figura 2.6 se puede ver la antena que vienen incorporada en el módulo Raspberry Pi 3.

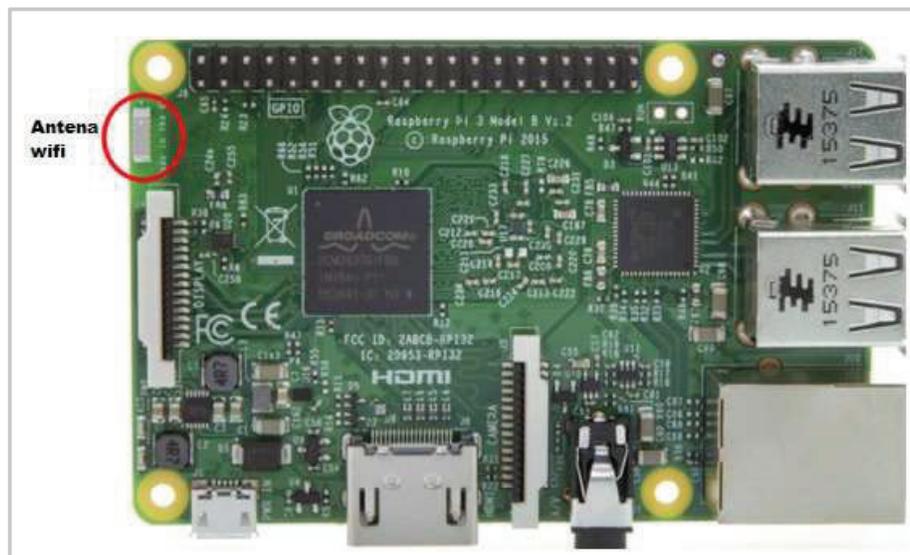


Figura 2.6 LAN Inalámbrica 802.11 b/g/n [17]

### 2.1.4 GPS UBLOX NEO -7M

Para la obtención de la ubicación, cuando se toma una imagen térmica desde el aire, se requiere un módulo GPS para cumplir con este requerimiento se escogió el

GPS Ublox Neo – 7M debido a la compatibilidad con la tarjeta Raspberry Pi 3 además se tomó en cuenta su disponibilidad en el mercado. A continuación, se presenta sus principales características técnicas.

- Modulo NEO-7M
- Comunicación serial
- Voltaje de alimentación: (3.5 – 5) VDC
- Antena cerámica activa incluida
- LED indicador de señal
- Tamaño de antena: 25x25mm
- Tamaño de modulo: 25x35mm
- *BAUDRATE*: 9600
- EEPROM para guardar configuración de parámetros.

En la Figura 2.7 se presenta la fotografía del módulo GPS.



Figura 2.7 GPS Ublox Neo-7M [18]

El módulo GPS se conecta a los pines GPIO de la tarjeta Raspberry Pi 3. A continuación, en la Figura 2.8 se presenta un diagrama conexión.

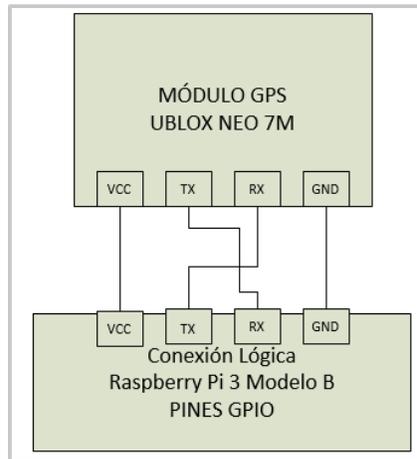


Figura 2.8 Diagrama de Conexión GPS

El módulo de que GPS para la conexión con la tarjeta Raspberry Pi 3 utiliza los pines del interfaz UART, que corresponde a una comunicación bit a bit sin la ayuda de una señal de reloj para la sincronización.

### 2.1.5 CÁMARA TÉRMICA

Para el presente prototipo se revisó las características técnicas entre el módulo Raspberry Pi 3 y la cámara térmica Flir LWIR Lepton y se pudo verificar que estos dispositivos son compatibles. Dicho cámara sirve para capturar imágenes térmicas que responden a radiaciones infrarrojas con longitudes de onda comprendidas entre 8 a 14 micras. En la Figura 2.9 se puede observar la cámara térmica que se utilizó en el presente trabajo de titulación.



Figura 2.9 Cámara térmica Flir Lepton

### 2.1.5.1 Características Técnicas

Dentro de sus principales características se puede mencionar las siguientes:

- Dimensiones: 8.5 x 11.7 x 5.6 mm (sin zócalo); 10.6 x 11.7 x 5.9 mm (incluyendo el zócalo).
- 51 grados de HFOV (*Horizontal Field of View*, Campo de visión horizontal), 63.5° diagonal (f/1.1 doblete de silicio).
- 80 píxeles activos (h) x 60 (v) píxeles activos.
- Sensibilidad térmica < 50 mK.
- Velocidad de fotograma compatible con la exportación (< 9 Hz).
- Utiliza fuentes de alimentación compatibles con teléfonos celulares estándares: 2.8 V al sensor, 1.2 V a la base digital y IO flexible de 2.5 V a 3.1 V.
- Tiempo rápido de imagen (<0.5 seg).
- Potencia de funcionamiento 150 mW.

### 2.1.5.2 Distribución de Pines

La conexión entre la cámara térmica Flir LWIR Lepton y la tarjeta Raspberry Pi 3 se realiza utilizando los pines del GPIO (*General Purpose Input/Output*). En la Figura 2.10 y en la Tabla 2.4 se presenta la distribución de pines del dispositivo.

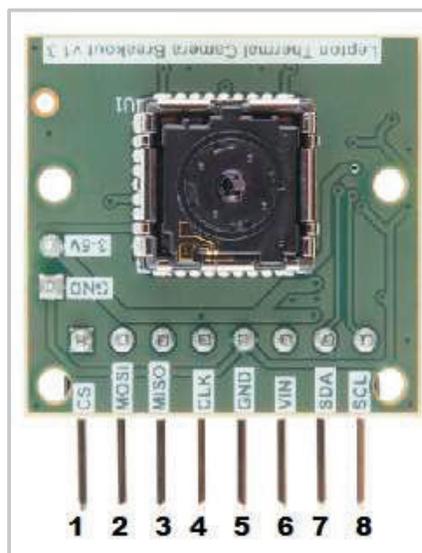


Figura 2.10 Pines cámara térmica

PIN	FUNCIÓN
1	CS (Chip <i>Select</i> – Interfaz SPI)
2	MOSI (Datos – Interfaz SPI)
3	MISO (Datos –Interfaz SPI)
4	CLK (Reloj-Interfaz SPI)
5	GND (Tierra)
6	VIN (Alimentación)
7	SDA (Datos –Interfaz I2C)
8	SCL (Reloj - Interfaz I2C)

Tabla 2.4 Distribución de Pines

### 2.1.6 BATERÍA

La fuente de energía que se utilizará en el prototipo son las comúnmente llamadas *power bank* o baterías externas que generalmente son utilizadas para cargar dispositivos móviles.

Este tipo de baterías portátiles se diferencian una de otra por su capacidad de almacenamiento, la unidad que manejan son los miliamperios por hora (mAh) por lo que entre mayor sea el amperaje por hora de la batería, mayor será la autonomía del sistema.

Las baterías usadas en el prototipo tienen las siguientes características

- 1 puerto de entrada microUSB para cargarla.
- 1 puerto de salida USB.
- Batería de Ion-Litio
- 2600 mAh de capacidad
- Voltaje de salida 5V/1A
- Voltaje de entrada 5V/1A
- Dimensiones: 70 x 30.5 x 21 mm
- Peso: 63.1 gramos

Se escogió este tipo de baterías externas ya que se encuentran con facilidad en el mercado y presentan adecuadas características técnicas. Sus dimensiones fueron

determinantes en la elección, ya que como se ha mencionado el dron tendrá que llevar el Sistema Aéreo y su peso debe ir acorde con la capacidad de los motores que tiene el dron. En la Figura 2.11 se presenta la batería utilizada en el presente trabajo de titulación.



Figura 2.11 Batería externa [19]

### 2.1.7 ARMAZÓN

La batería, tarjeta Raspberry Pi 3 y todos sus periféricos se deben contener en un armazón que permita acoplar a la base del dron de forma segura. El armazón que contiene dichos elementos electrónicos debe estar fabricado con un material liviano como Poli-estireno expandido o plástico ya que el peso muy importante para que no afecte al desplazamiento del dron. A continuación, se presenta la Figura 2.12, donde se puede ver la carcasa de una cámara de video de seguridad antes de ser ensamblada que podría utilizarse en el presente trabajo.



Figura 2.12 Protección de dispositivos electrónicos del Sistema Aéreo

## 2.1.8 SOFTWARE

Los requerimientos de software en el Sistema Aéreo son:

- SO Raspbian
- Software

### 2.1.8.1 Sistema Operativo Raspbian

El primer requerimiento de software en el Sistema Aéreo es el sistema operativo Raspbian ya que la base para que funcione todos los procesos, conexiones, servicios y *scripts* de la tarjeta Raspberry Pi. Se escogió Raspbian porque como se mencionó en el primer capítulo tiene 35.000 paquetes pre-compilados y que han sido optimizados para que funcionen de la mejor manera con la tarjeta Raspberry Pi.

El sistema operativo Raspbian puede ser descargado desde la página web oficial de la tarjeta Raspberry Pi en el siguiente *link* "<https://www.raspberrypi.org/downloads>" En la Figura 2.13 se puede observar el sistema operativo Raspbian, también la aplicación NOOBS (*New Out Of Box Software*) que permite instalar los sistemas operativos como por ejemplo Raspbian, Arch Linux, RaspBMC, Pidora u OpenELEC. [20]

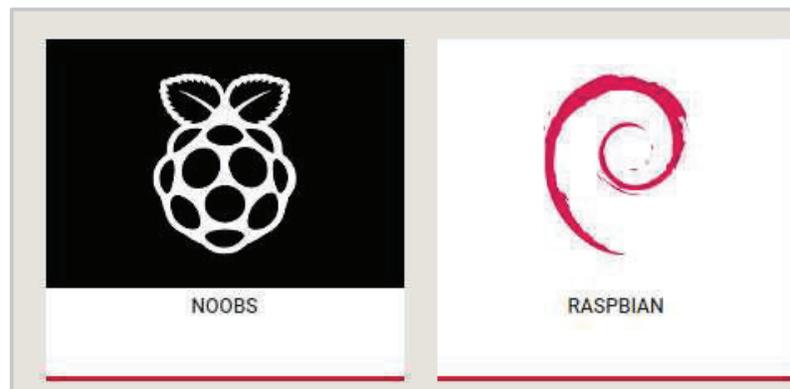


Figura 2.13 Sistema Operativo Raspbian [20]

A parte del sistema operativo Raspbian existen otros sistemas que se pueden instalar en la tarjeta Raspberry Pi. En la Figura 2.14 se puede ver que hay la posibilidad de descargar el sistema operativo PINET, WINDOWS 10 que es utilizado proyectos relacionados con el internet de las cosas, Risc OS, etc.

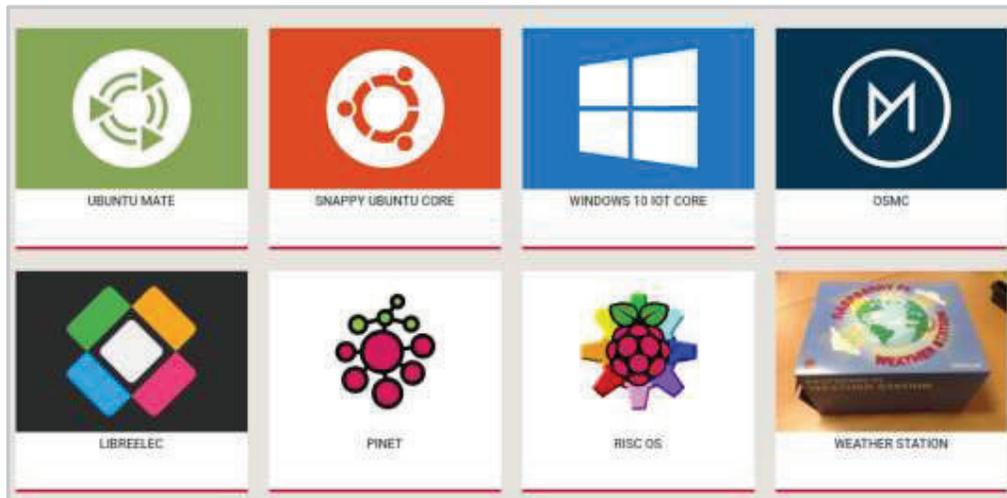


Figura 2.14 Sistemas Operativos para Raspberry Pi [20]

### 2.1.8.2 Software

El segundo requerimiento de software en el Sistema Aéreo son los programas que se encargan de capturar, procesar, fusionar y enviar la información que es proporcionada por el dispositivo GPS y la cámara térmica.

Una vez instalado el sistema operativo Raspbian en la tarjeta Raspberry Pi, se puede realizar la programación en varios lenguajes como por ejemplo JavaScript, Python, C/C++, aunque los más usados son Python y C, ya que vienen instalados con el paquete Raspbian.

### 2.1.8.3 Python

Es un lenguaje de programación sencillo, orientado a objetos y compatible con la tarjeta Raspberry Pi debido a su naturaleza *open source*. Cuando se instala el sistema operativo Raspbian, por defecto se incluye el intérprete del lenguaje de programación Python. A continuación, se indica las principales características.

- Simple
- De propósito general
- *Open source*
- Lenguaje Orientado a Objetos
- Lenguaje de Alto Nivel
- Posee extensas librerías

Se escogió el lenguaje Python debido a que se pudo verificar que existen librerías realizadas en dicho lenguaje que permiten interactuar con la cámara térmica y el módulo del GPS y en comparación con librerías realizadas en el lenguaje C su complejidad es menor. Las librerías *pylepton* y *gps* son la base de los *scripts* del Sistema Aéreo.

## 2.2 SISTEMA TIERRA

Para poder poner en funcionamiento el Sistema Tierra se requieren los elementos de software y de hardware. En la Figura 2.15 se presenta las partes que conforman el sistema en tierra.



Figura 2.15 Partes del Sistema Tierra

### 2.2.1 COMPUTADOR PORTÁTIL

En el Sistema Tierra se utiliza un computador portátil para poder visualizar la información del Sistema Aéreo mediante la ejecución de un programa y una base de datos. Las características que debe tener el equipo portátil para ejecutar el programa receptor como también la base de datos se presenta en la Tabla 2.5.

En la Figura 2.16 se presenta el equipo que se podría utilizar como receptor en tierra.

Especificaciones Técnicas	
Procesador	Intel® Core™ i3-3217U 1.8 GHz
Memoria RAM	8GB RAM
Disco Duro	500 GB 5400rpm
Conectividad	Wi-Fi 802.11b/g/n
Sistema Operativo	Windows 7

Tabla 2.5 Especificaciones Técnicas Laptop [21]



Figura 2.16 Laptop DELL 3421 [21]

### 2.2.2 SOFTWARE

El Sistema Tierra recibe la información enviada por el Sistema Aéreo. Por lo que se requiere la ejecución de un programa en tierra que reciba y presente el *streaming* de video térmico, ubicación geográfica, fecha, hora y altitud. El programa se desarrollará en Visual Studio 2013 en el lenguaje de programación orientado a objetos C#.

### 2.2.3 BASE DE DATOS

Para el almacenamiento de las imágenes con su ubicación geográfica se requiere una base de datos por lo que se utilizará el sistema de manejo de bases de datos Microsoft SQL Server 2014 que es desarrollado por la empresa Microsoft. La base de datos constará de una sola tabla que servirá para el registro de las imágenes térmicas con la información de ubicación y tiempo.

## 2.3 DISEÑO PROTOTIPO

Con los requerimientos detallados anteriormente se presenta el diseño del prototipo donde se presenta el diagrama de bloques del Sistema Aéreo y del Sistema Tierra, diagrama de conexión del prototipo, diagrama de casos de uso y diagrama de la base de datos.

El diseño del prototipo se realizará en dos etapas, un diseño del Sistema Aéreo y un diseño Sistema Tierra ya que estos dos subsistemas están conformados por diferentes elementos y cumplen diferentes funciones dentro del prototipo.

### 2.3.1 DISEÑO SISTEMA AÉREO

#### 2.3.1.1 Diagrama de bloques

Con los requerimientos anteriormente detallados se presenta un diagrama de bloques tanto de software y de hardware, que sirven como base para entender el funcionamiento del Sistema Aéreo y también se utiliza como una hoja de ruta para desarrollar esta etapa. En la Figura 2.17 se presenta un diagrama de bloques correspondiente a la parte de hardware del Sistema Aéreo, como se puede observar se tiene una base principal que es vehículo no tripulado o Dron, en un segundo nivel se tiene la tarjeta Raspberry Pi 3 y la tarjeta wifi, como se detalló en los requerimientos dicha tarjeta de red viene acoplada a la placa Raspberry Pi 3. En un tercer nivel se tiene el módulo GPS y la cámara térmica que son los que alimentan al sistema tanto con datos de ubicación como con video térmico.

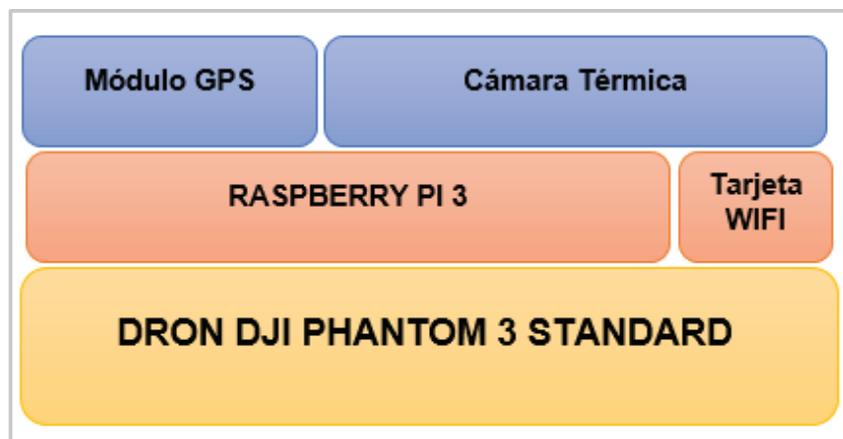


Figura 2.17 Diagrama de bloques del hardware aéreo

De la misma manera se realizó el diagrama de bloques para la parte de software del Sistema Aéreo, con esto se tienen los pasos a seguir en esta etapa. En la Figura 2.18 se presenta como primera etapa la instalación del sistema operativo Raspbian, una vez completado este nivel, se utilizará el intérprete de programación de Python para la construcción de los *scripts*, como se mencionó anteriormente, una vez que se instala el sistema operativo Raspbian por defecto viene instalado el intérprete de Python.

Finalmente se tiene la tercera etapa que corresponde a los *scripts*, el *script* “Programa\_GPS” recoge la información obtenida por el GPS. El *script* “Programa\_Cámara” captura imágenes de forma continua y de esta forma se crea un *streaming* de video térmico.

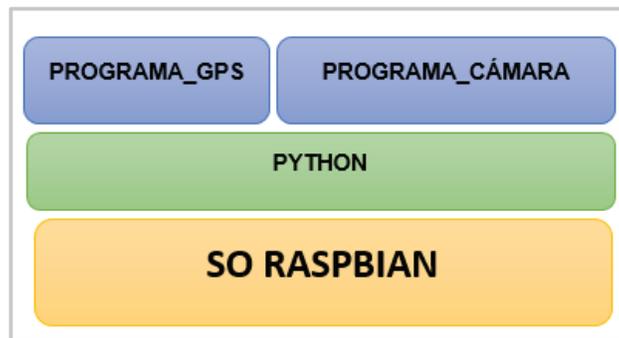


Figura 2.18 Diagrama de bloques software aéreo

### 2.3.1.2 Método de Transmisión

El proceso de transmisión se presenta entre dos hosts en el que uno de ellos requiere obtener la información que su par posee por lo que se podría usar protocolos de comunicación TCP y UDP para lograr el objetivo deseado.

Para el presente prototipo se evaluó cuál de los dos protocolos se debe utilizar, por lo que a continuación se detalla como primer paso las características principales de cada uno.

#### 2.3.1.2.1 Protocolo TCP (*Transmission Control Protocol, Protocolo de Control de Transmisión*)

- Para el envío de datos el protocolo establece, envía y cierra la comunicación punto a punto.

- El protocolo TCP es un protocolo orientado a conexión.
- El protocolo TCP garantiza que la información llegue en orden a su destino.
- Por cada paquete enviado se tiene un paquete de confirmación.

#### 2.3.1.2.2 *Protocolo UDP (User Datagram Protocol, Protocolo de datagrama de usuario)*

- Es un protocolo no orientado a conexión.
- No garantiza la entrega de la información a su extremo.
- No se garantiza la integridad de los datos.
- La información no llega en un orden determinado.

Por lo revisado se puede deducir que el protocolo que se ajusta a la comunicación entre el Sistema Aéreo y Sistema Tierra es el protocolo TCP ya que la información que se transmite debe ser íntegra y ordenada ya que en un solo paquete se envía la información de ubicación e imagen térmica.

Al ser una comunicación entre dos hosts el modelo a utilizarse es el cliente-servidor donde el Sistema Aéreo es el servidor y el Sistema Tierra es el cliente.

#### 2.3.1.3 **Diagrama de conexión**

En la Figura 2.19 se presenta el diagrama de conexión física de los elementos de hardware del Sistema Aéreo. Para la conexión entre el Raspberry Pi 3 y la cámara térmica se utilizaron los pines 1 (3.3 V), 3 (SDA), 5 (SCL), 19 (MOSI), 21 (MISO), 23 (CLK), 26 (CS) del interfaz GPIO de la tarjeta Pi para mejor visualización se representa el cableado con líneas de color azul. En la conexión del módulo GPS se representa el cableado con líneas naranjas y se utilizaron los pines 8 (TX), 9 (GND), 10 (RX), 17 (3.3 V), los pines 8 y 10 que corresponden al interfaz UART se conectan de forma cruzada con los pines del GPS es decir los pines 8 y 9 de la tarjeta Pi se conectan a los pines RX y TX respectivamente. No se presenta la conexión de la tarjeta inalámbrica porque se encuentra integrada con la tarjeta Pi.

Finalmente, la conexión de la batería va al puerto mini USB que tiene la tarjeta Raspberry Pi la cual entrega un voltaje de 5V / 1 A.

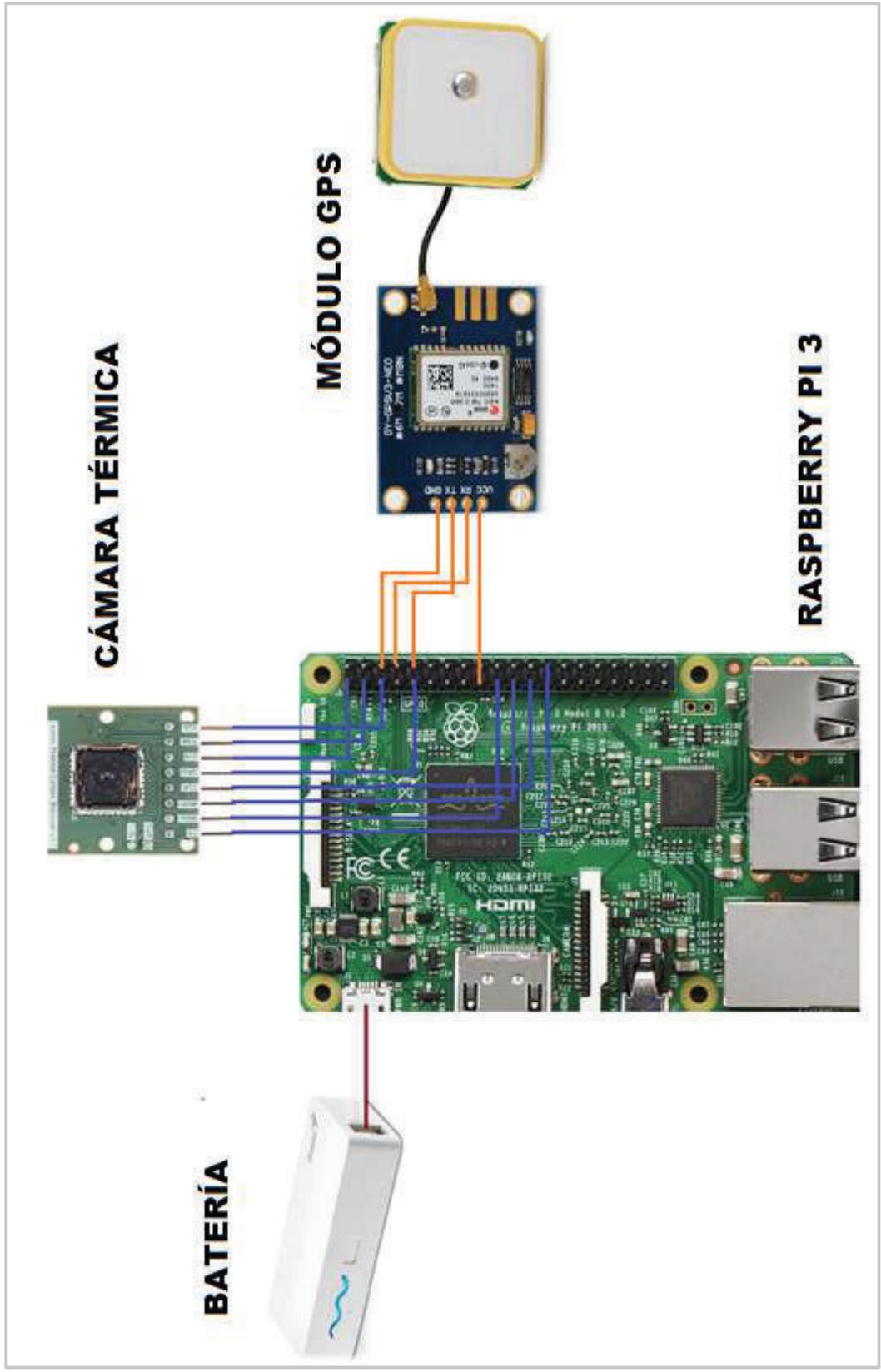


Figura 2.19 Diagrama de conexión

### 2.3.1.4 Diagrama de casos de uso

El Sistema Aéreo está constituido por software el cual está realizado en el lenguaje de programación Python para lo cual se realizó un diagrama de casos de uso que se presenta en la Figura 2.20.

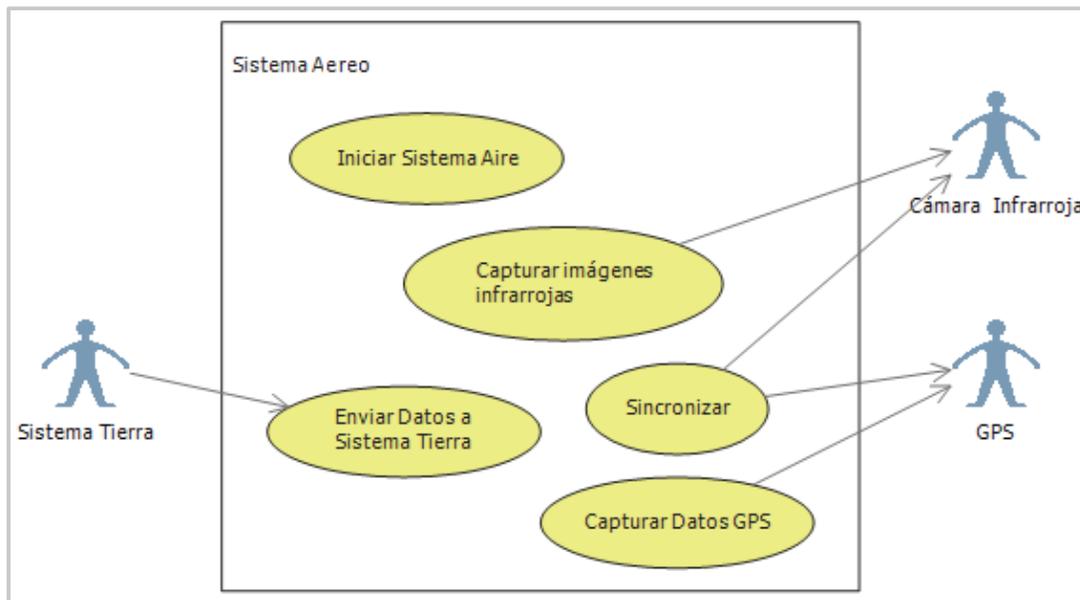


Figura 2.20 Diagrama casos de uso Sistema Aéreo

A continuación, se presenta las tablas correspondientes a cada caso de uso para el Sistema Aéreo.

En la Tabla 2.6 se detalla el caso de uso que permite iniciar el Sistema Aéreo.

<b>Nombre</b>	Iniciar Sistema Aéreo.
<b>Objetivo</b>	Iniciar los servicios de la tarjeta Raspberry Pi.
<b>Actores</b>	Usuario
<b>Precondición</b>	Tarjeta Raspberry Pi está apagada.
<b>Secuencia</b>	<ol style="list-style-type: none"> <li>1. Se enciende la Tarjeta Raspberry Pi.</li> <li>2. Se conecta a la red ad hoc.</li> <li>3. Se inicia el servicio "Servicio_Camara.service".</li> <li>4. Se inicia el servicio "Servicio_GPS.service".</li> </ol>
<b>Postcondición</b>	El sistema se encuentra iniciado, esperando conexión con el Sistema Tierra.
<b>Excepciones</b>	-

Tabla 2.6 Caso de uso: Iniciar Sistema Aéreo

En la Tabla 2.7 se detalla el caso de uso para capturar las imágenes térmicas.

<b>Nombre</b>	Capturar imágenes térmicas
<b>Objetivo</b>	Capturar continuamente imágenes térmicas.
<b>Actores</b>	Cámara térmica
<b>Precondición</b>	Servicio cámara iniciado.
<b>Secuencia</b>	<ol style="list-style-type: none"> <li>1. Acceder a la cámara térmica.</li> <li>2. Tomar imagen térmica.</li> <li>3. Pintar la imagen con una paleta de colores para visualización.</li> <li>4. Guardar en un archivo con extensión JPEG.</li> </ol>
<b>Postcondición</b>	La imagen se encuentra disponible hasta la próxima captura.
<b>Excepciones</b>	-

**Tabla 2.7 Caso de uso: Capturar Imágenes Térmicas**

En la Tabla 2.8 se detalla el caso de uso para capturar la información del GPS.

<b>Nombre</b>	Capturar Datos GPS
<b>Objetivo</b>	Capturar información de ubicación.
<b>Actores</b>	Módulo GP
<b>Precondición</b>	Servicio GPS iniciado
<b>Secuencia</b>	<ol style="list-style-type: none"> <li>1. Acceder al módulo GPS.</li> <li>2. Capturar datos de longitud, latitud, altura y tiempo</li> <li>3. Transformar los datos de tiempo a la zona Horaria correspondiente.</li> <li>4. Guardar los datos en un archivo con extensión txt.</li> </ol>
<b>Postcondición</b>	Los datos GPS se encuentran disponibles.
<b>Excepciones</b>	-

**Tabla 2.8 Caso de uso: Capturar Datos GPS**

En la Tabla 2.9 se detalla el caso de uso para sincronizar la información

<b>Nombre</b>	Sincronizar
<b>Objetivo</b>	Fusionar datos de ubicación e imágenes térmicas
<b>Actores</b>	Módulo GPS, Cámara térmica
<b>Precondición</b>	Servicio Cámara iniciado Servicio GPS iniciado
<b>Secuencia</b>	<ol style="list-style-type: none"> <li>1. Capturar imagen térmica</li> <li>2. Capturar ubicación GPS</li> <li>3. Crear paquete de información</li> </ol>
<b>Postcondición</b>	El paquete de información está listo para su transmisión
<b>Excepciones</b>	-

**Tabla 2.9 Caso de uso: Sincronizar**

En Tabla 2.10 se detalla el caso de uso para enviar los datos al Sistema Tierra.

<b>Nombre</b>	Enviar Datos al Sistema Tierra
<b>Objetivo</b>	Enviar el paquete de información imagen e ubicación.
<b>Actores</b>	Sistema Tierra
<b>Precondición</b>	Información sincronizada
<b>Secuencia</b>	<ol style="list-style-type: none"> <li>1. Recibir petición del Sistema Tierra</li> <li>2. Enviar datos al Sistema Tierra.</li> <li>3. Esperar un tiempo y reenvío continuo</li> </ol>
<b>Postcondición</b>	-
<b>Excepciones</b>	-

Tabla 2.10 Caso de uso: Enviar datos a Sistema Tierra

### 2.3.1.5 Diseño del Armazón

Para poder integrar los dispositivos del Sistema Aéreo se utilizó un armazón de una cámara de vigilancia, se retiró todos los elementos de la cámara de vigilancia y se dejó el espacio libre para introducir los elementos electrónicos del presente prototipo.

En la Figura 2.21 se presenta el modo de fijación entre el dron y el armazón que contiene los elementos electrónicos del Sistema Aéreo. El armazón tiene que estar en el centro de gravedad para que el dron pueda elevarse y moverse sin problemas.



Figura 2.21 Armazón Sistema Aéreo

### 2.3.2 DISEÑO SISTEMA TIERRA

En la Figura 2.22 se presenta el diagrama de bloques del Sistema Tierra. Para este sistema no se dividió en software y hardware ya que el computador portátil es un elemento que ya está implementado y es utilizado para ejecutar el software a realizarse. De la misma manera que la tarjeta Raspberry Pi 3, la antena wifi está integrada en el computador portátil es por ello que no es necesario otro elemento inalámbrico.

El diagrama de bloques sirve como guía para llegar al objetivo que es la implementación del Sistema Tierra. Como se puede ver en la Figura 2.22, se tiene como primer nivel el hardware que corresponde al computador portátil, en esta fase se debe cumplir con los requerimientos señalados en la sección 2.5.1. En el segundo nivel se utiliza el sistema operativo Windows 7. En el tercer nivel se requiere la instalación del programa Visual Studio 2013 y del sistema de manejo de bases de datos SQL Server.

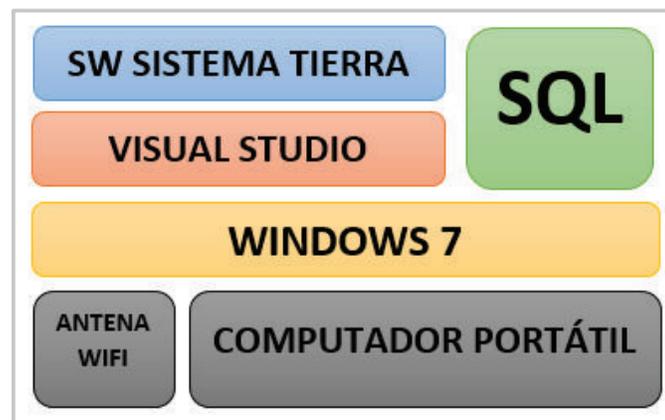


Figura 2.22 Diagrama general Sistema Tierra

Finalmente, completado estos niveles se crea el software que recibe la información, se presentará en un interfaz amigable al usuario y permitirá almacenar los datos de interés que en este caso es la ubicación del Sistema Aéreo y la imagen térmica.

#### 2.3.2.1 Diagrama de casos de uso

El Sistema Tierra está constituido por software realizado en el lenguaje de programación C# para lo cual se bosquejó un diagrama de casos de uso que se presenta en la Figura 2.23.

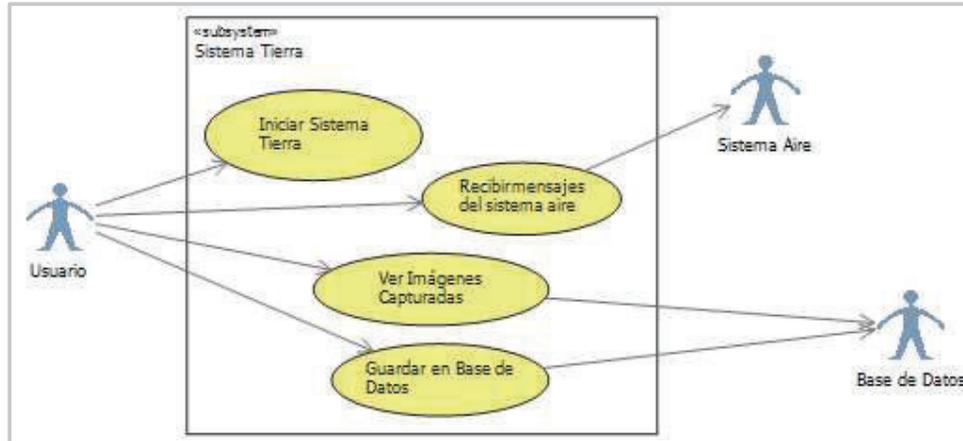


Figura 2.23 Diagrama casos de uso Sistema Tierra

A continuación, se presenta las tablas correspondientes a cada caso de uso para el Sistema Tierra.

En la Tabla 2.11 se detalla el caso de uso para iniciar el Sistema Tierra.

<b>Nombre</b>	Iniciar Sistema Tierra
<b>Objetivo</b>	Mostrar interfaz principal
<b>Actores</b>	Usuario
<b>Precondición</b>	El Sistema Tierra está apagado.
<b>Secuencia</b>	1. Se une a la red ad-hoc. 2. Se crea el formulario principal de visualización de información.
<b>Postcondición</b>	El sistema se encuentra listo para pedir mensajes del Sistema Aéreo.
<b>Excepciones</b>	-

Tabla 2.11 Caso de uso: Iniciar Sistema Tierra

En la Tabla 2.12 se detalla el caso de uso para recibir los mensajes del Sistema Aéreo.

<b>Nombre</b>	Recibir mensajes del Sistema Aéreo
<b>Objetivo</b>	Visualizar el video térmico y la información de ubicación.
<b>Actores</b>	Usuario y Sistema Aire
<b>Precondición</b>	Sistema debe estar iniciado
<b>Secuencia</b>	1. Se envía un mensaje de petición al Sistema Aire. 2. Se reciben los datos. 3. Se visualiza la información en el interfaz principal.
<b>Postcondición</b>	El sistema se mantiene recibiendo mensajes del Sistema Aéreo
<b>Excepciones</b>	-

Tabla 2.12 Caso de uso: Recibir mensajes del Sistema Aéreo

En la Tabla 2.13 se detalla el caso de uso para ver las imágenes capturadas.

<b>Nombre</b>	Ver imágenes capturadas
<b>Objetivo</b>	Ver imágenes capturadas con su información de ubicación.
<b>Actores</b>	Usuario, Base de datos.
<b>Precondición</b>	Recibir mensajes desde el Sistema Aéreo.
<b>Secuencia</b>	1.Presentar formulario de imágenes guardadas. 2.Consultar la base de datos. 3.Presentar imagen con información de ubicación.
<b>Postcondición</b>	-
<b>Excepciones</b>	-

Tabla 2.13 Caso de uso: Ver imágenes capturadas

En la Tabla 2.14 se detalla el caso de uso para guardar los datos en la base de datos.

<b>Nombre</b>	Guardar en base de datos
<b>Objetivo</b>	Guardar la imagen y ubicación deseadas.
<b>Actores</b>	Usuario, base de datos
<b>Precondición</b>	Recibir mensajes desde el Sistema Aéreo.
<b>Secuencia</b>	1. Capturar imagen y ubicación deseada 2. Guardar en la base de datos el mensaje
<b>Postcondición</b>	-
<b>Excepciones</b>	-

Tabla 2.14 Caso de uso: Guardar en base de datos

### 2.3.2.2 Diagrama de base de datos

En la Figura 2.24 se presenta el diagrama de la base de datos. La base de datos está conformada por una sola tabla que permite almacenar la información que el usuario captura.

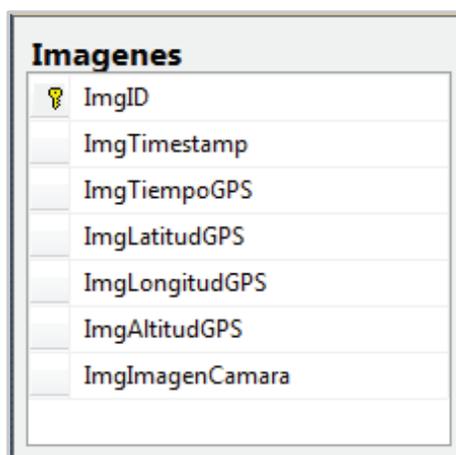


Figura 2.24 Diagrama de base de datos

### 2.3.2.3 Diseño del Interfaz

En el Sistema Tierra los datos serán recibidos y presentados en una interfaz principal donde se tiene el *streaming* de video térmico, la información de la ubicación, fecha, hora y la altitud. El interfaz principal donde se muestre los datos y el *streaming* de video térmico debe ser simple y fácil de usar ya que el prototipo pretende ser utilizado en situaciones de emergencia por lo que la simplicidad para la utilización del mismo es muy importante. En la Figura 2.25 se presenta el *sketch* del interfaz principal.

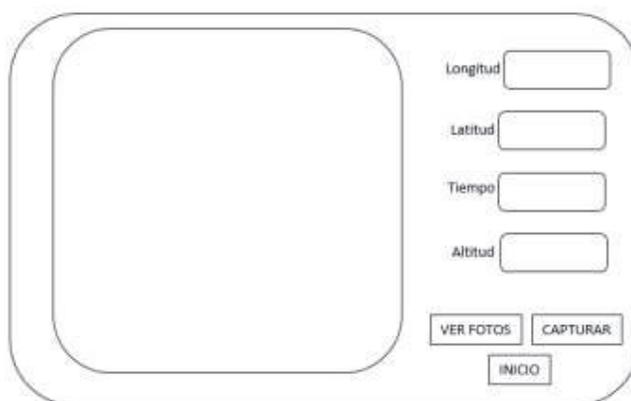


Figura 2.25 Sketch interfaz principal

Las imágenes capturadas se podrán guardar y también se deberá permitir la visualización de las mismas, en la Figura 2.26 se muestra el *sketch* del interfaz de presentación de las imágenes térmicas capturadas. Se presentan todas las imágenes capturadas en una grilla con la información correspondiente.

LONGITUD	LATITUD	TIEMPO	ALTITUD	FOTO
-78,43333333	-0,00245646	01/01/2017 15:30	2600	

Figura 2.26 Sketch interfaz de almacenamiento

Finalmente, en la Figura 2.27 se presenta un *sketch* del interfaz que permite visualizar un ítem específico que se presenta dando clic en registro de interés de la grilla.

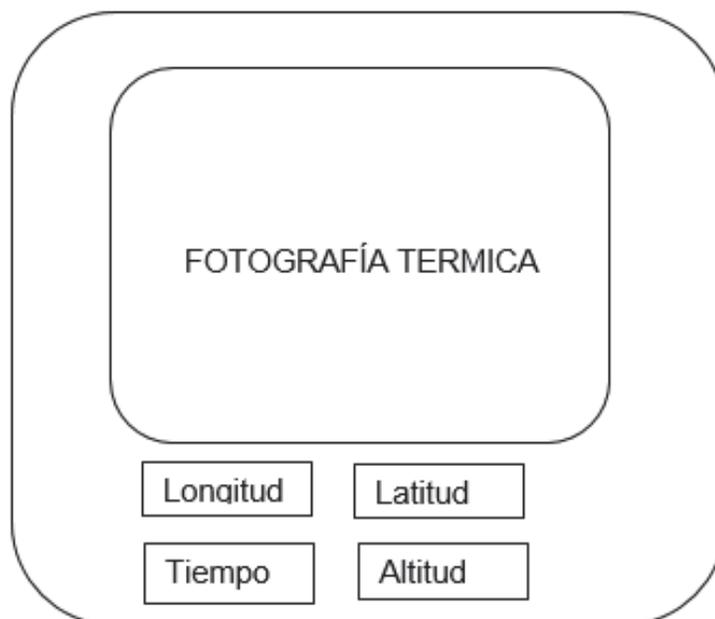


Figura 2.27 Sketch interfaz ítem específico

## 2.4 DIRECCIONAMIENTO IP

La comunicación inalámbrica entre los sistemas se establece a través de red ad hoc. La red utilizada en el prototipo es la 10.10.10.0 y se requiere 2 host como número máximo de *host*.

En la Tabla 2.15 se presenta el direccionamiento IP que se utilizó en el presente prototipo para establecer la comunicación.

Red	Máscara	Rango de Direcciones IP Válidas
10.10.10.0	255.255.255.252	10.10.10.1 -10.10.10.2

Tabla 2.15 Direccionamiento IP

Con la información detallada se asignó de forma estática la dirección 10.10.10.1/30 a la tarjeta Raspberry Pi y la dirección 10.10.10.2/30 al computador portátil del Sistema Tierra.

## CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS DEL PROTOTIPO

En base a los requerimientos y al diseño realizado mediante diagramas en el capítulo 2 se presenta la implementación del prototipo el cual incluye la parte de hardware y de software de los sistemas. Adicionalmente se presenta las pruebas realizadas del módulo GPS, de la cámara térmica, de la comunicación inalámbrica y de los módulos de software. Finalmente se presentarán fotografías térmicas tomadas en diferentes áreas de la cancha de la Escuela Politécnica Nacional y en un lugar donde haya la presencia de neblina.

### 3.1 CONFIGURACIÓN DE HARDWARE

#### 3.1.1 CONFIGURACIÓN DE RASPBERRY PI 3

Para la configuración de la tarjeta Raspberry Pi es necesario el uso de una memoria MicroSD donde se guardará la información del sistema operativo que se desee instalar, la memoria utilizada realiza la función de un disco duro en un computador.

La capacidad de la memoria MicroSD depende de qué sistema operativo se va a instalar, de acuerdo con la documentación oficial, se menciona que el sistema operativo Raspbian requiere 4Gb por lo que la capacidad mínima que debe tener la memoria es de 8 Gb para instalar el sistema operativo y tener espacio para la ejecución de otras aplicaciones. En el presente trabajo se utilizó una memoria con una capacidad de 32 Gb para no tener inconvenientes por falta de memoria, con esto se asegura que la tarjeta Raspberry Pi trabaje correctamente.

Se borró toda la información que haya estado en la memoria MicroSD, para esto se procedió a descargar la aplicación *SD Card Formatter 4.0 for SD/SDHC/SDXC* para el sistema operativo Windows.

##### 3.1.1.1 Instalación Sistema Operativo Raspbian

Para la instalación del sistema operativo se escogió la aplicación NOOBS (*New Out of Box Software*) la cual permite que se pueda instalar de forma sencilla el SO Raspbian.

Se descarga la aplicación desde la página oficial como se puede ver en la Figura 2.28.



Figura 2.28 Aplicación NOOBS [20]

Una vez descargado el paquete, se descomprime y se tiene los archivos que se presentan en la Figura 2.29.

defaults	27/09/2016 21:39	Carpeta de archivos	
os	27/09/2016 21:39	Carpeta de archivos	
bcm2708-rpi-b.dtb	18/03/2016 8:13	Archivo DTB	11 KB
bcm2708-rpi-b-plus.dtb	18/03/2016 8:13	Archivo DTB	11 KB
bcm2709-rpi-2-b.dtb	18/03/2016 8:13	Archivo DTB	12 KB
bcm2710-rpi-3-b.dtb	18/03/2016 8:13	Archivo DTB	13 KB
bootcode.bin	18/03/2016 8:18	BIN File	18 KB
BUILD-DATA	18/03/2016 8:18	Archivo	1 KB
INSTRUCTIONS-README.txt	18/03/2016 8:16	Documento de tex...	3 KB
recovery.cmdline	18/03/2016 8:18	Archivo CMDLINE	1 KB
recovery.elf	18/03/2016 8:18	Archivo ELF	600 KB
recovery.img	18/03/2016 8:18	Archivo de image...	2.419 KB
recovery.rfs	18/03/2016 8:18	Archivo RFS	22.840 KB
RECOVERY_FILES_DO_NOT_EDIT	18/03/2016 8:18	Archivo	0 KB
recovery7.img	18/03/2016 8:17	Archivo de image...	2.471 KB
riscos-boot.bin	18/03/2016 8:16	BIN File	10 KB

Figura 2.29 Archivos de la aplicación NOOBS

Los archivos presentados en la Figura 2.29 se deben copiar en la memoria MicroSD, una vez realizado este procedimiento se coloca la memoria en la tarjeta Raspberry Pi 3 y se enciende el equipo. En la Figura 2.30 se puede observar los *log's* previos a la instalación del sistema operativo.

```

1.896625] (<000b37e4>) (kernel_init) from [<0000fbc0>] (ret_from_fork+0x14-0x2c)
1.896763] mnc1: queuing unknown
1.896801] CPU1: stopping
1.896811] CPU: 1 PID: 0 Conn: swapper/0 Not tainted 4.4.11-07+ #868
1.897061] Hardware name: BCM2709
1.897061] (<00010724>) (dmesg_backtrace) from [<00014050>] (show_stack+0x20-0x24)
1.897061] (<00014050>) (show_stack) from [<00032054>] (dump_stack+0x4-0x11b)
1.897061] (<00032054>) (dump_stack) from [<00094e4>] (handle_IPI+0x24-0x24)
1.897201] (<00094e4>) (handle_IPI) from [<00094e4>] (hcn2836_arm_irqchip_handle_irq+0x80-0x8c)
1.897476] (<00094e4>) (hcn2836_arm_irqchip_handle_irq) from [<005b0e04>] (__irq_svc+0x44-0x5c)
1.897581] (<00094e4>) (hcn2836_arm_irqchip_handle_irq) from [<005b0e04>] (__irq_svc+0x44-0x5c)
1.897763] Exception stack(0xb933bf98 to 0xb933bf98)
1.897821] ff40: 00000000 b933bf98 00000000 00000000 b933a000 00000000 00000000 00000000
1.897971] ff40: 10c-0374 00000000 00010b30 60000013 ffffffff
1.898129] ff80: 00010b30 00010b30 00010b30 (<00010b30>) (arch_cpu_idle+0x34-0x4c)
1.898221] (<00010b30>) (__irq_svc) from [<00014050>] (arch_cpu_idle+0x34-0x4c)
1.898251] (<00010b30>) (arch_cpu_idle) from [<00063da0>] (default_idle_call+0x34-0x48)
1.898501] (<00063da0>) (default_idle_call) from [<00063f00>] (cpu_startup_entry+0x214-0x2b0)
1.898501] (<00063f00>) (cpu_startup_entry) from [<00015c80>] (secondary_start_kernel+0x12c-0x138)
1.898547] (<00063f00>) (cpu_startup_entry) from [<000095ac>] (<0x95ac>)
1.898771] (<00015c80>) (secondary_start_kernel) from [<000095ac>] (<0x95ac>)
1.898911] CPU0: stopping
1.899274] CPU: 0 PID: 0 Conn: swapper/0 Not tainted 4.4.11-07+ #868
1.899274] Hardware name: BCM2709
1.899319] (<00010724>) (dmesg_backtrace) from [<00014050>] (show_stack+0x20-0x24)
1.899331] (<00014050>) (show_stack) from [<00032054>] (dump_stack+0x4-0x11b)
1.899411] (<00032054>) (dump_stack) from [<00016100>] (handle_IPI+0x24-0x24)
1.899451] (<00016100>) (handle_IPI) from [<000094e4>] (hcn2836_arm_irqchip_handle_irq+0x80-0x8c)
1.899581] (<000094e4>) (hcn2836_arm_irqchip_handle_irq) from [<005b0e04>] (__irq_svc+0x44-0x5c)
1.899581] Exception stack(0x0005ff00 to 0x0005ff00)
1.899621] ff80: 00000000 b97623c8 00000000 00000000 00000000 00000000 00000000 00000000
1.899641] ff20: 00060500 00060500 005b0e04 000c4758 0005ff64 000614f8 0005ff58
1.899641] ff40: 00010b30 00010b30 60000013 ffffffff
1.900291] (<00010b30>) (__irq_svc) from [<00010b30>] (arch_cpu_idle+0x34-0x4c)
1.900441] (<00010b30>) (arch_cpu_idle) from [<00063da0>] (default_idle_call+0x34-0x48)
1.900541] (<00063da0>) (default_idle_call) from [<00063f00>] (cpu_startup_entry+0x214-0x2b0)
1.900701] (<00063f00>) (cpu_startup_entry) from [<00053704>] (rest_init+0x80-0x80)
1.900841] (<00053704>) (rest_init) from [<007e7d34>] (start_kernel+0x3dc-0x3e0)
1.900961] (<007e7d34>) (start_kernel) from [<00000070>] (<0x0070>)

```

Figura 2.30 Log's de encendido de la tarjeta Raspberry Pi

Una vez que termina el encendido, se presenta una ventana como se observa en la Figura 2.31, donde se escoge el sistema operativo Raspbian y se procede a la instalación.

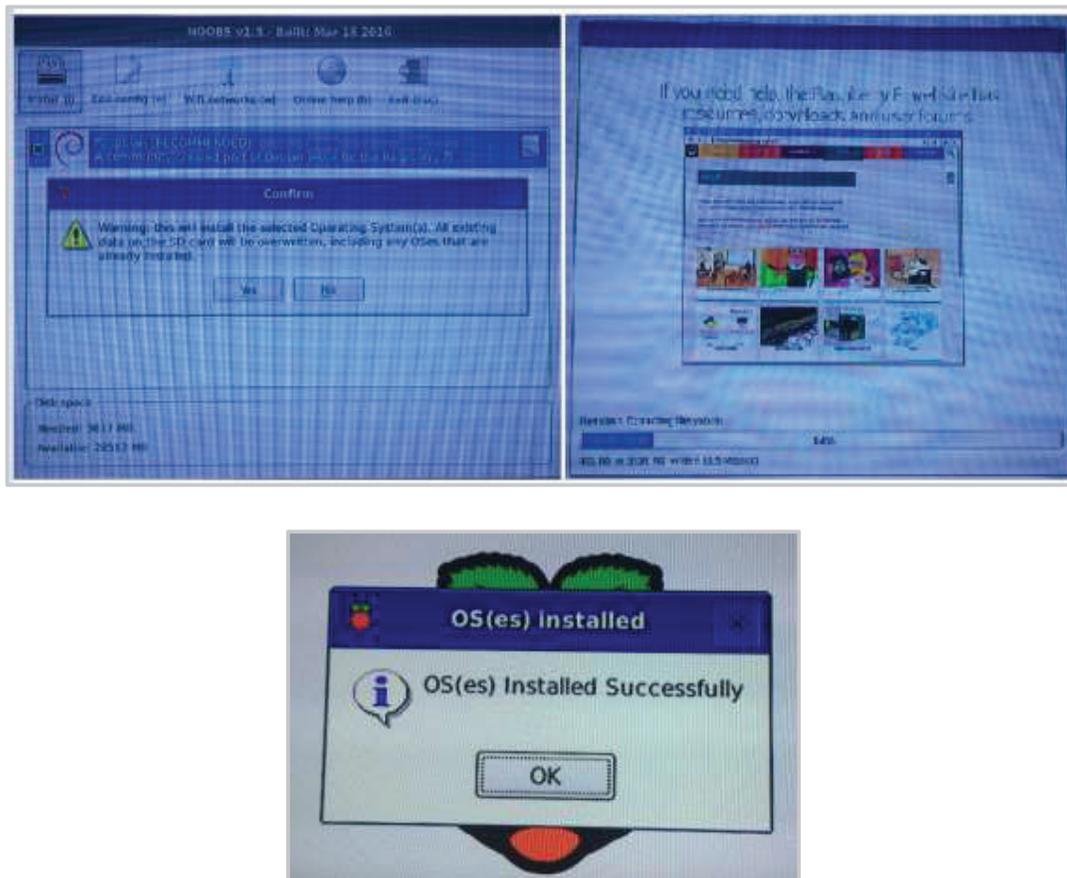


Figura 2.31 Instalación sistema operativo Raspbian

Las credenciales que se encuentran configuradas para el acceso remoto a la tarjeta Raspberry son las que vienen por defecto, el usuario es “pi” se puede ver en la Figura 2.32 y la contraseña es “raspberrypi”.

Una vez configurado estos parámetros se puede ingresar a la tarjeta Raspberry Pi 3 mediante una conexión remota ssh y así se podrá realizar la configuración del resto de periféricos.

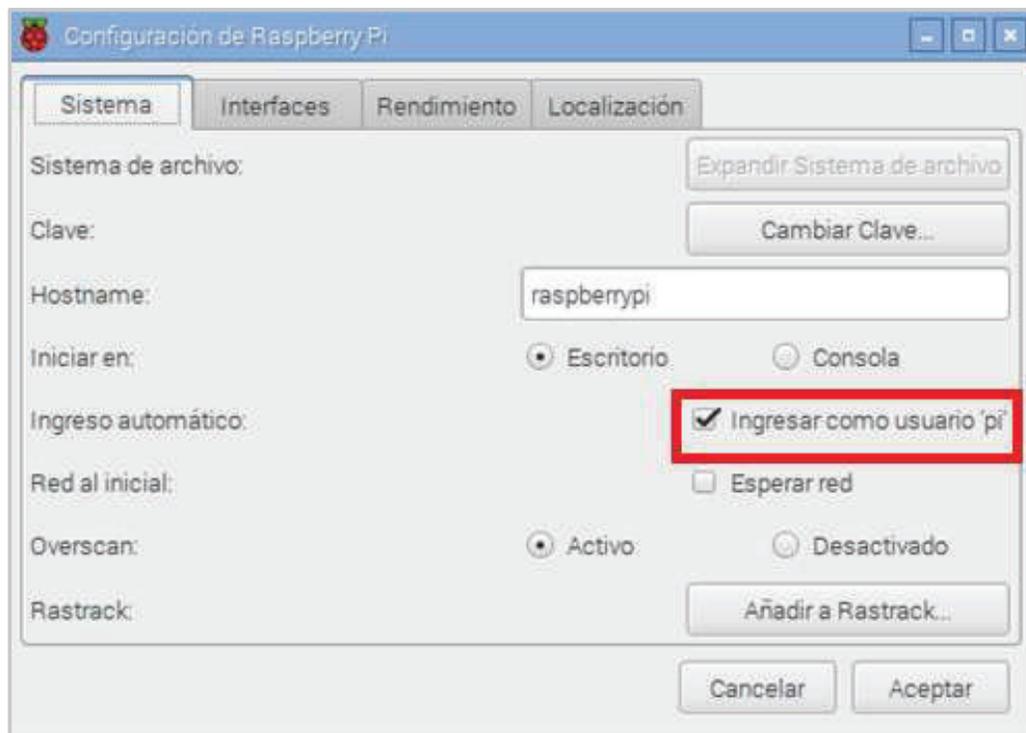


Figura 2.32 Usuario tarjeta Pi

### 3.1.2 CONFIGURACIÓN CÁMARA TÉRMICA

Posterior a la configuración inicial de la tarjeta Raspberry se procedió a realizar la configuración de la cámara térmica Flir Lepton. El primer paso para su configuración es la conexión física con la tarjeta Raspberry Pi, a continuación, se describe el procedimiento.

#### 3.1.2.1 Conexión Cámara Térmica

La cámara térmica posee 8 pines los cuales deben ir conectados al interfaz GPIO de la tarjeta Raspberry Pi. En la Figura 2.33 se puede ver la conexión física de estos dos dispositivos. Para la conexión se utilizó cables llamados *jumpers*.

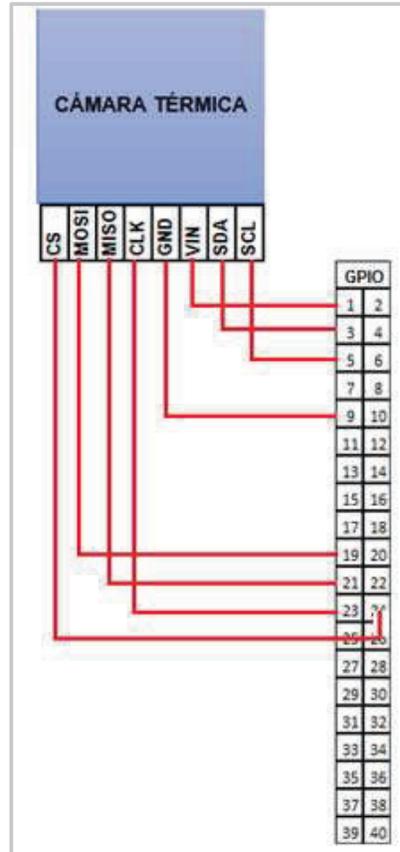


Figura 2.33 Conexión cámara térmica

En la Figura 2.34 se puede observar una fotografía de la conexión de la cámara térmica con la tarjeta Raspberry Pi 3.

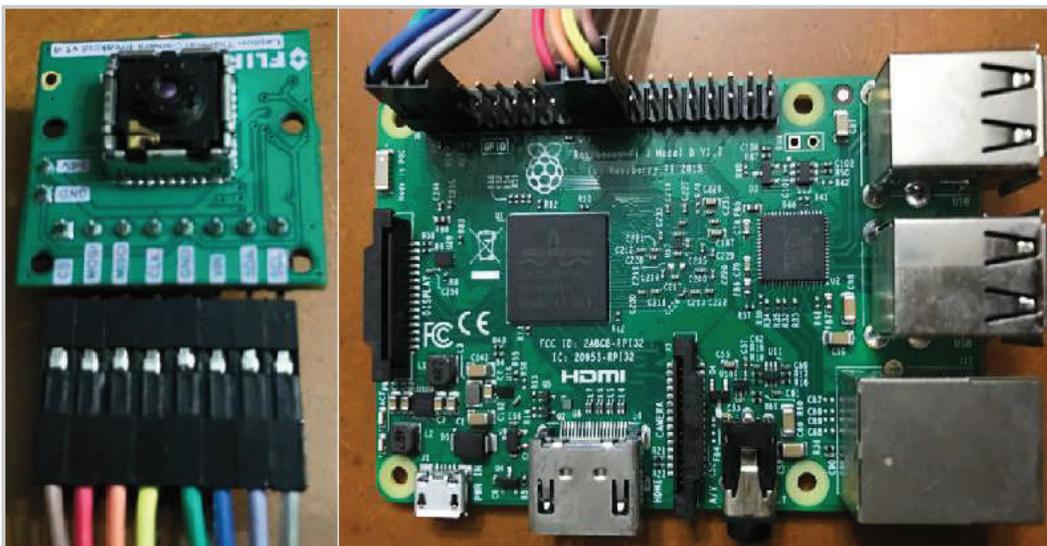


Figura 2.34 Conexión cámara térmica real

### 3.1.2.2 Configuración de Interfaces SPI – I2C

Luego de la conexión física es necesario habilitar las interfaces SPI e I2C de la tarjeta Raspberry Pi 3, son con las que trabajan la cámara térmica.

Para la activación de las interfaces se utiliza la herramienta de configuración de la tarjeta Raspberry Pi 3 que se llama “raspi-config”. Este aplicativo se accede desde el terminal del sistema operativo Raspbian. El comando que se utiliza para ejecutar la utilidad es “sudo raspi-config”. En la Figura 2.35 se presenta el resultado de la ejecución de dicho comando. Como se puede observar se despliega un menú con varias opciones que permite la personalización de la tarjeta Raspberry Pi 3.

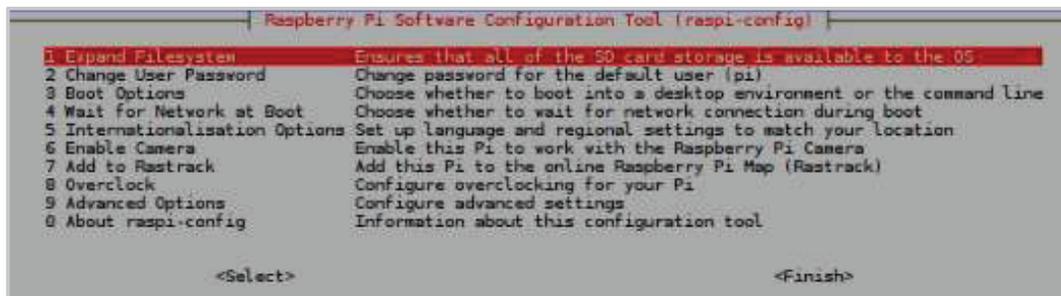


Figura 2.35 Utilidad raspi-config

Se escoge la opción 9 “Advanced Options”, a continuación, se presenta otra ventana como se puede ver en la Figura 2.36. Finalmente, se habilita las interfaces SPI e I2C que son las que utiliza la cámara térmica Flir Lepton.

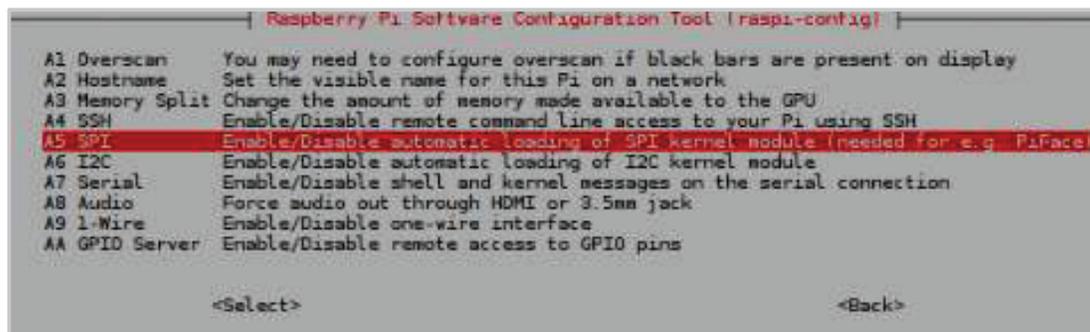


Figura 2.36 Habilitar interfaces SPI – I2C

### 3.1.2.3 Pruebas de Cámara Térmica

Para la comprobación del funcionamiento de la cámara se utilizó la aplicación QT desarrollada por la empresa Pure Engineering.

Como primer paso la tarjeta Raspberry Pi debe estar conectada a Internet para poder descargar todos los ficheros necesarios para poder probar la cámara térmica.

Se ingresa en el terminal la siguiente sentencia: `sudo apt-get install qt4-dev-tools`, luego que se completó la descarga se descomprime la carpeta con la sentencia: `unzip LeptonModule-master.zip`.

Una vez que se desempaquetados los archivos se ingresó la siguiente sentencia: `cd /home/pi/LeptonModule-master/raspberrypi_video/leptonSDKEmb32PUB` y se procede a compilar el programa utilizando el comando “make”.

Posteriormente se procede a ingresar en la carpeta `raspberrypi_video` donde se ejecuta el comando “`qmake && make`” para construir el programa portable con el comando “`qmake`”, se compila con el comando “`make`”.

Finalmente, una vez realizado los pasos anteriores se ejecuta el programa con la sentencia `sudo ./raspberrypi_video`. En la Figura 2.37 se puede observar el resultado de lo que capta la cámara térmica.

En la captura de prueba se observa que la imagen con mayor cantidad de radiación térmica la parte en amarillo.



Figura 2.37 Pruebas cámara térmica

### 3.1.3 CONFIGURACIÓN MÓDULO GPS

En el presente trabajo de titulación se utilizó el módulo de GPS NEO-7M-C para obtener la ubicación geográfica, fecha, hora de las imágenes térmicas para lo cual se requiere como primer paso la conexión física del módulo.

La conexión del módulo se lo realizó de acuerdo con la Figura 2.7 presentado en el capítulo 2.

#### 3.1.3.1 Instalación de librerías

Para configurar el módulo GPS se debe instalar las librerías para que procesen la información enviada por el módulo y mediante la ejecución de *scripts* se pueda presentar dicha información que para el presente trabajo son: longitud, latitud, tiempo y altitud.

Una vez encendida la tarjeta Raspberry Pi 3, desde el terminal se ingresa la sentencia *dmseg | grep -i tty*, mediante este comando se puede ver en qué puerto serial se encuentra conectado el módulo.

En la Figura 2.38 se presenta el resultado de la ejecución de la sentencia *dmseg*.

Previamente a este paso se debe revisar el archivo *cmdline.txt* que está ubicado en la dirección */boot*. Con la modificación de este archivo se desbloquea el puerto serial.

Por defecto el archivo contiene la siguiente información:

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200
console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait
```

Se modifica el archivo y se guarda con la siguiente información:

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4
elevator=deadline rootwait
```

Para que los cambios realizados tengan efecto se procede al reinicio de la tarjeta Raspberry Pi 3, con esto se tiene el puerto serial para el envío y recepción de información.

Se conecta la tarjeta Raspberry Pi a Internet para poder descargar las dependencias que ayudan a interpretar los datos que el Raspberry Pi recibe del módulo de GPS. La sentencia que se utilizó es la siguiente:

```
sudo apt-get install gpsd gpsd-clients python-gps
```

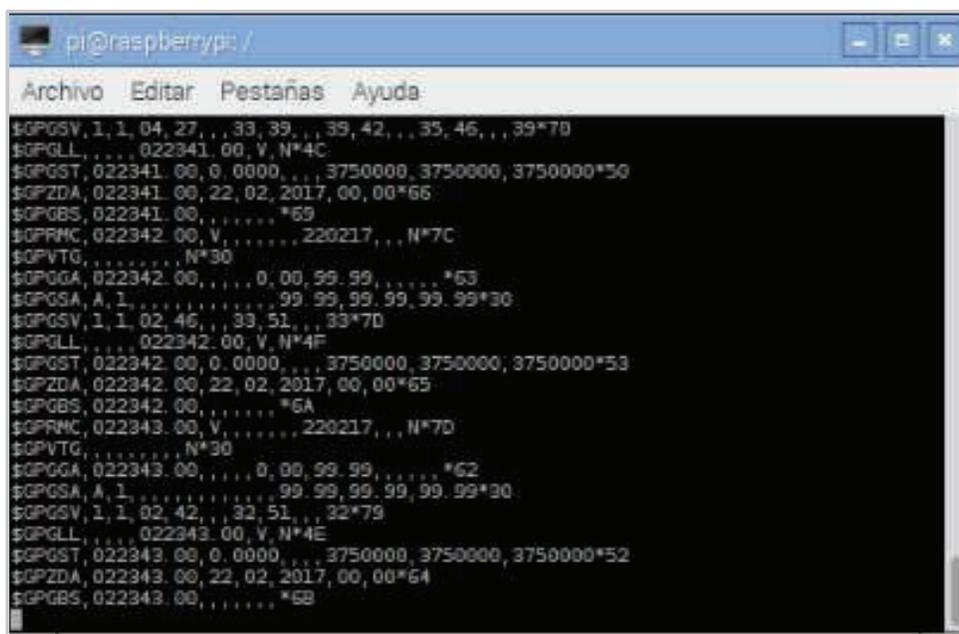
GPSD es una aplicación que interpreta el protocolo NMEA y permite visualizar la información que se recibe del módulo GPS a través de un socket que debe ser enlazado con el puerto serial donde está instalado el GPS.

La sentencia utilizada para enlazar el socket con el puerto serial es la siguiente:

```
sudo gpsd /dev/ttyAMA0 -F /var/run/gpsd.sock.
```

### 3.1.3.2 Pruebas del Módulo GPS

Para comprobar el módulo GPS funcione correctamente se debe leer los datos del puerto serial donde se encuentra conectado, para esto se hace uso del comando “sudo cat /dev/ttyAMA0”. En la Figura 2.38 se puede observar los datos que envía el módulo GPS en el formato del protocolo NMEA.



```
pi@raspberrypi: /
Archivo Editar Pestañas Ayuda
$GPGSV,1,1,04,27,,33,39,,39,42,,35,46,,39*70
$GPGLL,,022341.00,V,N*4C
$GPGST,022341.00,0.0000,,3750000,3750000,3750000*50
$GPZDA,022341.00,22,02,2017,00,00*66
$GPGBS,022341.00,,,,,*69
$GPRMC,022342.00,V,,,,,220217,,N*7C
$GPVTG,,,,,N*30
$GPGGA,022342.00,,,,,0,00,99.99,,,,,*63
$GPGSA,A,1,,,,,99.99,99.99,99.99*30
$GPGSV,1,1,02,46,,33,51,,33*7D
$GPGLL,,022342.00,V,N*4F
$GPGST,022342.00,0.0000,,3750000,3750000,3750000*53
$GPZDA,022342.00,22,02,2017,00,00*65
$GPGBS,022342.00,,,,,*6A
$GPRMC,022343.00,V,,,,,220217,,N*7D
$GPVTG,,,,,N*30
$GPGGA,022343.00,,,,,0,00,99.99,,,,,*62
$GPGSA,A,1,,,,,99.99,99.99,99.99*30
$GPGSV,1,1,02,42,,32,51,,32*79
$GPGLL,,022343.00,V,N*4E
$GPGST,022343.00,0.0000,,3750000,3750000,3750000*52
$GPZDA,022343.00,22,02,2017,00,00*64
$GPGBS,022343.00,,,,,*6B
```

Figura 2.38 Salida del puerto ttyAMA0

Adicionalmente se utilizó la utilidad `cgps -s` para visualizar la información de ubicación que se recibió como se puede ver en la Figura 2.39.

Time:	2017-03-13T02:31:47.000Z	PRN:	Elev:	Azim:	SNR:	Used:
Latitude:	0.274551 S	6	58	274	33	Y
Longitude:	78.535116 W	17	21	345	33	Y
Altitude:	2926.7 m	28	45	022	38	Y
Speed:	0.1 kph	30	64	207	37	N
Heading:	90.5 deg (true)	133	67	271	28	N
Climb:	-1.6 m/min					
Status:	3D FIX (3 secs)					
Longitude Err:	n/a					
Latitude Err:	n/a					
Altitude Err:	+/- 141 m					
Course Err:	n/a					
Speed Err:	+/- 2 kph					
Time offset:	-565037.147					
Grid Square:	FI09rr					

Figura 2.39 Aplicativo cgps -s

### 3.1.4 CONFIGURACIÓN RED AD HOC

La comunicación entre el Sistema Aéreo y el Sistema Tierra se realizó por medio de una red ad hoc.

Para crear una red ad hoc en el sistema operativo Windows 7 se siguieron los siguientes pasos:

Se ingresa en Panel de control\Redes e Internet\Centro de redes y recursos compartidos. En la Figura 2.40 se presenta la ruta para el acceso a la sección “Configurar una nueva conexión o red”.



Figura 2.40 Creación Red AD HOC

Se selecciona Configurar una nueva conexión o red para crear la red ad hoc y de inmediato se presenta un *wizar* de configuración. En la Figura 2.41 se puede observar el inicio del *wizar* para configurar una red ad hoc.

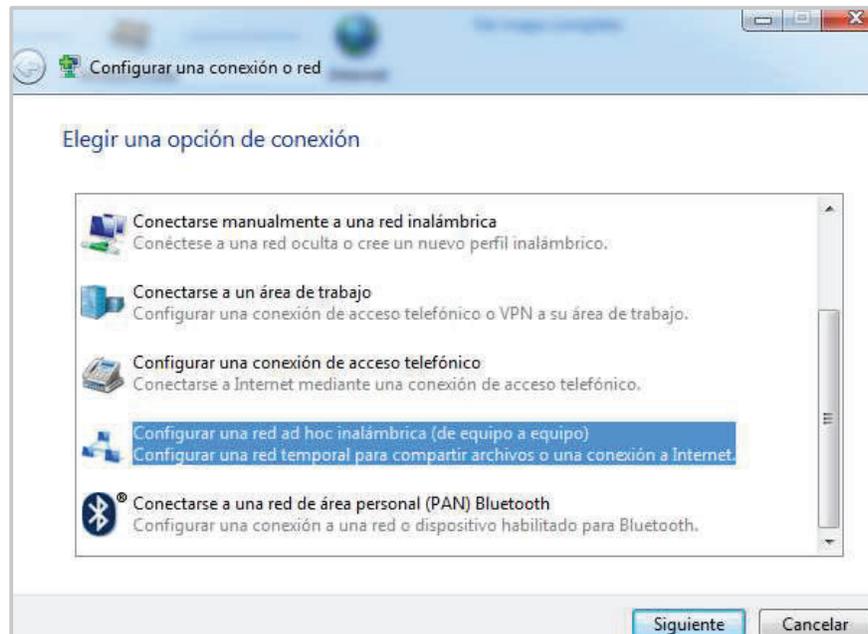


Figura 2.41 Configuración red ad hoc

En la Figura 2.42 se puede ver los parámetros de la red ad hoc se creó. El nombre de la red es "NETWORK". No se colocó contraseña de seguridad.

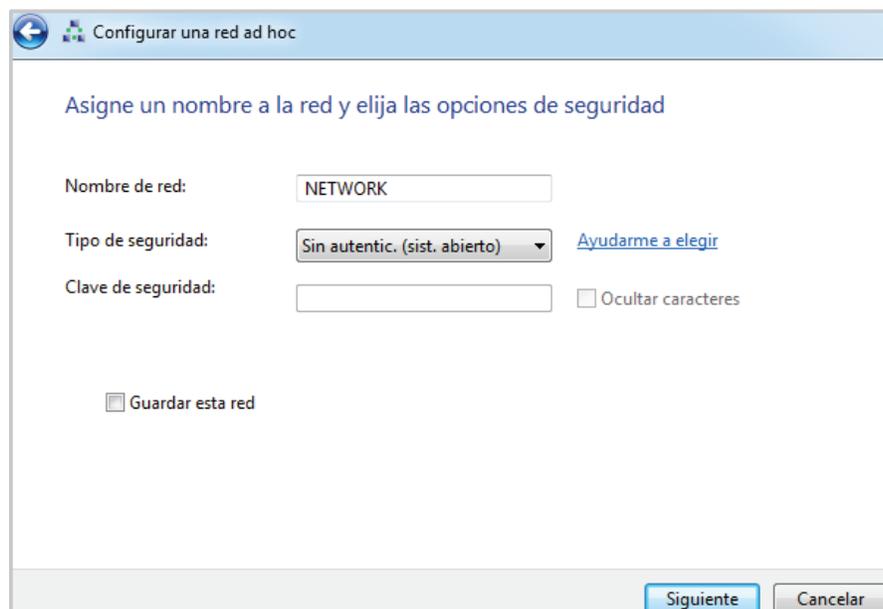


Figura 2.42 Parámetros de configuración

Una vez creada la red ad hoc se procedió a configurar la tarjeta inalámbrica del computador portátil. La dirección asignada será 10.10.10.2/30 ya que solo se necesita dos direcciones IP para el presente trabajo como se puede ver en la Figura 2.43.

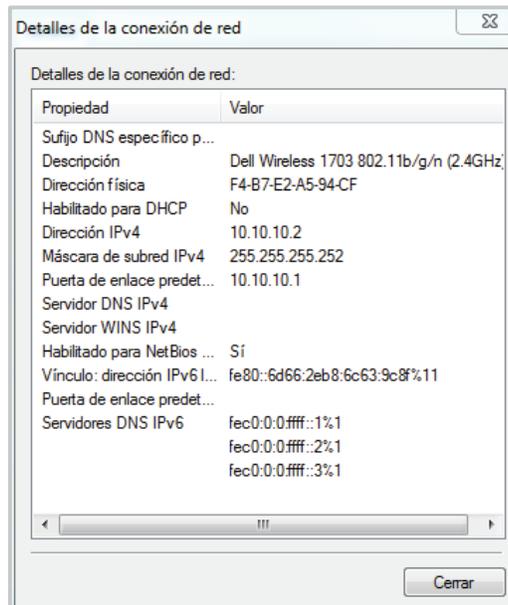


Figura 2.43 Dirección IP Sistema Tierra

### 3.1.4.1 Pruebas de conectividad

Para comprobar que existe comunicación entre los dos sistemas se utilizó el comando ping para comprobar el estado de la comunicación desde la tarjeta Raspberry Pi hacia el Sistema Tierra y viceversa como se puede apreciar en la Figura 2.44.

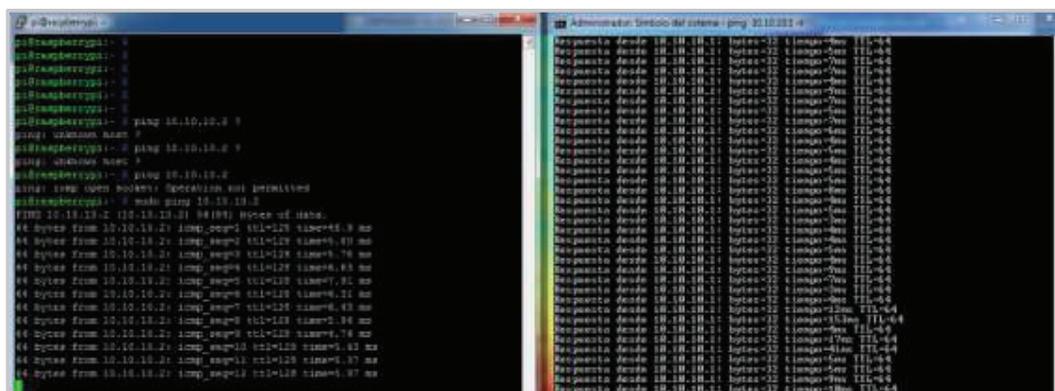


Figura 2.44 Pruebas de conectividad

## 3.2 IMPLEMENTACIÓN DE SOFTWARE

### 3.2.1 IMPLEMENTACIÓN SOFTWARE DEL SISTEMA AÉREO

El software del Sistema Aéreo está conformado por dos *scripts* realizados en el lenguaje de programación Python. Los *scripts* en términos generales son encargados de capturar la información tanto ubicación como imagen y enviar al Sistema Tierra.

#### 3.2.1.1 Script Programa\_GPS

En el Fragmento de Código 2.1 primero se importan las librerías que se van a utilizar en el *script* “Programa\_GPS.py”.

Se utilizó la librería *os* que permite acceder a las funciones del sistema operativo, así como al entorno del mismo. Para el procesamiento de fechas y tiempo se utiliza las librerías *dateutil.parser*, *datetime*, *dateutil*. Para interactuar con el módulo GPS se necesita la librería *gps*.

```
# -*- coding: utf-8 -*-
import os
import dateutil.parser
from gps import *
from datetime import datetime
from dateutil import tz
```

Fragmento de Código 2.1 Librerías para Script “Programa\_GPS”.

En el Fragmento de Código 2.2 se abre la conexión con el módulo GPS para obtener la información y en las siguientes dos líneas de código se obtiene las zonas horarias tanto local como la zona UTC.

```
session = gps(mode=WATCH_ENABLE) #Iniciar una sesión del GPS.
zona_local = tz.tzlocal() #Se obtiene la zona horaria local.
zona_utc = tz.tzutc() #Se obtiene la zona horaria en formato UTC.
```

Fragmento de Código 2.2 Conexión a módulo GPS.

Luego de abrir la conexión con el GPS y tener la información necesaria se procede con el tratamiento de la ubicación para presentarlo de acuerdo con nuestra zona horaria local.

El procesamiento de información que entrega el módulo GPS se la realiza de forma continua por medio de un lazo infinito utilizando la sentencia *while*.

En el Fragmento de Código 2.3 se muestra cómo se obtiene los valores de interés como la latitud, longitud, altitud, fecha y hora mediante la sentencia `session.fix.<valor gps>`.

En caso de no recibir información de fecha y hora por cualquier tipo de desconexión el programa no se detendrá y marca cualquier tipo de las variables antes dichas con el valor “nan”. El valor “nan” es utilizado por el protocolo NMEA para identificar cuando no se recibe.

```
while 1:#Lazo infinito
os.system('clear')#Limpia la consola
session.next()#El objeto session llama al metodo next que obtiene la siguiente lectura del GPS.
latitud=str(session.fix.latitude)#Se extrae la propiedad latitud.
longitud=str(session.fix.longitude)#Se extrae la propiedad longitud.
altitud=str(session.fix.altitude)#Se extrae la propiedad altitud.
tiempoGPS=str(session.utc)#Se extrae el tiempo.

if tiempoGPS=="nan" or tiempoGPS=="":
    fecha_final ="nan"
```

Fragmento de Código 2.3 Procesamiento de información GPS

En el Fragmento de Código 2.4 se realiza un procesamiento de la fecha y hora ya que el módulo de GPS entrega esta información en la zona horaria UTC y con caracteres que no son de interés como se puede ver en la Figura 2.45.



Figura 2.45 Fecha y Hora UTC

Se utiliza los métodos de la clase *datetime* para transformar a los valores de fecha y hora a un valor del tipo *string* y posteriormente se eliminó los caracteres “T” y “.000Z”.

Finalmente se obtiene un valor de fecha y hora de acuerdo con la zona local.

```
else:
    try:
        tiemposinT= datetime.datetime.strptime(tiempoGPS)#Se transforma la informacion a un string.
        stringtiemposinT=tiemposinT.strftime("%Y-%m-%d %H:%M:%S")#Se modifica el formato de la fecha y hora.
        utc = datetime.datetime.strptime(stringtiemposinT, "%Y-%m-%d %H:%M:%S")#Se transforma a fecha.
        utc = utc.replace(tzinfo=zona_utc)#Se le indica la zona horaria UTC(Coordenada de tiempo Universal).
        fecha_final = utc.astimezone(zona_local)#Se transforma a UTC -3.
    except:
        fecha_final ="nan"
```

Fragmento de Código 2.4 Transformación de fecha y hora a zona local

En el Fragmento de Código 2.5 se presenta la parte final del *script* y muestra como los datos procesados se procedieron a guardar en el archivo gps.txt.

```

archivotxt=open("/home/pi/Programas/gps.txt",'w')
archivotxt.write(latitud+'@'+longitud+'@'+str(fecha_final)+'@'+altitud)
archivotxt.close()

```

Fragmento de Código 2.5 Escritura de archivo gps.txt

Los datos se sobrescriben cuando el programa realiza una nueva lectura de datos del módulo de GPS. El tiempo que realiza cada lectura de aproximadamente cada segundo.

### 3.2.1.2 Script Programa\_Camara

El segundo *script* que conforma el software aéreo permite que se obtengan imágenes térmicas de forma continua, ensambla con la información de ubicación, fecha y hora, pero también contiene líneas de programación que corresponden al servidor TCP que recibe las peticiones del Sistema Tierra.

De la misma manera que el *script* Programa\_GPS las primeras líneas de código son las librerías que se utilizan en el desarrollo del programa como se puede ver en el Fragmento de Código 2.6.

```

import socket
import sys
import os, time
from gps import *
import numpy as np
import cv2
from pylepton import Lepton
import base64
from thread import *

```

Fragmento de Código 2.6 Librerías *script* Programa\_Camara

Las librerías utilizadas sirven para la captura y procesamiento de imágenes térmicas, conexión con Sistema Tierra, transformación de información.

Se importa la librería *socket* porque el envío de información se hará a través de sockets. La librería *os* permite abrir y leer el archivo donde se guarda la información de ubicación obtenida utilizando el módulo GPS. La librería *numpy* permite

interactuar con arreglos de bytes, y *cv2* es usada para el procesamiento de imágenes. Con *base64* se puede realizar la codificación de bytes en un *string* de *base64*, y *thread* para el uso de hilos. Para la captura de imágenes térmicas, se usa la librería *pylepton*.

Para la conexión entre el Sistema Aéreo y el Sistema Tierra se programó un servidor TCP donde se define la dirección IP y el puerto por donde el servidor va a escuchar la petición del Sistema Tierra como se puede verificar en el Fragmento de Código 2.7 .

```
HOST = '10.10.10.1' #Direccion IP del interfaz que escucha
PORT = 8082 #Puerto que escucha

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM) #Crear el socket IPv4-Stream.
print 'Socket creado'
#Bind socket to local host and port
try:
    s.bind(('10.10.10.1', 8082)) #Utilizo el metodo bind para unir socket con la IP y el puerto
except socket.error as msg:
    print 'Error : ' + str(msg[0]) + ' Error' + msg[1]
    sys.exit()
print 'Socket bind completo'
s.listen(1) #Socket comienza a escuchar (esperar conexiones). 1 número conexión.
print 'Socket escuchando'
```

Fragmento de Código 2.7 Servidor TCP

En el Fragmento de Código 2.8 se presentan las líneas de código que abren y realizan una lectura del archivo “*gps.txt*” donde se encuentra almacenada la información de ubicación, fecha y hora.

```
while 1:

    #Abrir Leer y Cerrar archivo con informacion de ubicacion, fecha y hora
    archivotxtGPS=open("/home/pi/Programas/gps.txt", 'r') #Se abre el archivo para leer(r)
    respuestaGPS=archivotxtGPS.read()
    archivotxtGPS.close()
```

Fragmento de Código 2.8 Lectura de archivo *gps.txt*

En el Fragmento de Código 2.9 se presenta la ejecución de la captura de las imágenes térmicas que son capturadas de forma continua. Para el efecto se captura la imagen, se realiza una normalización de los bits de cada pixel de la imagen y se guarda la imagen con el nombre de “*Foto\_Normalizada.jpg*”.

En las tres últimas líneas del Fragmento de Código 2.9 se ejecuta el procesamiento de la imagen que consiste en obtener la imagen en escala de grises y posterior presentar con una paleta de colores donde se puede observar las diferencias térmicas.

```

#Tomar foto
with Lepton() as camara:
    foto_tomada, idFoto = camara.capture() #Captura de imagen
foto_normalizada=foto_tomada
foto_8=foto_normalizada
cv2.normalize(foto_tomada, foto_normalizada, 0, 65535, cv2.NORM_MINMAX) # Se normaliza el arreglo de 12 bits
np.right_shift(foto_normalizada, 8, foto_8) # Luego de normalizar se hace un arreglo con 8 bits.
cv2.imwrite("Foto_Normalizada.jpg", np.uint8(foto_8)) # Se guarda la imagen termica

imagen_grisea=cv2.imread("Foto_Normalizada.jpg",cv2.IMREAD_GRAYSCALE)#La foto convierte a una escala de grises
imagen_termica=cv2.applyColorMap(imagen_grisea,cv2.COLORMAP_JET)#A la foto le pone una paleta de colores
cv2.imwrite("Foto_Termica.jpg",imagen_termica)#

```

#### Fragmento de Código 2.9 Captura de la foto térmica

Para realizar el envío de la imagen térmica al Sistema Tierra hay que realizar el cálculo de tamaño de la imagen térmica para lo cual se abre el archivo “Foto\_Termica.jpg” y mediante el método `seek ()` se puede mover al final de la cadena de bits que contiene la imagen y calcular su número de bits.

El tamaño de la imagen se guarda en la variable “buf” para ser utilizado en la transmisión del paquete de información al Sistema Tierra.

```

#Leer archivo de la foto
archivoImg=open("Foto_Termica.jpg",'r')
#Obtener el tamaño del arreglo de bytes de la foto.
archivoImg.seek(0,2)#mover el cursor al final de la fila.
tamañoImg=archivoImg.tell()#El valor del posición y este el número de bits.
strTamañoImg=str(tamañoImg)
archivoImg.seek(0,0)#Para regresar a la primera posición
buf=archivoImg.read(tamañoImg)#Lee las información de acuerdo al tamaño "tamañoImg"
archivoImg.close()

```

#### Fragmento de Código 2.10 Procesamiento de la imagen “Foto\_Termica.jpg”

La información se envía de un solo paquete que contiene ubicación, fecha, hora, tamaño de la imagen y la imagen térmica en una sola cadena del tipo *string*.

La imagen térmica es transformada primero a base64 para no ser alterada en el envío y de ahí es transformada a código ASCII para poder ser enviada al Sistema Tierra como una cadena de caracteres. Se utiliza la variable “buf” que se encuentra en el Fragmento de Código 2.11 y que corresponde al tamaño de la imagen térmica, que sirve para que el Sistema Tierra pueda decodificar dicho paquete. Adicionalmente la información es separada por medio del carácter “@”.

```

imgBase64=base64.b64encode(buf).decode('ascii')#Se transforma a ascii.
print 'Enviando: '+respuestaGPS+'@'+strTamañoImg+'@'+imgBase64
conexion.send(respuestaGPS+'@'+strTamañoImg+'@'+imgBase64) #Se envía el paquete.
print 'Enviando Imagen: '+strTamañoImg +imgBase64

```

#### Fragmento de Código 2.11 Envío de información a Sistema Tierra

### 3.2.1.3 Automatización de los Scripts

Los scripts del Sistema Aéreo se ejecutan automáticamente una vez que la tarjeta Raspberry Pi se enciende por lo que se generaron dos archivos con extensión `.service` y están localizados en la ruta `/etc/systemd/system`.

En la Figura 2.46 se presenta el servicio `Servicio_GPS.service` que ejecuta el *script* `Programa_GPS.py`.

```
[Unit]
Description=Servicio GPS
After=Servicio_Echo.service

[Service]
Type=idle
ExecStart=/usr/bin/sudo /usr/bin/python /home/pi/Programas/Programa_GPS.py

[Install]
WantedBy=multi-user.target
```

Figura 2.46 Servicio `Servicio_GPS.service`

En la Figura 2.47 se muestra el servicio que maneja a la cámara térmica y al servidor TCP para la transmisión de los datos al Sistema Tierra.

```
[Unit]
Description=Servicio Camara
After=sshd.service

[Service]
Type=idle
ExecStart=/usr/bin/python /home/pi/Programas/Programa_Camara.py

[Install]
WantedBy=multi-user.target
```

Figura 2.47 Servicio `Servicio_Camara.service`

### 3.2.1.4 Pruebas de funcionamiento

Para comprobar el funcionamiento en cada uno de los scripts se añadió sentencias de *print* y se procedió a ejecutar individualmente cada *script* para verificar los datos que entregan cada uno de los *scripts*.

En la Figura 2.48 se presenta el resultado que se obtiene al ejecutar el *script* `Programa_GPS.py`.



### 3.2.2 IMPLEMENTACIÓN DE SOFTWARE DE SISTEMA TIERRA

En el Sistema Tierra se realizó un programa en la plataforma Microsoft Visual Studio con el lenguaje de programación C#, que tiene como tareas principales la recepción de información, presentación de imágenes, ubicación, fecha y hora, además captura imágenes de interés juntamente con la información de coordenadas geográficas y el tiempo.

En la Figura 2.50 se presenta el interfaz principal del Sistema Tierra donde se presenta los datos de latitud, longitud, altitud y el *streaming* de video térmico. Adicionalmente el interfaz principal tiene tres botones que permite al usuario capturar la información de interés, ver las imágenes capturadas y el botón que realiza la conexión con el Sistema Aéreo.



Figura 2.50 Interfaz principal

A continuación, se muestran los principales fragmentos de código que se han utilizado para realizar la programación del interfaz principal de la Figura 2.50.

Para la comunicación entre el Sistema Aéreo y el Sistema Tierra se requiere realizar la conexión utilizando el protocolo TCP y como se mencionó en el apartado 3.2.1.2 en el Sistema Aéreo se realizó un servidor TCP por lo que en el Sistema Tierra se realizó el cliente TCP como se puede ver en el Fragmento de Código 2.12.

De forma similar al script del Sistema Aéreo para construir el *socket* se define la dirección IP y el puerto de conexión, mediante las librerías de C# se crea el *socket*,

se realiza la conexión y esta se asigna al hilo que se lo nombró “HiloRecibirDatos”. En caso de que no se tenga conexión con el Sistema Aéreo se presenta el mensaje “SIN CONEXIÓN AL SISTEMA AÉREO” como se puede ver en la Figura 2.51.



Figura 2.51 Mensaje de conexión

```
//Método que maneja la comunicación con el servidor
private void ComunicacionTCP()
{
    socketCliente = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
        ProtocolType.Tcp); //Crear Socket, IPv4-Stream-TCP

    IPAddress ipRemota = IPAddress.Parse("18.10.10.1"); //IP remota (servidor)
    int puertoRemoto = Int32.Parse("8082"); //Puerto remoto (servidor)
    IPEndPoint ipEndPointRemoto = new IPEndPoint(ipRemota, puertoRemoto); //IP y Puerto remoto (servidor)

    try
    {
        socketCliente.Connect(ipEndPointRemoto); //Intentar conectar con el servidor
        if (socketCliente.Connected) //Si está conectado Te devuelve un valor booleano
        {
            hiloRecibirDatos = new Thread(new ThreadStart(RecibirDatos)); //Asignar el método RecibirDatos al hilo
            hiloRecibirDatos.Start(); //Iniciar el hilo
        }
    }
    catch (Exception)
    {
        MessageBox.Show("SIN CONEXIÓN AL SISTEMA AÉREO");
    }
}
```

Fragmento de Código 2.12 Comunicación con Sistema Aéreo

Una vez que se ha asignado un hilo al método “RecibirDatos” se realiza la recepción de la información, se guarda en el variable “mensajeTotal” del tipo *string* que será utilizado por el método “RecibirTexto”.

```
int cantidadBytesRecibidos = socketCliente.Receive(buferRecibirDatos); //Recibir los datos

string mensajeFinal=null; //Variable para recibir el mensaje

if (cantidadBytesRecibidos > 0) //Si se han recibido datos
{
    //Recortar la cadena para incluir únicamente a los datos recibidos
    //Substring cortar desde 0 hasta lo recibido

    string mensajeTotal = Encoding.ASCII.GetString(buferRecibirDatos);
    mensajeFinal = mensajeTotal.Substring(0, cantidadBytesRecibidos);

    RecibirTexto(mensajeFinal); //Llamar al método Recibir Texto, encargado de decidir que hacer con el string
}
```

Fragmento de Código 2.13 Mensaje del tipo string

Finalmente, con el mensaje del tipo *string* se procede a la fragmentación del mismo utilizando como bandera de división el carácter “@” que fue introducido en el Sistema Aéreo previo al envío. En el Fragmento de Código 2.14 se puede observar cómo se realiza la división del mensaje. Con la información fragmentada se actualiza la información invocando a través del método *BeginInvoke* al método que actualiza los valores en la interfaz

```
private void RecibirTexto(string mensajeFinal)
{
    //El formato del mensaje es Longitud@Latitud@Tiempo@Altitud@Longitud del mensaje@Imagen
    string[] mensaje = mensajeFinal.Split('@');//Dividimos con un split y lo guardamos en el arreglo de string
    captura.Latitud = mensaje[0];
    captura.Longitud = mensaje[1];
    captura.TiempoGPS = mensaje[2];
    captura.Altitud = mensaje[3];

    //El string de la imagen debe ser convertida de Base 64
    byte[] byteImagenRecibida = Convert.FromBase64String(mensaje[5]);

    //Transformar los bytes a imagen y la guardamos en una variable del tipo imagen
    Image imgRecibida = byteArrayToImage(byteImagenRecibida);
    captura.Imagen = imgRecibida;

    //Invocar al metodo que actualice los datos en el formulario
    this.BeginInvoke(delegadoForm);
    Thread.Sleep(300);//
}
```

#### Fragmento de Código 2.14 Fragmentación del mensaje

En el interfaz principal se encuentra tres botones que permite realizar acciones como, conectar con el Sistema Aéreo, capturar imágenes, capturar información de ubicación y tiempo, así como también verificar las imágenes de interés que han sido capturadas. En la Figura 2.52 se presenta dichos botones que abren dos formularios para presentar la información.



Figura 2.52 Botones del interfaz principal

El botón Conectar realiza la comunicación entre el Sistema Tierra y el Sistema Aéreo por medio de una conexión TCP. En el Fragmento de Código 2.15 se presenta el método del botón Capturar que contiene el método “Comunicación” que se revisó en el Fragmento de Código 2.12.

```

1reference
private void btnRecibirDatos_Click(object sender, EventArgs e)
{
    ComunicacionTCP();//Llamar al método que controla la comunicación TCP
}

```

**Fragmento de Código 2.15 Código botón Conectar**

El botón Capturar es el siguiente botón que entra en funcionamiento una vez que están conectados los sistemas, sí el usuario detecta alguna imagen de interés, este puede almacenar la imagen, ubicación, tiempo y altitud en una base de datos. En el Fragmento de Código 2.16 se presentan las líneas de programación que ejecuta este procedimiento.

```

ImagenCapturadaBDD objetoBDD = new ImagenCapturadaBDD();//Crear objeto que interactúa con la BDD
Image imgAGuardar=new Bitmap(pcbCamara.Image);//Copiar la imagen del picture box en una variable
//Guardar la imagen actual del formulario
objetoBDD.GuardarImagen(txtTiempo.Text, txtLatitud.Text, txtLongitud.Text, txtAltitud.Text, imgAGuardar);
//Nos aseguramos lo que estamos viendo se deba guarda.

```

**Fragmento de Código 2.16 Código botón Capturar**

En el Fragmento de Código 2.17 se presenta el código que se utiliza para almacenar la información de interés para el usuario, en esta sección se puede verificar la conexión a la base de datos y la ejecución del comando insert.

```

public int GuardarImagen(string tiempoGPS, string latitudF, string longitudF, string altitudF, Image imagenInfrarrojaF)
{
    int resultado = 0;//Variable que indica si se logró guardar la imagen.

    byte[] imagenBytesParaEnviar = imageToByteArray(imagenInfrarrojaF);//Transformar la imagen a arreglo de bytes.

    //Se crea la sentencia SQL que se va a ejecutar para la BDD
    string sentenciaSqlEditorEvento = "INSERT INTO Imagenes (ImgTiempoGPS, ImgLatitudGPS,ImgLongitudGPS,ImgAltitudGPS,ImgImagenCamara) "+
        "VALUES ('"+tiempoGPSF+"','"+latitudF+"','"+longitudF+"','"+altitudF+"', @imagenEnBytes)";

    //Conexion a la base de datos , pasando como argumento la cadena de conexion
    SqlConnection conexion = new SqlConnection(cadenaConexion);
    //Comando, q se ejecutara con una sentencia y con la conexion a la BD
    SqlCommand comando = new SqlCommand(sentenciaSqlEditorEvento, conexion);
    // transforma la imagen de bytes a binario y la coloca en la sentencia SQL
    comando.Parameters.Add("@imagenEnBytes", SqlDbType.VarBinary, imagenBytesParaEnviar.Length).Value = imagenBytesParaEnviar;
}

```

**Fragmento de Código 2.17 Método GuardarImagen**

El botón Ver Imágenes permite verificar el historial de capturas obtenidas donde se podrá ver la información que se capturó. En la Figura 2.53 se presenta el formulario

“Imágenes Térmicas” donde se puede observar el registro de lo que obtiene el usuario.

Tempo	Latitud	Longitud	Altitud	Imagen
2017-02-09 21:30:09.00	-0.216143207	-76.53523207	2363.262	
2017-02-09 20:03:06.00	-0.216154826	-76.535243222	2369.952	
2017-02-09 20:03:07.00	-0.216158114	-76.535243634	2370.333	
2017-02-09 20:03:08.00	-0.216170803	-76.535243228	2370.477	
2017-02-09 20:03:09.00	-0.216197977	-76.535239422	2370.624	
2017-02-09 20:03:19.00	-0.216205379	-76.535259179	2368.961	
2017-02-09 20:03:20.00	-0.216197988	-76.535259115	2369.111	
2017-02-09 20:03:21.00	-0.216202987	-76.535259134	2369.264	
2017-02-09 20:03:22.00	-0.216202996	-76.535259152	2369.317	
2017-02-09 20:03:23.00	-0.216206447	-76.535259204	2369.361	
2017-02-09 20:03:24.00	-0.216207322	-76.535259177	2369.411	
2017-02-09 20:03:25.00	-0.216209907	-76.535259266	2367.279	
2017-02-09 20:03:27.00	-0.216210196	-76.535259304	2369.174	
2017-02-09 20:03:29.00	-0.216209774	-76.535259393	2369.475	
2017-02-10 08:10:52.00	-0.216234703	-76.535235406	2370.209	
2017-02-10 08:10:56.00	-0.216271964	-76.535107615	2369.331	
2017-02-10 08:12:57.00	-0.216271964	-76.535107615	2369.721	
2017-02-10 15:42:15.00	-0.216280324	-76.535306626	2340.625	

Figura 2.53 Registro de capturas térmicas

Los datos presentados en el formulario de la Figura 2.53 son obtenidos por medio de la base de datos que se conectó de forma automática como se puede ver en la Figura 2.54.

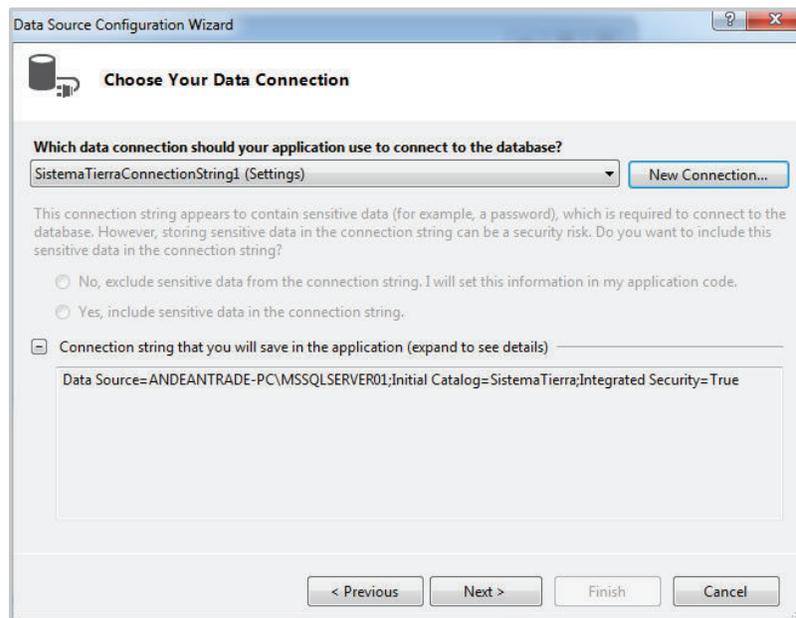


Figura 2.54 Conexión base de datos

Si se realiza una consulta en la base de datos SQL se puede verificar que están todas las imágenes capturadas como se puede ver en el Figura 2.55. Para comprobar el funcionamiento de esta parte se ingresa en el *Query* el comando `select * from Imagenes`, como resultado se obtiene toda la información de la base de datos. La base de datos almacena en una sola tabla toda la información.

SQLQuery2.sql - A...AndeanTrade (52) \* x

```

use SistemaTierra
select * from Imagenes

```

100 %

Results Messages

ImgID	ImgTiempo	ImgTiempoGPS	ImgLatitudGPS	ImgLongitudGPS	ImgAltudGPS	ImgImagen_Cadena
10	2017-01-06 11:45:38.237	2017-01-06T16:39:04.000Z	-78.437767167	-0.004901	2446.0	0xFFD8FFE00104A46494600010101006000600000FFD800...
11	2017-01-06 11:40:40.940	2017-01-06T16:39:04.000Z	-78.437767167	-0.004901	2446.0	0xFFD8FFE00104A46494600010101006000600000FFD800...
12	2017-01-06 11:40:49.090	2017-01-06T16:39:13.000Z	-78.43775936	-0.004890108	2445.428	0xFFD8FFE00104A46494600010101006000600000FFD800...
13	2017-01-06 11:40:50.203	2017-01-06T16:39:13.000Z	-78.437776	-0.004890167	2444.8	0xFFD8FFE00104A46494600010101006000600000FFD800...
14	2017-01-06 11:40:51.343	2017-01-06T16:39:14.000Z	-78.437777145	-0.004890289	2445.288	0xFFD8FFE00104A46494600010101006000600000FFD800...
15	2017-01-06 11:40:56.637	2017-01-06T16:39:17.000Z	-78.437780453	-0.00489203	2446.091	0xFFD8FFE00104A46494600010101006000600000FFD800...

Figura 2.55 Consulta base de datos

El tercer formulario que tiene el Sistema Tierra permite visualizar a detalle la imagen capturada como también navegar por todos los registros capturados y finalmente si es elección del usuario se puede eliminar la imagen térmica. En la Figura 2.56 se presenta el formulario Datos Imagen.

Latitud: -78.437779867

Longitud: -0.004886129

Tiempo: 2017-01-06T16:39:21.000Z

Altitud: 2446.869

Figura 2.56 Formulario Datos Imagen

Para ingresar a verificar el detalle de una imagen se debe dar clic sobre cualquier registro de interés del formulario “Imágenes Térmicas” presentado en la Figura 2.55.

A continuación, se presenta el Fragmento de Código 2.18 que permite realizar una lista de los registros guardados, posteriormente permite realizar operaciones como visualizar la imagen, navegar entre los registros y borrar los registros guardados.

```
int numFila = e.RowIndex;
int id= Convert.ToInt32(dgvCapturas.Rows[numFila].Cells[0].Value.ToString());//Primero se
//le convierte a un valor string - Celda o columna cero esta el valor del ID- me interesa el valor del ID

//Conseguir la lista de ID
int esteId = 0;//Variable para leer cada ID
int numerodeFila = 0;
int numeroDeFotosCapturadas=dgvCapturas.Rows.Count;
int[] listaDeId = new int[numeroDeFotosCapturadas];//Lista donde se van a guardar los IDs

foreach (DataGridViewRow fila in dgvCapturas.Rows)//Recorro cada fila
{
    esteId = (int)fila.Cells[0].Value;//Tomo el valor de cada celda de la columna 0 (donde estan los IDs)
    listaDeId[numerodeFila]=esteId;//Añado ID a la lista
    numerodeFila++;
}
```

Fragmento de Código 2.18 Código para enlistar los registros de guardados

Con el *id* de la base de datos se ejecuta el método “PresentarImagen” que permite mostrar en el formulario Captura Detalle de forma individual la información guardada. En Fragmento de Código 2.19 se presenta el método mencionado.

```
private void PresentarImagen()
{
    ImagenCapturadaBDD objetoBDD = new ImagenCapturadaBDD();//Objeto para interactuar con la BDD

    ImagenCapturada imagenDetalle = objetoBDD.ObtenerImagen(idImagen.ToString());//Obtener la imagen de la BDD

    pcbCamara.Image = imagenDetalle.Imagen;//Presentar la Imagen
    txtAltitud.Text = imagenDetalle.Altitud;//Presentar la Altitud
    txtLatitud.Text = imagenDetalle.Latitud;//Presentar la Latitud
    txtLongitud.Text = imagenDetalle.Longitud;//Presentar la Longitud
    txtTiempo.Text = imagenDetalle.TiempoGPS;//Presentar el Tiempo
}
```

Fragmento de Código 2.19 Código para obtener la imagen detalle

### 3.3 CONSTRUCCIÓN DEL SISTEMA AÉREO

Durante la construcción del Sistema Aéreo se debió realizar modificaciones a la estructura que iba a contener los elementos electrónicos, se tomó en cuenta su peso y la forma del mismo.

#### 3.3.1 ARMAZÓN POLIESTIRENO EXPANDIDO

Se realizó la construcción del primer armazón del Sistema Aéreo, el material que se usó fue el poli-estireno expandido o comúnmente llamado espuma flex. En la Figura 2.57 se puede observar el armazón que contenía a los elementos electrónicos del Sistema Aéreo.



Figura 2.57 Armazón de espuma flex

El armazón de poli estireno fue usado en el primer vuelo como se puede ver en la Figura 2.58, durante el vuelo se tuvo problemas de comunicación con el control del dron ya que desde tierra no se tenía línea de vista con la antena del dron y durante algunos minutos se perdió el control del dron. El armazón se sujetó al dron mediante el uso de amarras plásticas.



Figura 2.58 Prueba de vuelo 1

Debido a este problema se tomó la decisión de cambiar el tipo de armazón que va a llevar los elementos electrónicos del sistema aéreo.

### 3.3.2 ARMAZÓN PLÁSTICO

Luego de realizar la primera prueba de vuelo se evidenció que se necesita un armazón que contenga los dispositivos del Sistema Aéreo más pequeño y de un peso aproximadamente parecido al armazón del primer intento.

Para la construcción del segundo armazón se reutilizó una cámara de seguridad, en la Figura 2.59 se puede ver como se encuentra la tarjeta Raspberry Pi 3 dentro del armazón de la cámara de seguridad. El armazón plástico tiene forma circular con un diámetro de 11,2 cm y la tarjeta Raspberry Pi 3 mide 8,5 x 5,6 cm por lo tanto cabe perfectamente donde se encontraba las partes electrónicas de la cámara de seguridad.



Figura 2.59 Raspberry dentro de armazón

En la Figura 2.60 se puede observar cómo se colocó la cámara térmica. Se realizó una perforación en el plástico ya que la cámara térmica enviaba imágenes erróneas debido al plástico.



Figura 2.60 Posición cámara térmica

Finalmente se utiliza la tapa plástica de la cámara de seguridad para cubrir la tarjeta Raspberry Pi 3, en la parte superior de esta tapa se colocó la batería y el módulo GPS que debe tener línea de vista con los satélites para captar la información de ubicación. En la Figura 2.61 se observa el armazón con todos los elementos electrónicos a ser usados.



Figura 2.61 Armazón Plástico

### 3.4 PRUEBAS DE FUNCIONAMIENTO

Las pruebas de funcionamiento del prototipo se realizaron en la cancha de la Escuela Politécnica Nacional, adicionalmente se escogió el sector del volcán Pululahua, para probar el prototipo en condiciones de neblina.

Para realizar las pruebas de funcionamiento se requiere seguir el siguiente procedimiento:

1. Encender tarjeta Raspberry Pi.
2. Esperar que el GPS reciba información.
3. Ejecutar software del Sistema Tierra.
4. Realizar la conexión entre el Sistema Tierra y el Sistema Aéreo.
5. Visualizar video térmico e información de ubicación.
6. Realizar la calibración del dron.
7. Fijar armazón plástico al dron mediante cuerdas.
8. Realizar el vuelo.

### 3.4.1 PRUEBAS DE FUNCIONAMIENTO CANCHA ESCUELA POLITÉCNICA NACIONAL

Se realizó un sobrevuelo sobre la cancha de fútbol de la Escuela Politécnica Nacional y se procedió a realizar 5 capturas. A continuación, se presenta un resumen de las imágenes térmicas que se capturaron aproximadamente en un radio de 20 metros del punto que se puede ver en la Figura 2.62 y Figura 2.63.

Las siguientes coordenadas fueron tomadas desde el aplicativo de Google Maps y sirve como referencia para contrastar la información que el módulo GPS entregó en el momento de las pruebas.

Latitud: -0,211835

Longitud: -78,489206



Figura 2.62 Cancha de fútbol de la Escuela Politécnica Nacional



Figura 2.63 Sitio de pruebas de funcionamiento

### 3.4.1.1 Captura 1

La primera captura se realizó a los bordes donde se practica el deporte de salto alto, que está ubicado a un costado de la pista atlética. Como se puede observar en la Figura 2.64 tanto la pista como los bordes de cemento emiten mayor calor que el césped y tierra que está alrededor, se debe considerar que el día que se realizó las pruebas fue un día soleado por lo que cemento y la arena se encuentran cargados más calientes.

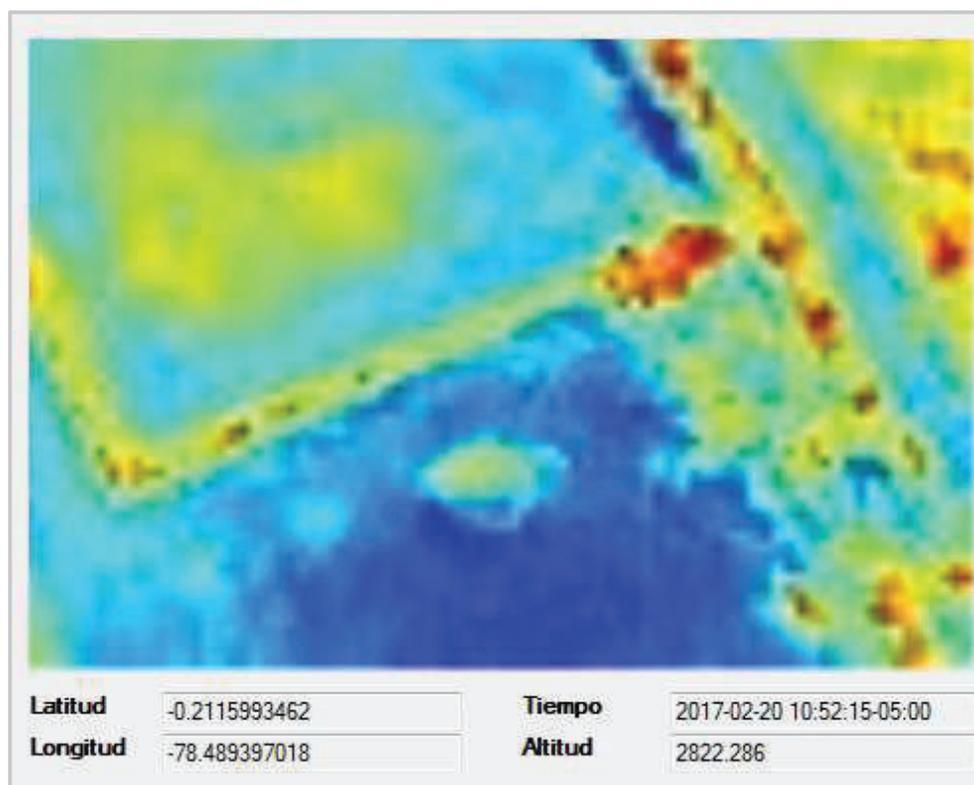


Figura 2.64 Escuela Politécnica Nacional: Captura 1

### 3.4.1.2 Captura 2

Sistema Aéreo se procedió a dirigirlo hacia la cancha de fútbol de la Escuela Politécnica Nacional para realizar más capturas de imágenes. En el trayecto se pudo capturar la imagen térmica del borde de césped y el comienzo de la pista atlética, en esta imagen se puede ver la diferencia de emisión de radiación térmica de las dos superficies. En la Figura 2.65 se presenta la captura térmica de esta parte del área de pruebas. La Figura 2.66 se presenta la imagen que se obtuvo del aplicativo Google Earth y ayuda a verificar qué área se capturó.

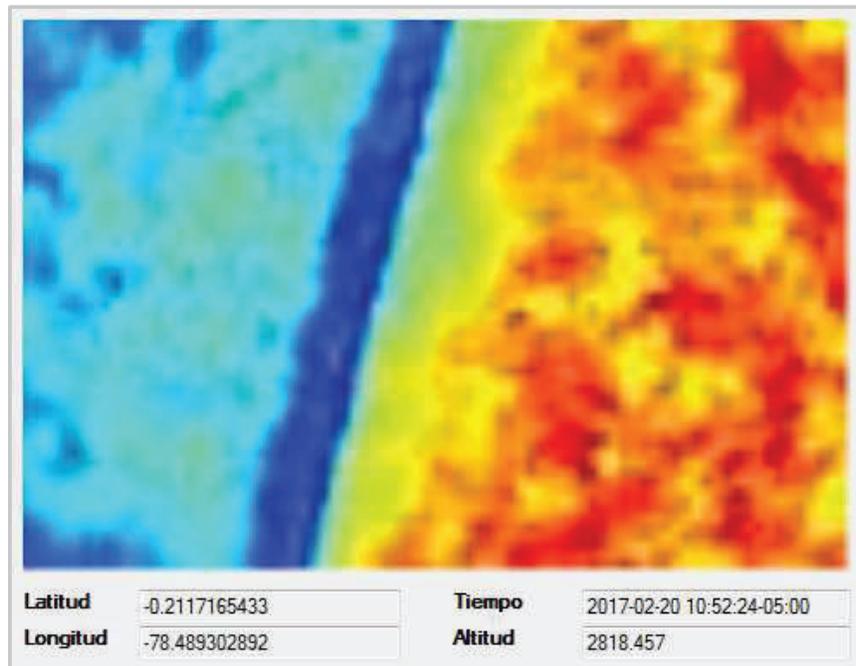


Figura 2.65 Escuela Politécnica Nacional: Captura 2



Figura 2.66 Borde de Cancha

### 3.4.1.3 Captura 3

La tercera captura se realizó aproximadamente en el centro de la cancha y lo que se trató de verificar es la imagen térmica de un arco de fútbol. En la Figura 2.67, en el

cuadro rojo se observa líneas rectas de color azul que corresponden a los tubos de un arco de fútbol que se encontraba en ese momento en esa posición del terreno.

Como se puede observar la radiación emitida por este material no es tan fuerte, pero se puede distinguir, adicionalmente en esta captura térmica se observa partes del terreno que no están cubiertas con césped es por ello por lo que se aprecia partes donde la tierra es la que emite la radiación térmica con mayor intensidad.

Según la información del módulo GPS, el Sistema Aéreo se encontraba a una altura de aproximadamente 15 metros más alto en comparación con la Captura 2 que se encontraba a una altura de 2.818 metros.

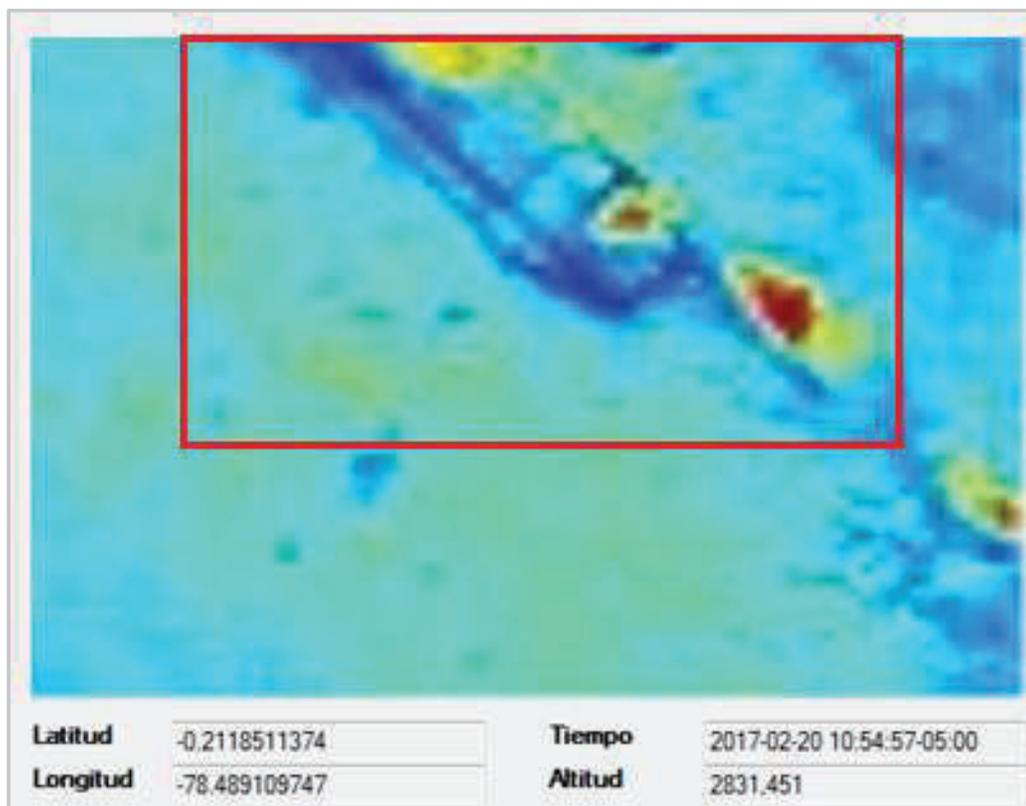


Figura 2.67 Escuela Politécnica Nacional: Captura 3

#### 3.4.1.4 Captura 4

El Sistema Aéreo se procedió a elevar a una altura de 2.383 metros de altura. Como se había mencionado anteriormente, a esa altura se aprecia con mayor intensidad de radiación los espacios de la cancha que no están cubiertos con césped.

En el cuadro de color rojo de la Figura 2.68 se presenta la imagen del usuario que se encontraba realizando las capturas desde el Sistema Tierra. Se procedió a ubicar a la persona sobre el espacio que no hay césped en la cancha de fútbol. A esta altura no se puede distinguir con claridad el cuerpo humano que se encuentra sobre este espacio.

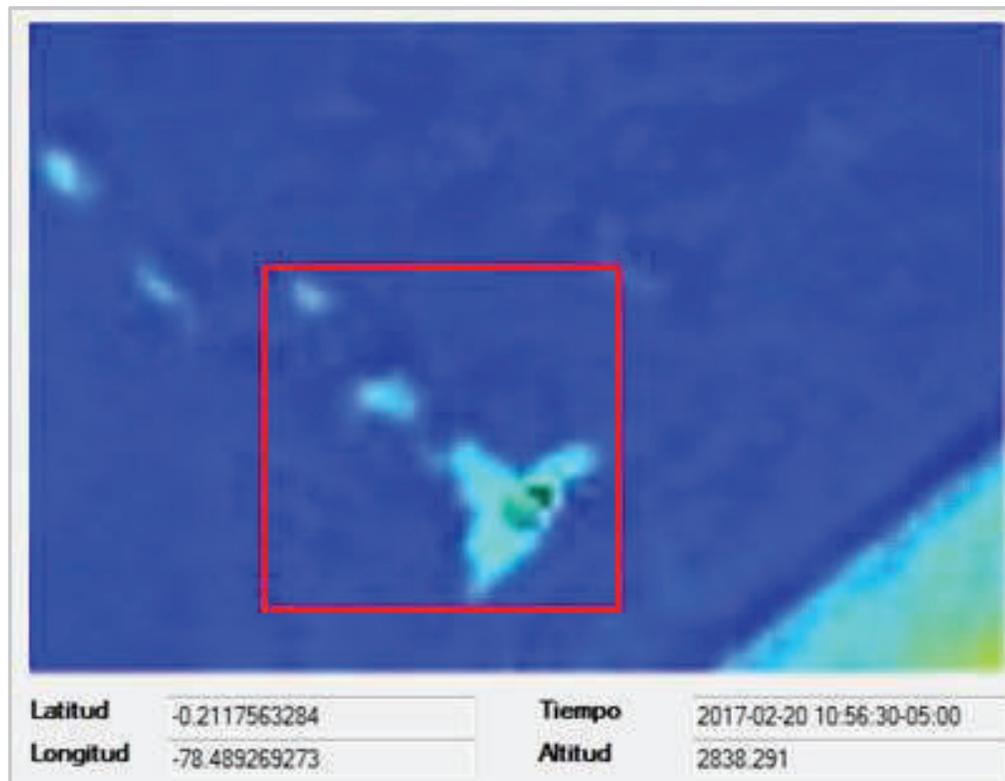


Figura 2.68 Escuela Politécnica Nacional: Captura 4

#### 3.4.1.5 Captura 5

En la quinta imagen térmica se desplazó a la persona a un espacio de la cancha de fútbol con césped. Para efectos de la prueba la persona se recostó sobre el césped y como se puede ver hay una mayor intensidad ya que el cuerpo humano emite mayor radiación térmica que la vegetación en este caso césped. Adicionalmente se puede apreciar un punto casi imperceptible a lado de la persona que corresponde al computador del Sistema Tierra.

En la Figura 2.69 se nota el mensaje “Sin conexión GPS” en el formulario, este mensaje se presenta cuando el módulo GPS no recibe información del Sistema Aéreo pero el *streaming* de video térmico se sigue recibiendo.

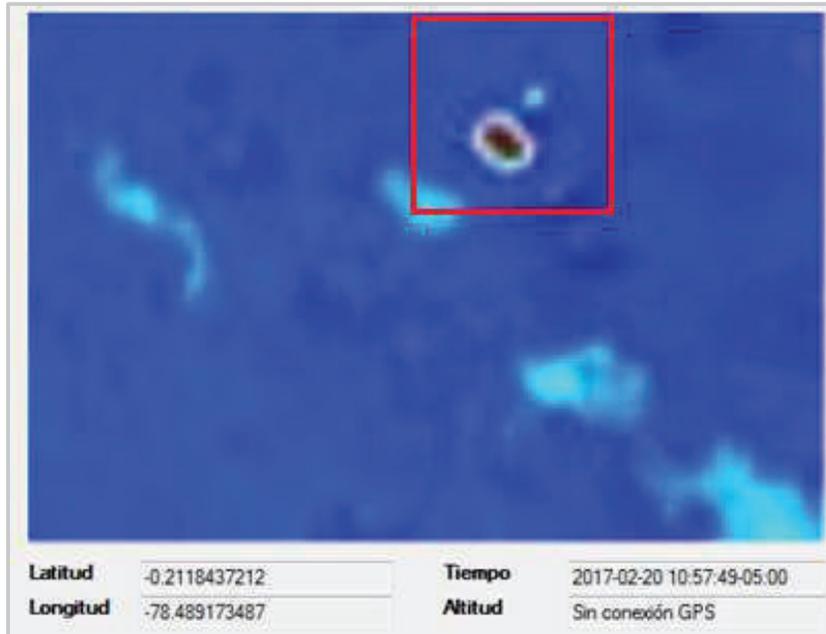


Figura 2.69 Escuela Politécnica Nacional: Captura 5

#### 3.4.1.6 Captura 6

Se desmontó del dron a la cámara térmica juntamente con la tarjeta Raspberry Pi 3, se captura edificaciones que se encuentran alrededor de la Escuela Politécnica Nacional. Como se puede observar en la Figura 2.70 se aprecia una parte de la cancha de césped, la pista atlética y edificaciones que se encuentran alrededor. Esta prueba se realizó como una prueba adicional y se puede verificar el funcionamiento de la cámara térmica la cual analiza que material tiene mayor calor y lo presenta con un color rojo.

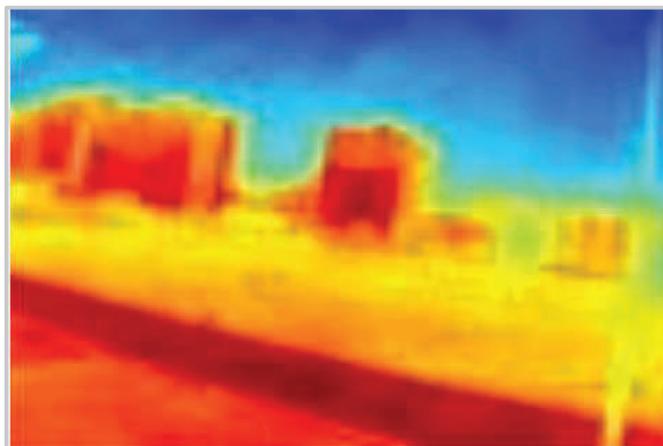


Figura 2.70 Escuela Politécnica Nacional: Captura 6

### 3.4.2 PRUEBAS EN EL VOLCÁN PULULAHUA

El presente prototipo se lo podría utilizar para la búsqueda de personas en lugares de difícil acceso y visibilidad por lo que se buscó un lugar de pruebas donde exista la presencia de neblina para comprobar su funcionamiento bajo estas condiciones.

Las pruebas se las realizaron en el sector del volcán Pululahua, se ubica a 17 km de la ciudad de Quito y se encuentra a una altura de entre 1.800 metros a 3.400 metros. Se puede evidenciar que existe la presencia de neblina aproximadamente a partir de 16:00.

En la Figura 2.71 se puede observar el lugar con la neblina el momento de realizar las pruebas de funcionamiento.



**Figura 2.71 Sector Pululahua**

Para verificar las coordenadas del sitio de pruebas con neblina se utilizó la herramienta Google Maps, en la Figura 2.73 se puede apreciar las coordenadas del sitio y posteriormente comparar con las obtenidas por el prototipo. Adicionalmente se utilizó un dispositivo GPS para comprobar la ubicación en la Figura 2.72 se puede ver las coordenadas y la altitud en el momento de las pruebas.

La información de ubicación que entregó el GPS fue  $0^{\circ}01'25.9''N$   $78^{\circ}30'14.9''W$ .



Figura 2.72 Medición de ubicación

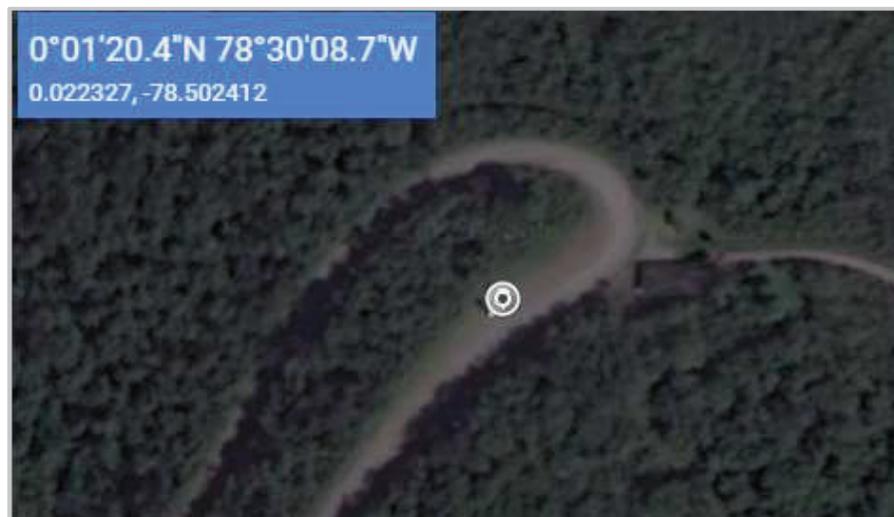


Figura 2.73 Coordenadas sector Pulahua

#### 3.4.2.1 Captura 1

La primera captura se realizó con el Sistema Aéreo aproximadamente a nivel del terreno. Como se puede ver en la Figura 2.74 se observa la forma de unos pies y se puede distinguir claramente que el Sistema Aéreo se encuentra a nivel de la tierra aproximadamente a 3.075.79 metros.

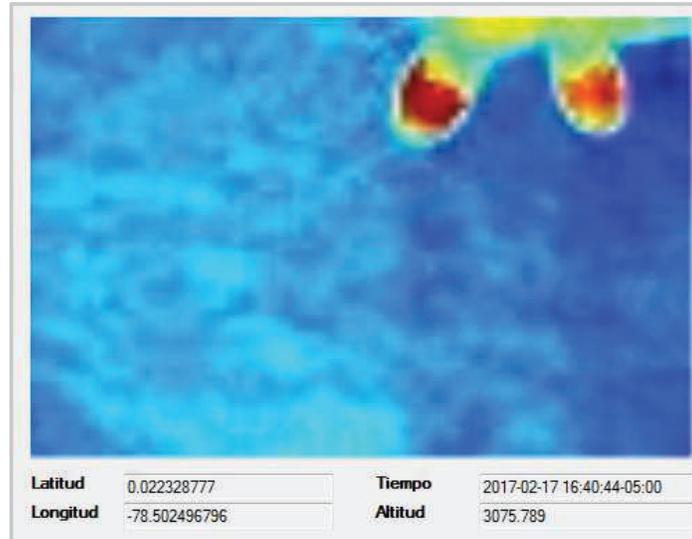


Figura 2.74 Prueba en neblina: Captura 1

### 3.4.2.2 Captura 2

En la segunda captura no se obtuvo la información de altitud ya que el módulo GPS no recibió la información de mínimo 4 satélites por lo que simplemente se puede ver las coordenadas. En la Figura 2.75 se presenta el resultado de esta captura y puede observar cuatro manchas de color amarillo que representan a un vehículo y a tres personas que se encontraban debajo del Sistema Aéreo.

El motor del vehículo es el que tiene mayor carga térmica y por ende se puede observar con mayor intensidad en la imagen.

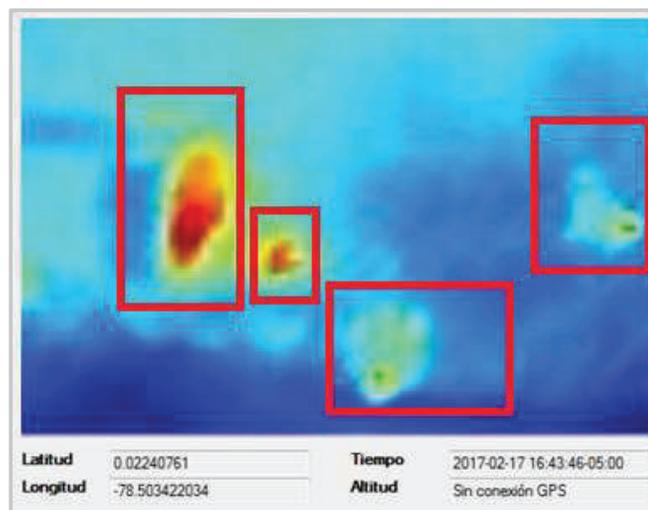


Figura 2.75 Prueba con neblina: Captura 2

### 3.4.2.3 Captura 3

La tercera imagen fue capturada a una altura de 3.097 metros. En la Figura 2.76 se puede ver que la radiación térmica de las personas desapareció en la imagen térmica, el único elemento que se puede distinguir es el vehículo por el tamaño del mismo. El calor del terreno empieza a observarse con mayor intensidad de amarillo en la imagen térmica.

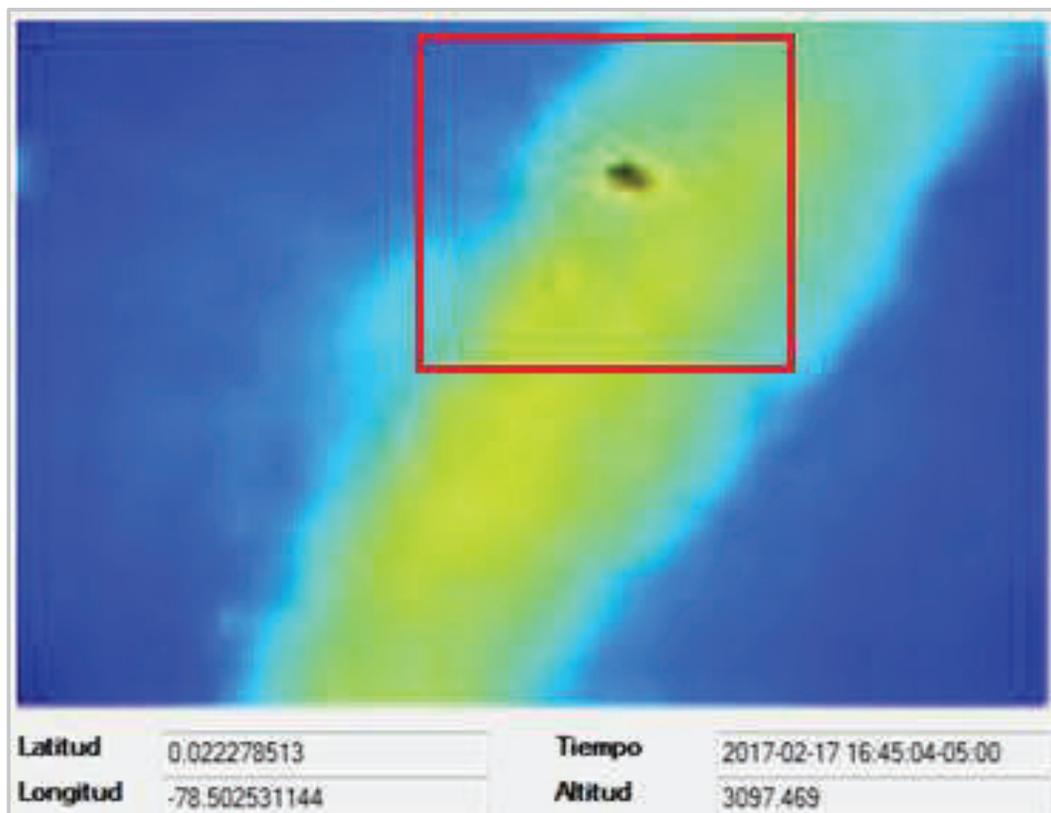


Figura 2.76 Prueba en neblina: Captura 3

### 3.4.2.4 Captura 4

Para obtener la cuarta imagen el Sistema Aéreo se elevó aproximadamente 3 metros más que la captura 3, como se puede ver en la Figura 2.77 el Sistema Aéreo se encontraba 3.100.185 metros.

Debido a la nueva posición del Sistema Aéreo, la cámara térmica detecta más radiación térmica del suelo, en este caso es la vía de tierra donde se realizó las pruebas, pero todavía se puede observar el calor que emite el motor del vehículo con un menor tamaño.

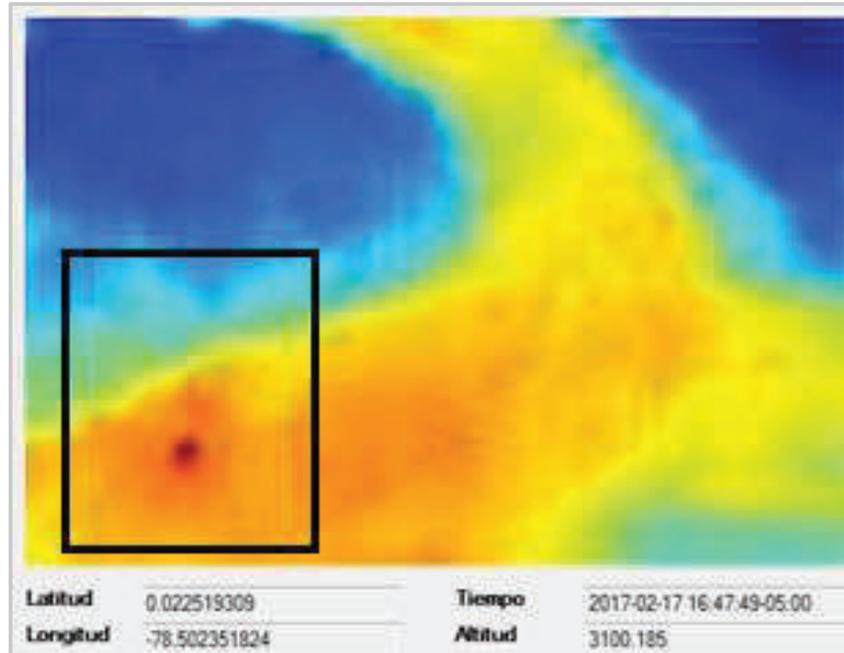


Figura 2.77 Prueba con neblina: Captura 4

Con ayuda de la herramienta Google Earth se puede contrastar la imagen térmica con la imagen entregada por la aplicación de Google. En la Figura 2.78 se presenta la imagen paralela de la imagen térmica.

La forma de la carretera en la imagen térmica se puede comparar con la imagen sin filtro y se puede ver que corresponde a esta parte del camino.



Figura 2.78 Prueba con neblina: Imagen Google Earth

### 3.4.2.5 Captura 5

Para la quinta imagen se realizó una elevación de aproximadamente 41 metros más con respecto a la primera imagen como se puede observar en la Figura 2.79. En

esta altura simplemente el Sistema Aéreo distingue exclusivamente la radiación de la carretera de tierra.

El Sistema Aéreo empezó a presentar problemas de conectividad con el Sistema Tierra como se puede ver en la Figura 2.80. La comunicación dron y control remoto se perdió por lo que el vehículo se quedó por varios segundos sin control desde tierra.

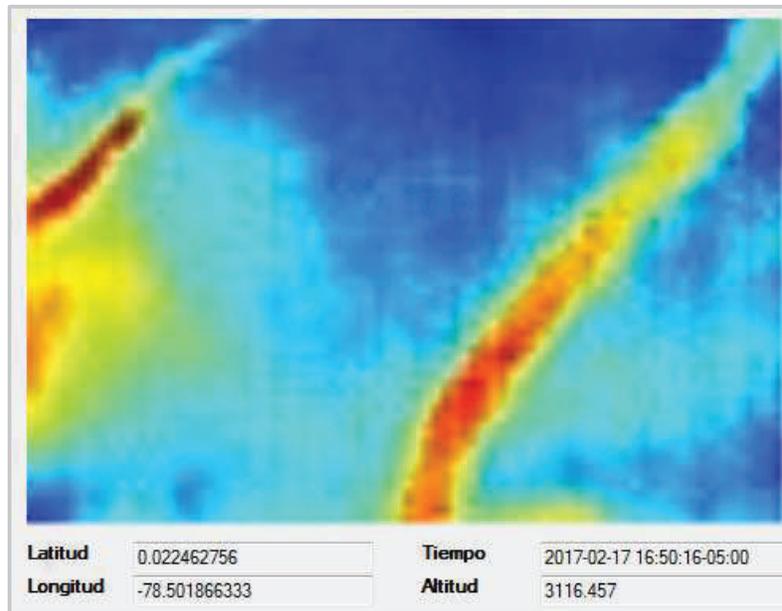


Figura 2.79 Prueba con neblina: Captura 5

```

Administrador: Símbolo del sistema - ping 10.10.10.1 -t
Respuesta desde 10.10.10.1: bytes=32 tiempo=0ms TTL=64
Respuesta desde 10.10.10.1: bytes=32 tiempo=5ms TTL=64
Tiempo de espera agotado para esta solicitud.
Respuesta desde 10.10.10.1: bytes=32 tiempo=16ms TTL=64
Respuesta desde 10.10.10.1: bytes=32 tiempo=215ms TTL=64
Respuesta desde 10.10.10.1: bytes=32 tiempo=7ms TTL=64
Respuesta desde 10.10.10.1: bytes=32 tiempo=7ms TTL=64
Respuesta desde 10.10.10.1: bytes=32 tiempo=29ms TTL=64
Respuesta desde 10.10.10.1: bytes=32 tiempo=9ms TTL=64
Respuesta desde 10.10.10.1: bytes=32 tiempo=5ms TTL=64
Respuesta desde 10.10.10.1: bytes=32 tiempo=13ms TTL=64
Respuesta desde 10.10.10.1: bytes=32 tiempo=18ms TTL=64
Respuesta desde 10.10.10.1: bytes=32 tiempo=103ms TTL=64
Respuesta desde 10.10.10.1: bytes=32 tiempo=10ms TTL=64
Respuesta desde 10.10.10.1: bytes=32 tiempo=471ms TTL=64
Tiempo de espera agotado para esta solicitud.
Respuesta desde 10.10.10.1: bytes=32 tiempo=12ms TTL=64
Respuesta desde 10.10.10.1: bytes=32 tiempo=187ms TTL=64
Tiempo de espera agotado para esta solicitud.
Respuesta desde 10.10.10.1: bytes=32 tiempo=31ms TTL=64
  
```

Figura 2.80 Prueba con neblina: Conectividad

De acuerdo con las pruebas realizadas se podría asumir que la distancia máxima en ambientes con neblina podría ser aproximadamente 40 metros ya que como se ve en la Figura 2.80 la comunicación inalámbrica es deficiente posiblemente debido a la humedad del ambiente.

## CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES

En este capítulo se escribirán las conclusiones y recomendaciones obtenidas en la implementación del trabajo de titulación.

### 4.1 CONCLUSIONES

- La tarjeta Raspberry Pi 3 permite realizar un sin número de proyectos añadiendo periféricos como cámaras, pantallas, sensores o cualquier dispositivo que se pueda conectar tanto en sus interfaces USB como en el interfaz GPIO. Raspberry Pi permite que se pueda aprender de mejor manera sistemas operativos libres y lenguajes de programación como Python.
- La cámara térmica Flir Lepton detecta los cambios de radiación de los cuerpos, el movimiento de la cámara puede en un instante presentar una imagen diferente si se cambia de posición, esto se pudo evidenciar en las pruebas realizadas en el sector del Pululahua ya que en ciertas fotos se puede apreciar como la radiación térmica de la carretera es muy fuerte y por ende la imagen presenta un color amarillo pero al realizar el movimiento del Sistema Aéreo en altura y posición la imagen va cambiando y se puede observar con mayor intensidad la radiación del motor del vehículo que se encontraba en el lugar de las pruebas.

En la Figura 2.74 también se puede evidenciar lo antes mencionado ya que al estar tan cerca se puede notar con mucha claridad la forma de los pies del operario y no la radiación del suelo.

- El módulo GPS para la entrega de información de ubicación necesita como mínimo la señal de tres satélites, con esto se tiene la información de longitud y latitud. Para tener información de altitud se necesita la señal de un cuarto satélite. Es por ello por lo que en ciertas capturas se puede verificar que no se tiene información de altitud ya que no se tuvo la señal del cuarto satélite. Además, si se tiene la señal de más de cuatro satélites aumentará la precisión de la información.
- Si se realiza un monitoreo de la información que entrega el módulo GPS cuando tiene el mínimo de satélites para funcionar el margen de error en

cuanto a la altura es de +/- 15 metros y eso varía según el nivel de señal que se tenga y el número de satélites.

- Debido a que la tarjeta Raspberry Pi utiliza un sistema operativo libre es posible encontrar librerías en repositorios que se reusaron y permitieron programar los scripts que conforman el Sistema Aéreo.
- El tiempo de autonomía del Sistema Aéreo depende de la batería del dron, según las especificaciones técnicas es un tiempo aproximado de 25 minutos. Mediante la aplicación de celular para controlar el dron se puede continuamente monitorear y verificar el tiempo de vuelo que se tiene de acuerdo con la distancia que se encuentre el dron.
- La tarjeta Raspberry Pi 3 presenta como característica principal la inclusión de una antena wifi en la placa que permite conectarse a redes inalámbricas con el estándar 802.11n, es por ello por lo que permitió suprimir la tarjeta inalámbrica que se debía adicionar al trabajo de titulación para la conexión con el Sistema Tierra.
- El aplicativo NOOBS permite a personas principiantes instalar sistemas operativos compatibles con la tarjeta Raspberry Pi.  
Además de permitir instalar el sistema operativo Raspbian mediante el aplicativo NOOBS se puede instalar sistemas operativos como Raspbian Lite, Windows 10 IoT, Lakka, etc.
- El alcance que tiene el prototipo se rige más por el tipo de imagen que se tiene ya que el Sistema Aéreo podría alcanzar alturas de más de 100 metros respecto al terreno donde se encuentra el Sistema Tierra, pero la imagen que se tendría no serviría de mucho si se requiere aplicar en el campo de la búsqueda y rescate.  
Como se puede observar en las pruebas realizadas en el lugar con neblina se puede ver que a una altura aproximada de 40 metros de altura la radiación térmica que predomina es la de la carretera o de la tierra por lo que si existiera un cuerpo en ese punto se podría perder en la imagen térmica.
- El lenguaje Python permite al usuario tener varias posibilidades el momento de programa un *script* ya que se tiene varias librerías que ayudan a administrar los periféricos, en el caso de este trabajo se utilizó la librería pylepton y la librería gps.

## 4.2 RECOMENDACIONES

- El manejo del dron se lo realiza desde un control remoto ubicado en tierra, este manejo del vehículo no tripulado se da a través de una comunicación inalámbrica que funciona en la banda de los 2.4 Ghz según las características técnicas del dron, se recomendaría establecer una red ad hoc entre el Sistema Tierra y el Sistema Aéreo mediante una red inalámbrica que funcione en la banda 5.6 Ghz que asegura que no se presenten interferencias entre la comunicación control remoto - dron y la del Sistema Aéreo – Sistema Tierra.
- Se recomienda que revisar la batería del control remoto, dron, Raspberry Pi para que el momento de realizar un sobrevuelo no se presente inconvenientes en el funcionamiento del prototipo.
- A pesar de que el prototipo se lo construye con un equipo de bajo costo se podría mejorar en el tipo de cámara para tener una mejor resolución de las imágenes, además se podría mejorar la antena del módulo de GPS para tener una mayor precisión en la información de ubicación.
- Se recomienda antes de montar el armazón con los dispositivos electrónicos verificar que tanto la cámara térmica como el módulo GPS se encuentren entregando la información deseada ya que si lo realiza durante el vuelo se desperdiciaría batería del dron sin obtener ninguna información.
- En un futuro trabajo se podría ver la opción de adicionarle un cámara que entregue fotografías sin ningún filtro que permitiría contrastar con la imagen térmica.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] Cole-Parmer. (2015) Repaso de la espectroscopia visible y ultravioleta. [En línea]. Disponible en: <http://www.coleparmer.com/TechLibraryArticle/6722> (Consultado el 15 de junio de 2016)
- [2] ¿Qué es el Infrarrojo? [En línea]. Disponible en: [http://legacy.spitzer.caltech.edu/espanol/edu/learn\\_ir/](http://legacy.spitzer.caltech.edu/espanol/edu/learn_ir/) (Consultado el 22 de junio de 2016)
- [3] CARACTERIZACIÓN TÉRMICA DE UN CONJUNTO DE EDIFICACIONES DEL PIRINEO OSCENSE MEDIANTE TERMOGRAFÍA INFRARROJA [En línea]. Disponible en: <https://riunet.upv.es/bitstream/handle/10251/17147/PFG%20ALEJANDRO%20MARRIÑO.pdf?sequence=1> (Consultado el 22 de junio de 2016)
- [4] ELDRONE [En línea]. Disponible en: <http://eldrone.es/historia-de-los-drones/> (Consultado el 22 de junio de 2016)
- [5] RCEXTREMO [En línea] Disponible en: <https://www.rcextremo.co/categoria-producto/accesorios-armar-drone-frame/chasis/> (Consultado el 22 de junio del 2016)
- [6] ROBOTSHOP [En línea] Disponible en: <http://www.robotshop.com/en/bldc-motors.html> (Consultado el 25 de junio del 2016)
- [7] EL VUELO DEL DRONE [En línea] Disponible en: <https://elvuelodeldrone.com/tienda-de-drones/accesorios-para-drones/helices/> (Consultado el 25 de junio del 2016)
- [8] ONE MAGAZINE [En línea] Disponible en: <http://www.onemagazine.es/industria-drones-tipos-existen-mercado-uav> (Consultado el 25 de junio del 2016)
- [9] PROMAX. (2014). Cámara Termográfica: Cómo funcionan y por qué pueden ser necesarias. [En línea]. Disponible en: <http://www.promax.es/esp/noticias/400/camara-termografica-como-funcionan-y-por-que-pueden-ser-necesarias> (Consultado el 26 de junio de 2016)
- [10] Dipler (2009). Funcionamiento del GPS [En Línea]. Disponible en: <http://www.dipler.org/2009/06/funcionamiento-de-gps/> (Consultado el 26 de junio de 2016)

- [11] Punto de referencia (2011). El protocolo NMEA. [En línea]. Disponible en: <http://geolocalizacion.blogspot.com/2011/01/lenguaje-del-gps-nmea.html> (Consultado el 26 de junio de 2016)
- [12] ComputerHoy.com (2015). Raspberry Pi: ¿Qué modelo me compro? [En línea]. Disponible en: <http://computerhoy.com/noticias/hardware/raspberry-pi-que-modelo-me-compro-23811> (Consultado el 26 de junio de 2016)
- [13] DJI. (2016). PHANTOM 3 STANDARD. [En línea]. Disponible en: <http://www.dji.com/es/product/phantom-3-standard/info#specs> (Consultado el 26 de junio de 2016)
- [14] International Business Times. DJI Phantom 3 Professional review (2015). [En línea]. Disponible en: <http://www.ibtimes.co.uk/dji-phantom-3-professional-review-4k-drone-perfect-amateurs-pros-alike-1507980> (Consultado el 26 de junio de 2016)
- [15] The Hacker News. Raspberry PI 3. [En línea]. Disponible en: <http://thehackernews.com/2016/02/raspberry-pi-3-microcomputer.html> (Consultado el 29 de julio de 2016)
- [16] PROMETEC. RASPBERRY Y GPIO. [En línea]. Disponible en: <http://www.prometec.net/rpi-gpio/> (Consultado el 29 de julio de 2016)
- [17] The 21st Century Digital Home (2016). [En línea]. Disponible en: <http://21stdigitalhome.blogspot.com/2016/02/two-raspberry-pi-3-hardware-hacks.html> (Consultado el 29 de julio de 2016)
- [18] ELECTRONILAB. [En línea]. Disponible en: <https://electronilab.co/tienda/modulo-gps-ublox-neo-7m-000-con-memoria-eeeprom-3v-5v/> (Consultado el 29 de julio de 2016)
- [19] PCEL [En línea]. Disponible en: <https://pcel.com/Genius-39800007102-107202> (Consultado el 29 de julio de 2016)
- [20] Raspberry Pi Foundation. [En línea]. Disponible en: <https://www.raspberrypi.org/downloads/> (Consultado el 29 de julio de 2016)
- [21] DELL. [En línea]. Disponible en: <http://www.dell.com/mx/p/inspiron-14-3421/pd> (Consultado el 29 de julio de 2016)

## **ANEXOS**

Anexo A: Código Sistema Aéreo.

Anexo B: Código Sistema Tierra.

Anexo C: UART GPS NEO-7M-C User Manual.

Anexo D: FLIR-Lepton-Datasheet.

Anexo E: Videos de las pruebas realizadas.